Ontology driven Pre and Post Ranking based Information Retrieval in Web Search Engines

Parul Gupta
Department of Computer Science and Engineering
Y.M.C.A. University of Science and Technology
Faridabad
parulgupta_gem@yahoo.com

Dr. A.K.Sharma
Department of Computer Science and Engineering
Y.M.C.A. University of Science and Technology
Faridabad
ashokkkale1@rediffmail.com

***Abstract* - With the tremendous growth of World Wide Web, it has become necessary to organize the information in such a way that it will make easier for the end users to find the information they want efficiently and accurately. This requires a pre-ranking of the underlying similar documents after the formation of the index. Thereafter the ranking of the search results in response to a query takes place which provides relevant results to user. This paper proposes an ontology driven pre ranking of the documents with identical context and hence post ranking of the search results using keyword matching of the expanded query terms and document keywords in the pre-ranked search results.**

Keywords- query expansion, ontology, ontology repository, context, domain, sub domain

I. Introduction

The ever-increasing amount of useful information on the web requires techniques for effective search and retrieval. A major problem is that users can be easily overwhelmed by the amount of information available. The transfer of irrelevant information in the form of documents retrieved by an information retrieval system and that are of no use to the user simply wastes network bandwidth and frustrates users. This condition occurs due to improper and time consuming ranking of the documents after the documents have been retrieved by the searcher from the index on the basis of keyword matching. Another reason for this loss of information and retrieval of irrelevant information is the inability of users to express their queries efficiently and accurately. Therefore, the major issue to be addressed in information selection is the development of a search mechanism that will help in getting maximum relevant documents. The possible solution to this problem can be ranking of documents beforehand and better ranking of the documents retrieved in response to the query. In the current scenario, the documents are retrieved if they contain keywords specified by the user. However, many documents contain the desired semantic information, even though they do not contain the user-specified keywords. Ontology is a technique that can be applied to extract the domain and sub domain of the specific keywords. So if once the documents have been searched on the basis of keyword matching, then the ranking can be performed by matching the keywords in the pre ranked retrieved documents and the keywords in the expanded query fired by the user and hence calculating the final rank. The pre ranking of the documents is done on the basis of the documents stored under the similar context. The extracted rank is stored in the context based index and hence we name it ranked context based index. The ontology repository can be used to extract the keywords in the domain and the sub domain of a particular context.

II. Related Work

A literature survey shows that ontologies have been employed to achieve better precision and recall in text retrieval systems. Query expansion has improved the effectiveness of ranked retrieval by automatically adding additional terms to a query. Guarino et al. [2] has attempted to perform query expansion through the use of semantically related terms and the use of conceptual similarity measures to find document similarity. The paper has tried to perform query expansion with a generic ontology WordNet that has been shown to be potentially relevant to enhanced recall as it permits matching a query to relevant documents that do not contain any of the original query terms.

Clerkin [3] used concept clustering algorithm (COBWEB) to discover automatically and generate ontology. They argued that such an approach is highly appropriate to domains where no expert knowledge exists, and they propose how they might employ software agents to collaborate, in the place of human beings, on the construction of shared ontologies.

The work in [4] chooses expansion terms from past user queries directly, rather than using them to construct sets of full text documents from which terms are then selected. The method consists of three phases: ranking the original query against the collection of documents; extracting additional query terms from the highly ranked items; then ranking the new query against the collection. The results show relative improvements over unexpanded full text retrieval of 26%–29%.

III. Proposed Work

A. Architecture of Ontology driven Ranking System

The current paper proposes an algorithm for pre ranking and post ranking in web search engines by query ontology matching with the keywords of the retrieved documents in response to the user query. The following figure presents the architecture of the ontology driven ranking system in search engines.

