AN EFFICIENT INFORMATION RETRIEVAL SYSTEM USING QUERY EXPANSION AND DOCUMENT RANKING

[1]P.JANARTHANAN, [2]A. KAVITHA,[3]N.RAJKUMAR,[4]S.P.VICTOR

[1]Sri Venkateswara College of Engg., Department of Computer Applications, Sriperumbudur, ,India

[2]R.M.K Engineering College, Department of M.C.A, Kavarapettai, India

[3]Sri Ramakrishna Engineering College, Coimbatore, India

[4]St. Xavier College, Tirunelveli, India

E-mail: [1]janap@svce.ac.in, [2]aka.mca@rmk.ac.in, [3]nrk29@rediffmail.com,[4]victorsp@rediffmail.com

ABSTRACT

Information retrieval is the process of searching and retrieval of information from documents that matches user query. The user information requirement is represented by a query or profile that contains one or more search terms. Indexing plays important role to retrieve the information. Researchers have been used indexing techniques only for document indexing and not focused on the speed up the search and retrieval time. In this paper, an enhanced inverted indexing technique is proposed to index all root terms of the documents. This approach maintains the weight of each term connected to the document in index structure. Multiple query terms are easy to handle using inverted index. After indexing, the searching and retrieval process is made by matching the query with the indexed terms. In order to solve the problem of false and null information retrieval, the proposed system includes query expansion and document ranking which improves retrieval accuracy.

Keywords: *Inverted index, Similarity Measure, Query Expansion, Document Ranking.*

1. INTRODUCTION

Information Retrieval (IR) is basically the requisite of information and documents from a corpus that should be retrieved to satisfy a user need of information. The document retrieval is one of the fastest growing and complex research areas in the field of information retrieval [5]. An effective Information retrieval can be achieved only under strong and efficient document retrieval methods. The user information requirement is represented in the form of query and comprised of one or more search terms with some additional information such as importance of term weights [6]. Hence, the retrieval decision is made by comparing the query term with the index terms appearing in the document itself. The decision may be binary (retrieve/reject) or it may involve estimating the degree of relevance that the document has to be queried. Internet is the ultimate source of information in the present world. One can find millions of documents related to any topic on the web. To have access to all the relevant documents related to a user query one need to have a system which can efficiently and effectively retrieve them. Information retrieval systems are widely used to help users find the required information and documents as per user needs[2]. In the information retrieval area the typical problem set is studied. A user tries to satisfy an information need in a given collection of documents [7]. For that purpose the user inputs a request into the information retrieval system containing the collection. The goal of the system is to retrieve documents with information content that is relevant to the user information necessity.

1.1 Data Pre-Processing

The data pre-processing method consist of three phases namely special character removal, stop words removal and case conversion.

The first phase removes all the 32 special characters from all documents. The 32 special characters show in Figure 1.

!, @, # , . , " , \ , $, % , ^ , &
, * , (,) , - , + , = , _ , { , } , [
,], ; , : , |, <, >, ? , /, ~ , `, , , \

Figure 1. Special Characters List

The second phase eliminates all the 677 stop words from all documents. The some of the stop words are shown in Figure 2.

a, an, the , is , are , there, who, what,
when, how, much, this, that,.. etc.

Figure 2. Stop Words List

The third phase converts the entire document from upper case to lower case to overcome the problem of context sensitivity while searching the information in the corpus. The user query is not necessary to match with the stop words like a, as, but. .etc. To overcome this problem the proposed method helps to reduce the searching time and identifying the root words and also help to minimize the inverted index size. This data preprocessing procedure is common for all text mining problems and operations [4].

