Context based Indexing in Information Retrieval System using BST

Neha Mangla[*], Vinod Jain[**]

[*,**] Department of Computer Science and Engineering,
B.S.Anangpuria Institute of Technology and Management, Village Alampur, Ballabgarh, Faridabad, Haryana, INDIA

Abstract- Searching of data relevant to our query is done by information retrieval system. Keyword searching is the basic idea of this system which tries to solve the large search space problem as the documents to be searched could be of any length. This means time to search will increase with length of document.
Search time will be reduced by reducing the search space. In this, we are constructing a method which reduces the searching area with the help of indexing that takes the help of stemming method and knowledge of stopwords. Representation of both, a word and more than one word are done by creating Indices using single concept. The recall is improved by including domain knowledge using ontology while searching.

Index Terms- Indexing, Information Retrieval, Keyword searching, ontology, search space

I. INTRODUCTION

The information retrieval takes into account- storing and representation of data as well as retrieval of relevant information according to users need. Searching of data relevant to a given query which is made by few words taken from a general language is called information retrieval system. The documents extracted during the indexing phase are compared with the query. The documents which resemble most are given to the users where they evaluate the relevance of document with respect to their need.

Earlier, the retrieval system was based on keyword searching. The documents and queries are matched by the words they contain in common. The document which have large number of words common with the query, the document will be said to be more relevant. This retrieval system is called to be coordination match system[6]. But, This system as few problem. First, a document can have words in many lexical forms 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, a set of token words representing their respective files is matched with the query which is very difficult and confusing task. Third problem, sometimes no match situation arise i.e. words in query do not match the files. In this case we have to increase our recall, where recall is fraction of relevant document that are retrieved. The unuseful words should not be saved in search space where token words are saved, these words are called stopwords. The problem of multiple form can be solved by some stemming algorithm. The query is expanded to add recall to our system using ontology and domain knowledge[4]. To maintain this idea, we use a good stemming algorithm, ontology using domain knowledge and a ranked retrieval approach that performs the ranking on documents based on different user query. A phrased query can also be an important term, therefore retrieving of document is done phrase based and term based separately.

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. Finally, Section 4 covers conclusions and future work.

II. LITERATURE SURVEY

The earlier used keyword based searching approach has many pros and cons and modification in this has been carried out in a positive way. Some important one are discussed here.
One of the approach of keyword searching is based on precision-oriented search tasks that fulfill above discussion also.
Second approach gave an idea is to produce output in some time limit then these outputs are shown to the user with the different query forms. This solution is checked over real world data by making some experiments for the feasibility of solution.
Third research approach on ontology driven information retrieval. On large-scale search systems, ontologies are efficiently applied for retrieving data. Now the ontology's are become useful in every system as it gives a deep understanding as well as easy integration of different documents[2]. These concepts are adapted for domain basics with the computation of vector space in retrieval system. One more approach made by researchers is comparing which technique is a better, manual or automatic indexing technique. It was presumed that manual indices are better than automatic machine indices. But by performing many demonstrations, it has been found that both techniques are equal for text retrieval techniques. If we take both text and phrase based techniques together then, manual indexing is better than automatic indexing.

III. PROPOSED METHOD

We proposed two types of text retrieval process i.e. term based text retrieval and phrase based text retrieval. They are discussed as follows:

A. Indexing

Preprocessing of document: A new representation is made from the raw documents known as bag of terms formation[4]. This change is called document representatives. For the above representation work, we first take each document and collect the words of it to create the file except the stop words. We order these words in the file so that we could have all important terms in their basic forms. A threshold value is set based on formula consisting of file size. Thereafter frequency of each word is counted and if the frequency count of any word exceeds the threshold value than that word is selected as index term. Index terms are collected for document representation to create index table for that document. The above work is same in phrase based search with a single difference. In this, phrase identification program identifies the phrases by making a check on its frequencies, instead of terms everything is done on phrases. A phrase having suitable count more than threshold value is selected for generating index table as its indices.

B. Query formulation

We must first have the domain knowledge of the query to expand it and create an ontological tree structure. Each word of the query is searched in the above created tree. The word parent, children and siblings are added in the query for some significant reason. As sibling represent similar and parent-children represent relative meaning. When the search is made against files then we can increase our recall. The next step is to apply the Preprocessing approach told above. This is how our query representation is done. Same set of process is used in phrase based search.

C. Comparison

The query given by user is compared preprocessed documents by the system itself and a list of document chosen from it. A classification decision is made for order the documents. These inorder or parts of documents are shown to user. A decision is made by the user if he want a tree or not for expanding the query. The above process of comparing the query with documents is done on the basis of matrix multiplication. In this approach, all document terms are converted into doc_id matrix by which a term matrix is generated. A multiplication is made on different documents with the user query. This multiplication identifies the relevancy of the document with the query. It is shown in the mathematical form as:

Lets consider 2 document (i and j) 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 document terms are 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.

Weighted values and implication of inverted document frequency (IDF) are compared with each other:

- Weight value is calculated as the no. of times a term appeared in particular document. Therefore, It implies the relevancy of document.
- Inverse document frequency calculates the terms which are found in less documents. The less the documents a term occur in, the higher this IDF weight is.
- Thus, both the factors together will tell the importance of document for particular term.