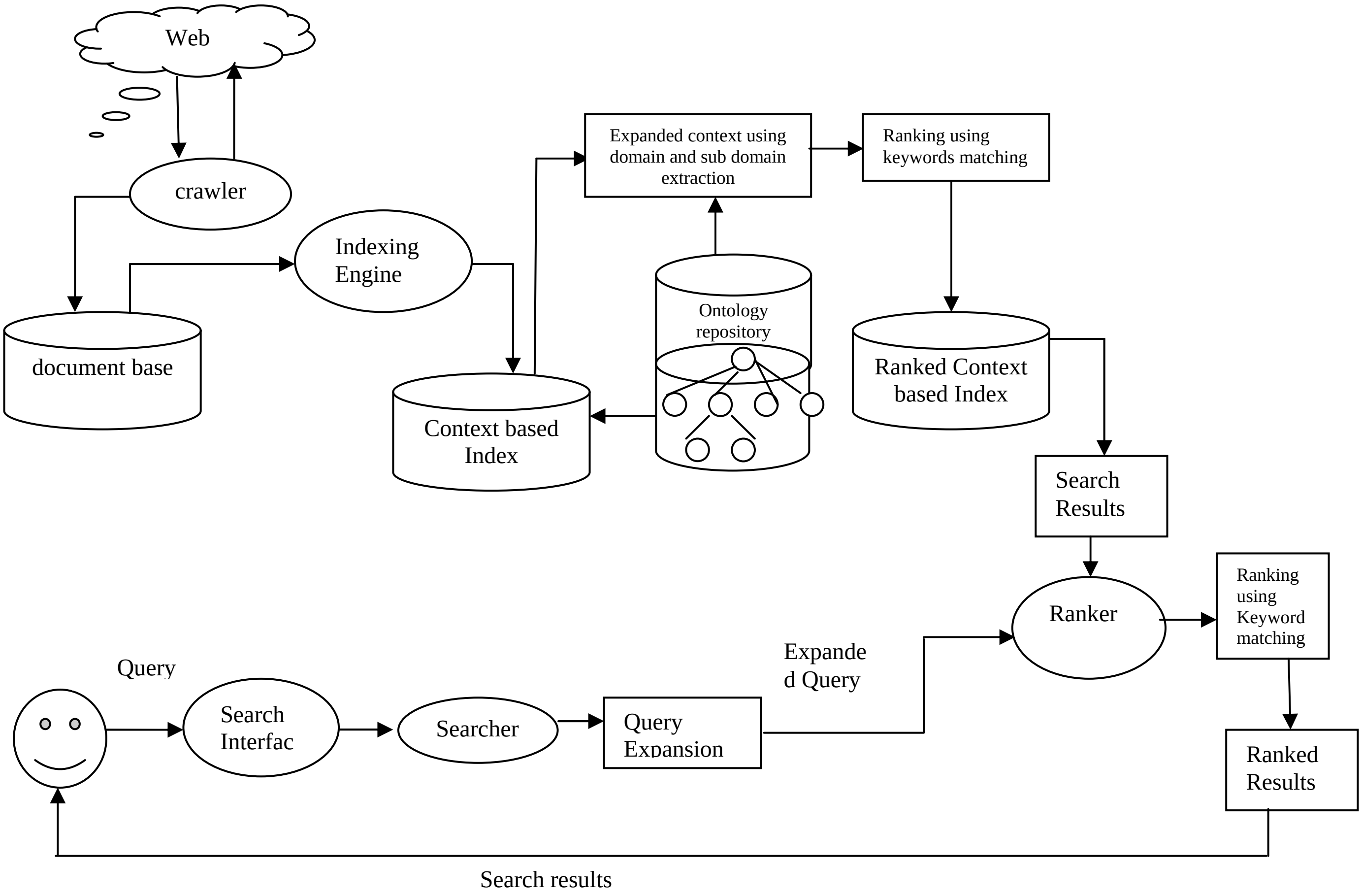

Figure 1. Architecture of Ontology driven Ranking System

The search engine provides results in response to the user queries. The query posed by the user is matched against the terms in the index and hence the relevant documents are retrieved in response to the user's query. However the returned documents may contain different order of relevance with the query. Thus the returned results are ranked according to their relevance with the user's request. So to attain the maximum level of the document relevancy ranking, ontology is applied to the ranking procedure. Moreover a pre ranking of documents done beforehand may be useful in reducing the time taken to rank the documents after being searched in response to user query. The pre ranking is done by matching the keywords in the documents under the same context with the terms extracted from the domain and sub domain of the context of the document. The query fired by the user is expanded using the query expansion module.Query expansion (QE) is the process of reformulating a seed query to improve retrieval performance in information retrieval operations. In the context of web search engines, query expansion involves evaluating a user's input (what words were typed into the search query area, and sometimes other types of data) and expanding the search query to match additional documents. Query expansion involves techniques such as searching for the synonyms as well finding all the various morphological forms of words by stemming each word in the search query. It also deals in fixing spelling errors and automatically searching for the corrected form or suggesting it in the results. The terms in the expanded query are then matched one by one with documents keywords and the document with the highest matching value is being ranked the highest. In this way, the post rank of the documents is calculated and hence final rank is calculated as the sum of pre rank and post rank.

B. Description of various modules in the architecture

1) Context based Index

This is the indexing structure that contains the context and hence the document identifiers of the documents belonging to that context. This context represents the theme of the document that has been extracted using context repository, thesaurus and ontology repository [7]. This is the final index that is constructed after extracting the context of the document. Rather than being formed on the term basis, the index is constructed on the context basis with context as first field and finally the document identifiers of the relevant documents.

2) Ranked Context based Index