1.2 Stemming

Stemming is a process to converting the words having morphological similarity into one common form. Stemming reduces inflected or sometimes derived words to their stem. Base or root form is normally a written word form. A stemming algorithm is used to reduce a word to its stem. The stem corresponds to the key terms of a query or document instead of the original word. One such transformation is conversion of plurals to singulars and removing endings of the word to obtain this stem[1]. Stemming process has been applied to improve retrieval effectiveness and to reduce the size of indexing files. For example,

Words → "*studies, studying*" Stem → "*study*"

1.3 Semantic Similarity

Semantic Similarity is an approach to find the Similarity between two different terms based on some metrics. The Similarity of the two words is related to their commonality. The targeted approach is to find that how the given term is semantically similar to another term. Method to compute the similarity between short texts or sentences has many applications in natural language processing and related areas such as information retrieval [11].There are many approaches available to find similarity between word pair, sentence and paragraph. Selecting any one of this approach is depending upon our problem requirement [12][13].

1.4 Query Expansion

Query expansion (QE) is an Information Retrieval (IR) technique. Its aim is to improve the relevance and quantity of the results retrieved by IR systems. It starts with the observation that many queries do not return the whole set of relevant documents[14]. Indeed, there usually is an inconsistency, stated and measured by several works between the queries and the corresponding indexed documents. QE overcomes this issue. It consists of adding new terms to the original query related to it or even to reformulate it[3] [9].

2. RELATED WORKS

The past research works on information retrieval suggested that retrieval of information was based on term frequency weight, document ranking, support vector machine and semantic similarity approaches.Bhavya Sukavasi, N.V. Ganapathi Raju et al proposed the Automatic Term Weighting and Inverted File Structure approaches for text documents and provides baseline of single term indexing to develop more elaborate techniques like content based analysis. They discussed the calculation of weight of the terms in a particular document with the help of TF (Term Frequency) and IDF(Inverse Document Frequency). Term Frequency refers to the number of times the word occurs in a particular document. IDF is ratio of total number of documents to the number of documents in which the word is present [4].They did not give importance to indexing the documents to improve the retrieval performance. B. Barla Cambazoglu and Cevdet Aykanat proposed the query processing in ranking-based text retrieval using similarity calculations and document ranking is computationally expensive. They mainly focused on information retrieval process with document ranking [7].

There have been many researches to develop methods in order to allow users to enter additional keywords into a search engine to uncover the majority of relevant documents or search outcome. Ping-I Chen, Shi-Jen Lin, Ya-Chi ChuIn presented a new algorithm named as Google latent semantic distance (GLSD) to mine the most significant sequence of keywords to provide nearly all relevant search results to the user [8].

Xiang Lian and Lei Chen proposed work deals with the problem of uncertain data management. Compared to certain data, the data in the uncertain database are not exact points which often reside within a region. They have used ranked queries over uncertain data [10].

Accurate measuring of semantic similarity between word pair is an important problem in web mining, information retrieval, and natural language processing. Danushka Bollegala, Yutaka Matsuo and Mitsuru Ishizuka proposed work identifies numerous semantic relations that exist between two given words. Two new algorithms were proposed namely, a novel pattern extraction algorithm and a pattern clustering algorithm. They concentrated to an automatic method to estimate the semantic similarity between words or entities using web search engines [5].

Benoit Gaillard, Jean-Leon Bouraoui, Emilie Guimier de Neef and Malek Boualem provides a new method that uses query expansion to develop multilingual information retrieval. Query Expansion (QE) module mainly uses linguistic resources to perform the expansion. The main aim is to use QE to overcome the limitations of machine translation and to retrieve more relevant results [9]. Pawan Gayal et al provided theoretical analysis of parametric query vector to satisfy need of user. They provided query presentation model that was better than vector model and language model. But it seems to be time consuming since response time is important [15].

3. PROPOSED WORK

In the field of information retrieval, a user tries to satisfy an information need from given collection of documents. For that purpose the user sends a request to the information retrieval system containing the documents. The goal of the system is to retrieve documents with information content that is relevant to the user query.

Three main concepts are used in proposed work. They are Inverted Indexing, Query Expansion, Document Scoring and Ranking. There are few pre-processing steps that must be performed prior to the information retrieval process. The pre-processing process contains the most common document preprocessing tasks like special characters removal, stop words removal and the stemming.