D. Id by terms matrix generation

1. Weight is calculated for each term in the list.
2. IDF is calculated for each term in the documents as:
   Idf (i) = N/ni;

Where N = no. of documents in repository,
ni = No. of document in which term i occurred

3. W(i) is calculated for each term in the list.
   W(i) = weight*Idf(i)
4. Create a matrix(id X term) 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.

IV. ARCHITECTURE

The architecture of the proposed indexing is given in figure-1. The indexer module creates a context based index. This index is used by page parser to get token list with framing. Finally the token list is added in bst for frequency etc.

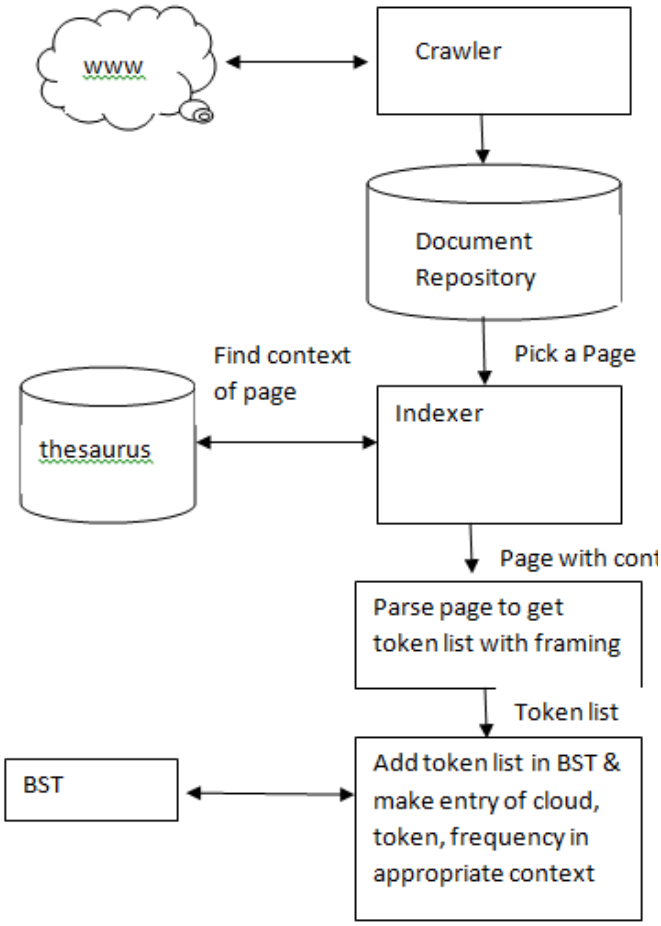

Figure-1 Proposed Architecture of context based indexing using bst

Description of various components

- Crawler

The crawler's job is to download and store the pages in a repository. This repository stores every documents to be indexed and searched for a user query.

- Indexer

The indexer is used to parse the documents of the page repository and make every token's entry in the index. It also assign context to a document. Finding the context of a document is not the area of concern of this paper. It is assumed that a component of the indexer will do this job.

- Thesaurus

A thesaurus is a work that lists words grouped together according to similarity of meaning, in contrast to a dictionary, which provides definitions for words, and generally lists them in alphabetical order.

- BST

BST is a node-based using binary tree data structure where each node has a comparable key and satisfies the restriction that the key in any node is larger than the keys in all nodes in that node's left subtree and smaller than the keys in all nodes in that node's right sub-tree.

V. CONCLUSION

In this paper we have described various problems and tried to solve them. A concept called stemming is used for lexical form word's. Index table length is varied for different documents by creation of threshold. Stopwords are not helpful; therefore search space is reduced by removing them. Our comparing and multiplication method gives the relevant result, therefore became useful as index based retrieval system. Our main objective is to generate the contextual sense for getting the most relevant documents.

REFERENCES

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

[2] Jan Paralic and Ivan Kostial " Ontology-based Information Retrieval," Department of Cybernetics and AI, Technical University of Kosice,Letna 9, 040 11 Kosice, Slovakia

[3] Urvi Shah, Tim Finin, Anupam Joshi, R. Scott Cost and James Mayfield "Information Retrieval On The Semantic Web," Dept. Comp. Sci. University of Maryland Baltimore County Baltimore, MD 21227

[4] Ambesh Negi, Mayur Bhirud, Dr. Suresh Jain and Mr. Amit Mittal "Index based information retrieval system, " vol.2 issue.3 may-june2012 pp-945-948

[5] Khaled M. Hammouda, Student Member, IEEE, and Mohamed S. Kamel "Efficient Phrase-Based Document Indexing for Web Document Clustering." IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 16, NO. 10, OCTOBER 2004

[6] http://en.wikipedia.org/wiki/Information_retrieval

AUTHORS

First Author – Neha Mangla is a M.Tech student in computer science and engineering at B.S.Anangpuraia Institute of technology and Management, Faridabad. (nehamangla.06@gmail.com).

Second Author – Vinod Jain is working as a lecturer in information technology department at B.S.Anangpuraia Institute of Technology and Management, Faridabad since September 2008. He has completed master of computer application (MCA) in June 2004 and Master of Technology in 2012. His area of research include IR systems and Genetic Algorithms. (jainvinod81@gmail.com)