This is the expanded indexing structure that contains the context, document identifiers of the documents relevant to the context as well as an added pre rank that has been computed using the ontology repository by extracting the terms in the domain and subdomain of the considered context.

3) Query Expansion Module

Query expansion (QE) is the process of reformulating a seed query to improve retrieval performance in information retrieval operations. In the context of web search engines, query expansion involves evaluating a user's input (what words were typed into the search query area, and sometimes other types of data) and expanding the search query to match additional documents. Query expansion involves techniques such as searching for the synonyms as well finding all the various morphological forms of words by stemming each word in the search query. It also deals in fixing spelling errors and automatically searching for the corrected form or suggesting it in the results. The algorithm for the query expansion is given below. It is being assumed that query log has i number of queries.

```
Algorithm Query_Expansion ($q_{i+1}$)
Input:  User Query $q_{i+1}$
Output : Search Results
begin
Qlog $_{i+1}$ =  Qlog$_i$   U   $q_{i+1}$
Cluster (indexed_doc)    // will return clus_doc
Association_Rule_Mining ( clus_doc)   // will return rules
for each new query $q_{i+1}$
Association_Rule_Mining (Qlog $_{i+1}$ )   // will return rules
KB <- rules
 End for
Query <- “$q_{i+1}$  + Extractd_Rules”   // query expansion
Conj_query <- “Query Λ semantics” // adding semantics
Search (Conj_query) // semantic expanded query to the
                                //searcher
end
```

Figure 2. Algorithm for Query Expansion

In the above mentioned algorithm, the query log provides detailed and summary information about queries. The query log lists the time that each search occurred, the IP address of the web user performing the search, the number of hits for the search, and the user's query. For URL clicko vers, it displays the query instead of the number of hits and the actual URL instead of the query. KB is the knowledge base which is a repository of extracted rules that have been derived using the association rule mining. A knowledge base containing rules is shown in the figure.

R1 : automobile → car
R2 : automobile → engineering
R3 : automobile → repair
R4 : ………… → ………..
R5 : ………… → ………..
R6 : ………… → ………..

Figure 3. Knowledge Base

4) *Ontology repository*

This is a database of ontologies which contains the various relationships among objects in various domains. Ontology repository contains various concepts with their relationships. This repository is used by the indexer to expand the stored contexts and also by the query expansion module to extract the context information from it.