Word stemming is an important feature that has been developed for information retrieval in order to reduce morphological variants to their root form. Most of the words in a text document have various morphological variants. Since the variants have a similar semantics they can be considered as equivalent for the purpose of many retrieval tasks. Stemming is the process of reducing a word to its stem and a stemmer or a stemming algorithm is a sequence of steps that automates the task of stemming. The proposed works applied Paise/Huesk stemming algorithm to obtain root words.

Indexing is a technique that helps to index the unique terms of the document by maintaining the sentence structure. An indexing technique called inverted index is used that attaches each unique term with a list of all documents that contains the term. The proposed enhanced inverted index technique arranges the root word and its document list in an efficient manner. The term weight is obtained in this technique.

Our enhanced inverted index stores the root word name, the number of documents that contain the unique word, number of times the word occurs in a particular document and the position of the line where the word occurs in the document. Finally, the term frequency and document frequency of the unique term is stored.

An information retrieval process begins when a user enters a query into the system. The query is first Pre-Processed to remove special characters and stop words. Query expansion is done for the user query using Synonyms, Antonyms, Hypernyms and Hyponyms. The expanded query is then stemmed to obtain the root words of each term. The expanded query is then compared with all the unique terms in the inverted index and based on the similarity values two set of documents are retrieved. After the documents are retrieved, a rank is assigned for each document based on the weight obtained while indexing the document. Based on the rank value, the highest rank document is displayed followed by other subsequent ranked documents.

The proposed view of the system is shown in Figure 3. It clearly represents the step by step Process flow of the proposed system.

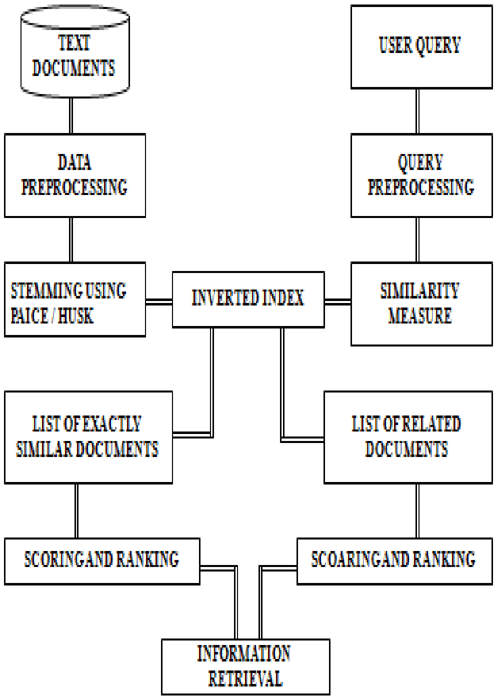

Figure 3. Proposed System Architecture

3.1 Stemming (Paice/Husk Technique)

Stemming is a process of obtaining root words from given documents. Stemming minimizes the size of index. There are several stemming algorithms available. Most of the research works in existing system employ Porter Stemming Algorithm proposed by M.F.Poter. It is also known as “Suffix Stripping”. The algorithm removes the suffixes from words to get the root word.

Porter’s algorithm does not remove suffix when the stem is too short that is when the number of vowel-consonant pairs in a word is zero. Porter’s algorithm has five major steps and hence it is time consuming. For large documents the algorithm is not efficient. Some of the research works in existing system utilize the “Table Lookup Approach”, that maintains a table of index terms and their stems such that the terms from queries and indexes could be stemmed very fast.
Most of the stemmed words are not available in English vocabulary and some terms are domain dependent. The size of the table used in Table Lookup Approach increases dramatically, decreasing the efficiency of the search process .The task of maintaining stem words for all terms is tedious job. The stemming technique employed encounters certain problem while obtaining the root words that delays response time. In addition, it fails to reduce document volume as well as index size. To overcome the existing system stemming technique problem, the proposed system employs a simple stemming algorithm called Paice/ Husk Technique. It is a simple affix deletion and replacement stemming algorithm. It is a recursive algorithm and consists of 120 rules as in table 1 for deletion and replacement. This proposed system stemming technique obtains the root word in the document with high accuracy. There are two important steps associated with this technique. They are

Step1: Reverse the word
Step2: Compare characters to its 120 rules.

If any one of the rule matches, perform corresponding removal and replacement and then perform step 2 until the minimum character length is 3.The few rules and its replacements are represented in table 1.

Table 1. Paice / Husk Rules

Rule	**Deletion**	**Replacement**	**Rule**	**Deletion**	**Replacement**
ai*2	-ia	---	j1s	-j	-s
a*1	-a	---	lbaifi6	-ifiabl	---
bb1	-bb	-b	lbai4y	-iabl	-y
city3s	-ytic	-ys	lba3	-abl	---
ci2	-ic	---	lbi3	-lbi	---
cn1t	-nc	-nt	lib2l	-bil	-bl
dd1.	-dd	-d	lc1	-cl	-c
dei3y	-ied	-y	lufi4y	-iful	-y
deec2ss	-ceed	-cess	luf3	-ful	---
dee1	-eed	-ee	lu2	-ul	---
de2	**-ed**	---	lai3	-ial	---
dooh4	-hood	---	lau3	-ual	---
e1	E	---	la2	-al	---
feil1v	-lief	-live	ll1	-ll	-l

If we want to stem the word *played,* we must follow the two steps:

Step 1 Reverse the word (“deyalp”)
Step 2 Compare last character ‘d’ with the 120 rules, until a rule is satisfied.

If the last character of word doesn’t satisfy rule, then compare last two characters “de” again with 120 rules, until a rule is satisfied. In our example, the rule is

Rule: *de2* → *-ed* Replacement: —

de2 satisfies our word *de* . *-ed* denotes the deletion part and "—"denotes replacement part. Here deletion is "ed" and replacement is null ('—"). After stemming, the root word "play" is formed from the word "played". Stemming technique plays important role to improve retrieval effectiveness and to reduce the size of indexing files.

3.2. Inverted Index

Indexing is the key part of the information retrieval system. It helps to obtain an inter-relation between the documents. The indexing technique is used for the arrangement of the root words in the documents. The existing approach of inverted index stores all the words in the index without maintaining the weight of term in a particular document. In this proposed system, an enhanced inverted indexing technique in figure 4 is used that indexes the entire root terms of the documents. This approach maintains the weight of each term corresponds to the document that contains the term. Multiple query terms are easy to handle in inverted index. Maintaining the weight of each term helps in ranking the documents according to frequency. The proposed inverted index technique is used to arrange the root word and its document list in a proper manner. When the retrieval process begins, it takes constant time to find the documents that contains a query term.

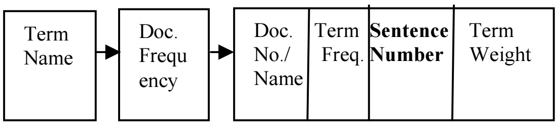

Figure 4. Proposed Inverted Index Structure

The structure of inverted index contains entire detail about the root term such as root word name, the number of documents that contains the root term, document number or name that contains the root term, number of times the word occurs in a particular document, sentence number where the word occurs in a corresponding document and the term frequency weight of the particular document. The term frequency weight is calculated with the help of Term Frequency (TF) and Document Frequency (IDF) using equation 1.

$$W\,(\text{term},\, d1) = (1{+}\log \text{tf}\,(\text{term},\, d1)) * \log\,(1{+}\, N/\, \text{df}\,(\text{term}) \qquad (1)$$