An ontology can be defined as a formal explicit specification of a shared conceptualization. It is a formal and declarative representation which includes vocabulary for referring to the terms in that subject area and logical statements that describe the relationships among the terms. It also provides a vocabulary for representing and communicating knowledge about some topic and the relationships that hold among the terms in that vocabulary. Ontologies are used across a number of domains. Ontologies often contain a model of a domain, its taxonomy the relationships between its entities. Context Ontology defines a common vocabulary to share context information in a pervasive computing domain. For example: Figure1 depicts a simple ontology for apple consisting of a set of concepts C_{apple} = {apple, computer device, fruit, eatable, iphone} and a set of relationships R_{apple} = {brandname_of (apple, iphone), type_of (apple, fruit)}. Superclass_of represents the taxonomic relationship.

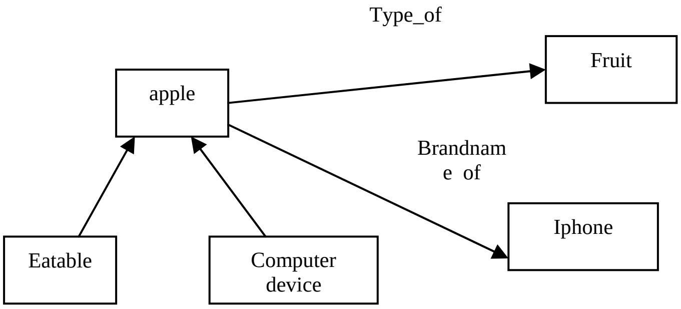

Figure 4. An Ontology Representation

C. *Proposed Algorithm for Pre Ranking of documents*

Algorithm(Pre_rank)

{

- Let D= {D_1, D_2,……D_n) be the collection of n textual documents stored in the context based index under the same context. Each document D_i can be represented by a corresponding set S_i such that S_i is a set of all the terms contained in D_i . Let us denote that set by D* such that D*={S_1, S_2,……….. S_n}.
- The terms in the domain and sub domain of the context are extracted using ontology repository and stored in array C.
- Now consider each document D_i and calculate the similarity of the document with the terms in the array C using the following similarity measure:
- Similarity_value[i] = $| S_i \Lambda C | / | S_i \cup C |$.
- Now the document i with the highest similarity value is assigned the highest rank and so on.
- This rank is stored in the context based index as pre rank and this index is termed as ranked context based index.

}

D. *Proposed Algorithm for Ontology driven Ranking System*

Algorithm Post_rank

{

- The query is accepted from the user.

- The next step is to expand the query.
- All the keywords in the expanded user query are stored in the array A[]
- Let D= {D_1, D_2,……D_n) be the collection of n textual documents retrieved after the keywords matching in response to the user query. Each document D_i can be represented by a corresponding set S_i such that S_i is a set of all the terms contained in D_i . Let us denote that set by D* such that D*={S_1, S_2,……….. S_n}.
- Now consider each document D_i and calculate the similarity of the document with the terms in the array A using the following similarity measure:

  Similarity_value[i] = $| S_i \Lambda A | / | S_i U A |$
- Now the document i with the highest similarity value is assigned the highest rank.
- Hence the documents are ranked in decreasing order of their similarity values stored in the array similarity_value and this rank is termed as post rank.
- The final rank is calculated as the sum of pre rank and post rank.
- Hence the documents are provided to the user ranked in order of their final rank.

}

E. Example Illustrating Pre ranking of the documents

Let us consider 4 documents D1, D2, D3, D4 related to the context “Indexing in Search Engines” with D1 based on *Indexing using pipeline architecture*, D2 based on *Indexing using distributed computing*, D3 based on *Parallel Indexing in Search Engines* and D4 based on *Indexing using clustering in Search Engines*. Now when the context based index is formed, its structure will be as shown in figure.

Context of the document	**Document Identifiers**
Indexing in search engines	1, 2, 3, 4
Crawling in Search Engines	6, 8…
…….	………

Figure 5. Context based Index

Now the domain and sub domain of the context “Indexing in Search Engines” is extracted using ontology repository and stored in array C as per the algorithm. Now pre rank is calculated by keyword matching of the documents 1, 2, 3, 4 with the words in the array C using the similarity measure Jaccard Measure and hence the all the documents are given rank with highest being assigned to the document which has maximum similarity with the array C. Now we get the ranked context based index as shown in figure.