Proposed Algorithm: Inverted Indexing Construction

Input : Unstructured documents

Output: Inverted Index

Steps to be followed,

```
Step 1: Compare and count 'index term' occurs in
        how many documents
Step 2: do comparison of 'index term' with each
        Document
Step 3: if 'index term' occurs in document 'd1' or
        Other Documents
Step 4: count how many times occurs in a same
        Document.
Step 5: calculate 'weight of the term' in particular
        Document.
Step 6: Then add 'd1' details or other document
        details in 'Index term list'.
Step 7: end if
Step 8: while until all documents
Step 9: add 'Index term list' to inverted index
        structure.
```

The above proposed algorithm indexes all terms in the document collection and also it takes care of the root term occurrence in one or more documents. The document details are created with another inverted index structure and added to the previously created indexed structure for the same term if created already. The system process the documents one after another and also term by term. For each unique term while scanning, it creates inverted index structure and updates the whole detail about the term. If the same term appearing more than one time in the same document, its frequency will be incremented in the index structure. If the term appears more than one document, another inverted index structure will be created to the term for that document and linked with previously created inverted index structure. This process will be continued for all terms in the entire document Collections. The indexing of two documents related to cricket and football is represented in figure 5.

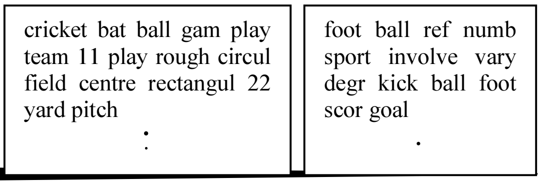
cricket bat ball gam play team 11 play rough circul field centre rectangul 22 yard pitch
:
foot ball ref numb sport involve vary degr kick ball foot scor goal
.

Figure 5. Stemmed Document

The term weight calculation for the term *cricket* is computed using equation 1 as

W (cricket, d1) = (1+log 1) * log (1+ 2/1)
W (cricket, s1) = (1+0) *(1.73) = 1.73

The above sample documents "cricket.txt" and "football.txt" in figure 5 are inserted into inverted index using proposed system algorithm. The construction of inverted index is represented in figure 6.

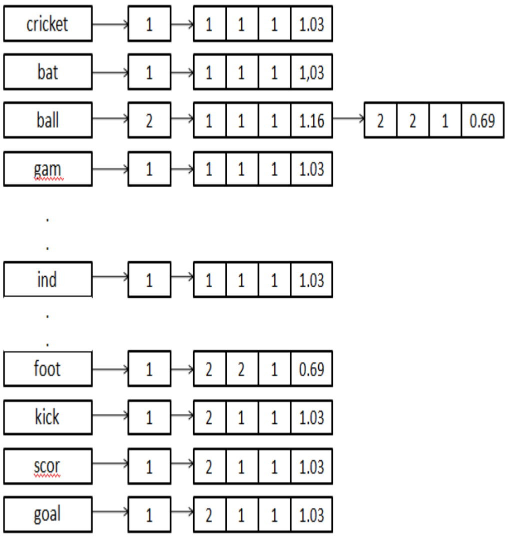

Figure 6. Proposed Inverted Index Construction

3.3. Query Expansion

In the existing system, Information is retrieved from inverted index that exactly matches user query with indexed term. But in some cases the user query may not match with inverted index term. In such cases, the information is not displayed.

In the proposed system, the user query is expanded with the help of synonyms, antonyms, hyponyms and hyponyms technique. Then the expanded terms are stemmed and matched with inverted index. The system has less possibilities of not retrieving the information.

After getting the query from user the query term is mapping to the synonyms list, if any terms found the synonyms list that word taken from the list and find the similarity value to all the inverted index terms and based on the similarity values retrieve the relevant information from exact document or relevant document. The same procedure will be applied for antonyms and hyponyms and hyponyms lists.

If the user inputs the query word "abacus", then the query will be expanded with help of above tables. The overall search words for the expanded query term is

"abacus": "abacus", "calculator", "tablet"

The information available in the documents related to this terms and query term will be retrieved.

3.4 Similarity Measure

Semantic similarity measure plays an important role in information retrieval Processing. It finds the similarity between two candidate terms. It computes the similarity between conceptually similar terms, not necessarily lexically similar terms. It will be very useful to retrieve relevant information and to satisfy the user. The proposed work uses N-gram approach to find semantic similarity between two words.N-grams Technique mainly used to find the probability of sentence. The proposed N-gram technique employs bigram method for finding similarity between words using equation 2.

$$S = \frac{2n_t}{n_x + n_y} \qquad (2)$$

Where **n_t** - indicates the number of bigrams common in both string, **n_x** is the number of bigrams in 1st string and **n_y** is the number of bigrams in 2nd string. The following simple walkthrough example illustrates how to calculate word similarity with the help of bi-gram technique. Consider two words *cricket* and *cricketer.*

The following simple walkthrough example illustrates how to calculate word similarity with the help of bi-gram technique. Consider two words *cricket* and *cricketer.*

Let x= "cricket" and y= "cricketer" .

Then,

Bigrams of x = {cr,ri,ic,ck,ke,et} = 5 bigrams
Bigrams of y ={cr,ri,ic,ck,ke,et,te,er}=8 bigrams

In the above bigram conversion, the term x contains 5 bigrams and y contains 8 bigrams.

$n_t = \{x \cap y\} = 6 \qquad n_x = 6 \qquad n_y = 8$

$$S = \frac{2(6)}{6+8} = 0.857$$

Finally, the similarity between the terms x and y is 0.857.

4. Ranking and Retrieval

In the proposed system, the user query term is compared to the terms in the Inverted Index table using the Similarity Measure Technique. Based on the Similarity Measure value the document is divided into two groups. One group has terms with higher similarity value and the other with medium similarity value. The existing system returns only those results that match the query exactly, whereas the proposed system returns the terms that exactly match as well as those are relevant to the query.

The documents are ranked according to the term weight. The Similarity measure value range will be 0 and 1. If the Similarity value within 0.7 to 1.0 range, then retrieve the sentences and documents that are considered as exact matching to query. If the Similarity value within 0.4 to 0.6 ranges then retrieves the sentences and documents which are relevant to user query. If the user gives a query "cricket", then the query will be expanded using synonyms, antonyms, hypernyms and hyponyms .The similarity between these expanded queries will be calculated with indexed terms. The similarity value between the queries terms with indexed terms are shown in Table 5.

Table 5. Similarity Values of Query term with Indexed Terms

Query Term	Inverted Index Terms	Similarity Value
Cricket	cricket	1.0
Cricket	ball	0.02192020
Cricket	Bat	0.25
Cricket	kick	0.44444445
Cricket	foot	0.00110101
.	.	.
Cricket	scor	0.28571431
Cricket	goal	0.11212122

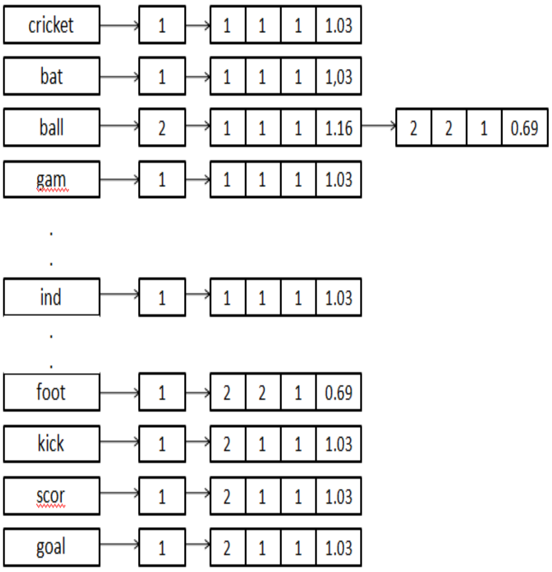

Figure 6. User Query Matching Using Similarity Value

5. PERFORMANCE EVALUATION

The model of the system has been developed using Net Beans as a front end tool and SQL Server as a backend data sources. In order to strengthen the experiment, the standard documents are collected from www.Wikipedia.com source and also some collections from www.uc.dataset.org to create benchmark corpus. This system is very effective because it retrieves both exact matches and relevant matches based on the user query with the help of query expansion technique. It fetches and provides information to user from most likely information to least likely information with the help of document ranking method. Multiple terms in query also retrieves information easily and effectively with the help of enhanced inverted index. Stemming Technique reduces the size of the inverted index. The searching time will be reduced because unique or root words can be obtained by Paice / Husk stemming algorithm. The indexing time of proposed inverted indexing increases when the system founds more unique terms in the document as in figure 7. Otherwise it simply increments the frequency count of the term.

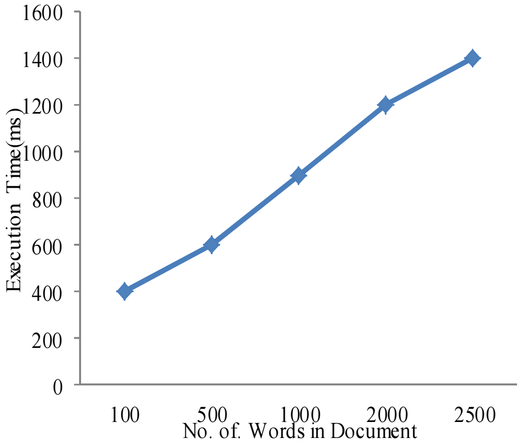

Figure 7. Indexing Time

The performance of the information retrieval system and amount of information it retrieve is better when we apply query expansion than retrieval of information without query expansion as in figure 8. Figure 8. Information Retrieval without Query Expansion and With Query Expansion

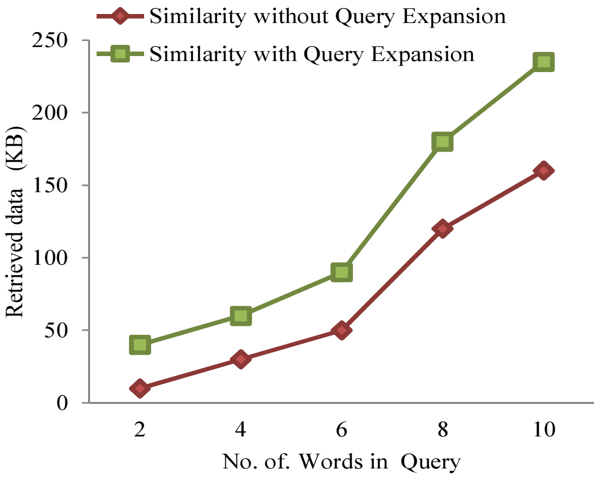

Figure 8. Information Retrieval without Query Expansion and With Query Expansion

The response time of the system is good. It is increased based on the number of documents in the corpus as in figure 9 and also frequency of occurrence of the query and relevant term in corpus.

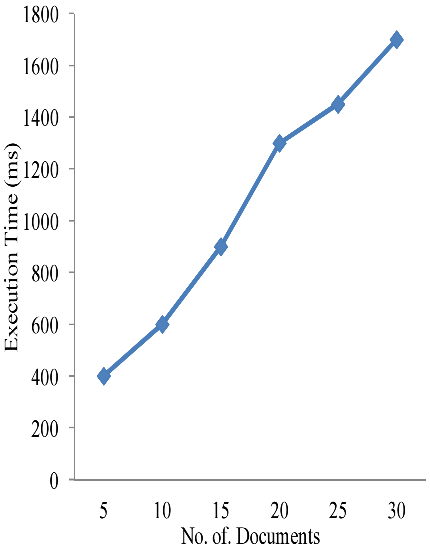

Figure 9. Retrieval Time. Of Proposed Method

6. CONCLUSTION AND FUTURE WORK

This paper presents an efficient information retrieval system which retrieves the information exact to the user query as well as most relevant to the query. The advantage of proposed inverted index is that the weight of each term is computed and maintained in index structure itself. It reduces the time to calculate weight of the term later as separate process. Query expansion is performed on user query that helps to find the exact matching documents and relevant matching documents. So, there is no false retrieval of information too. The bigram approach in our method reduces the similarity computation time and simple to do it. The ranking procedure improves the accuracy information retrieval process. Ontology based information retrieval also possible to implement in future along with this work. Our focus will be on improvement of searching time and retrieval time in future.

REFERENCES

[1] Deepa Gupta, Rahul Kumar Yadav , Nidhi Sajan, " Improving Unsupervised Stemming by using Partial Lemmatization Coupled with Data-based Heuristics for Hindi" International Journal of Computer Applications , Volume 38, No.8,pp.1-8, 2012.

[2] Zhisheng Li et al "An Efficient Index for Geographic Document Search", Knowledge And Data Engineering, Vol. 23, No. 4, pp. 585-599,2011.

[3] Wee Keong Ng, Yonggang Wen, Huafei Zhu ,"Private Query Processing on Inverted Index", IEEE Latin-American Conference, Brazil,pp.1-5, 2011.

[4] Bhavya Sukavasi et al, "An Application of Statistical indexing for Searching and Ranking of documents", International Journal of Computer Applications, Volume 28, No.3, pp.22-27, 2011.

[5] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka, , "A Web Search Engine-Based Approach to Measure Semantic Similarity between Words", Knowledge And Data Engineering, Vol. 23, No. 7, pp. 977-990 2011.

[6] Jia-Ling Koh and Nonhlanhla Shongwe, Chung-Wen Cho , "A Multi-level Hierarchical Index Structure for Supporting Efficient Similarity Search on Tag Sets " , RCIS,6th International Conference, Spain, pp.1-12, 2012.

[7] B. Barla Cambazoglu and Cevdet Aykanat," Performance of query processing implementations in ranking-based text retrieval systems using inverted indices", Information Processing and Management, Vol.42, No.4, pp.875-898, 2006.

[8] Ping-I Chen , Shi-Jen Lin, Ya-Chi Chu , "Using Google latent semantic distance to extract the most relevant information ",Expert Systems with Applications ,vol.38,No.6,pp. 7349-7358,2011.

[9] Benoit Gaillard, et al , "Query Expansion for Cross Language Information Retrieval Improvement", IEEE Conference,RCIS, France,pp.337-342 ,2010.

[10] Xiang Lian and Lei Chen , "Ranked Query Processing in Uncertain Databases", Knowledge And Data Engineering, Vol. 22, No. 3, pp. 420-436,2011.

[11] Dekang Lin , "An Information-Theoretic Definition of Similarity" , International Conference on Machine Learning, pp. 296-304,Canada,1998.

[12] Syed Raza Ali Rizvi, Shawn Xiong Wang , "Using Semantic and Structural Similarities for Indexing and Searching Scientific Papers", Computer Science and Automation Engineering, Vol. 26, pp.133-137,2011.

[13] Giuseppe Amato, Paolo Bolettieri, Fabrizio Falchi, Claudio Gennaro, Fausto Rabitti , "Combining Local and Global Visual Feature Similarity using a Text Search Engine", International Workshop on Content-Based Multimedia Indexing, pp.49-52,Spain, 2011.

[14] Mauro Dragoni, Célia da Costa Pereira, Andrea G.B. Tettamanzi , "A conceptual representation of documents and queries for information retrieval systems by using light ontologies", Expert Systems with Applications,Vol. 39,No.12,pp. 10376–10388,2012.

[15] Pawan Goyal et al, "Query Representation through lexical Association for Information Retrieval", IEEE, Knowledge and Data Engineering, Vol.24, N0.12, pp.2260-2273, 2012.