Context of the document	**Document Identifier**	**Pre Rank**
Indexing in Search Engines	1	2
Indexing in Search Engines	2	1
Indexing in Search Engines	3	3
Indexing in Search Engines	4	4
……	..	..
…….	..	..

Figure 6. Ranked Context based Index

Now let us suppose that the user fires the query “Pipeline Indexing in Search Engines” with the context *Indexing in Search Engines*, then first of all the keywords in the domain and sub domain of the fired query are extracted *using* the ontology repository and stored in array A as per the algorithm and hence keyword matching is done with the documents extracted as search results of this query. The document with the highest similarity is assigned the highest rank. This rank is being termed as the post rank. The final rank of the document is calculated as sum of pre rank and post rank and the documents are provided to the user in order of their final rank.

In reference to the above example, documents 1, 2, 3, 4 are extracted as search results for the user fired query and now their final rank is calculated as shown in figure.

Document Identifier	**Pre Rank**	**Post Rank**	**Final Rank**
1	3	3	6
2	1	4	5
3	2	2	4
4	4	1	5
.			

Figure 7. Diagram representing post rank, pre rank, final rank

Hence the document having the maximum relevance to the user query is doc 1 which otherwise is ranked 3 rather than being highest ranked. So the calculation of the final rank leads to better ranking of the retrieved search results in response to the user query.

IV. Conclusion

The proposed solution overcomes the shortcomings of the existing ranking algorithms. It uses available contextual information and ontologies to rank the underlying documents as well as the search results. The use of contextual information results in better ranking of the documents and hence results in higher quality of the retrieved results. A context ontology is utilized to resolve inconsistent vocabularies in knowledge sharing and rule merging.

References

[1] Patel C et al, 2003. Ontokhoj: A semantic web portal for ontology searching, ranking, and classification. In Proc. 5th ACM Int. Workshop on Web Information and Data Management, New Orleans, Louisiana, USA, pp. 58–61.

[2] Guarino N, Masolo C,VetereG(1999), “OntoSeek: content-based access to the Web” IEEE Intell Sys 14(3):70–80.

[3] Clerkin, P., Cunningham, P., and Hayes, C., “Ontology Discovery for the Semantic Web Using Hierarchical Clustering” , Trinity College Dublin, Ireland, TCD-CS-2002-25.

[4] B. Billerbeck, F. Scholer, H. E. Williams, and J. Zobel. “Query expansion using associated queries” pages 2–9, NewOrleans, USA, 2003.

[5] Robertson S E. On term selection for query expansion. *Journal of Documentation,* 46, 359- 364, 1990.

[6] Min Song , Il-Yeol Song , Xiaohua Hu, Robert B. Allen, “Integration of Association Rules and Ontology for Semantic-based Query Expansion”, MIT Press,2006.

[7] R. Baeza- Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison Wesley, 1999.

[8] Dr. A.K.Sharma, Parul Gupta. Context based Indexing in Search Engines using Ontology.

AUTHORS PROFILE

Parul Gupta received the B.E (CSE) and M.Tech (Computer Engineering) degrees with Hons. from Maharshi Dayanand University in 2002 and 2005 respectively. Presently she is working as a Asst. Professor in Computer Engineering Department at YMCA Institute of Engineering, Faridabad. She is also pursuing her PhD in Computer Engineering and her interest spans the area of Indexing in Search Engines.

Prof. A. K. Sharma received his M.Tech. (Computer Science & Technology) with Hons. from University of Roorkee in the year 1989 and Ph.D (Fuzzy Expert Systems) from JMI, New Delhi in the year 2000. From July 1992 to April 2002, he served as Assistant Professor and became Professor in Computer Engg. at YMCA University of Science & Technology, Faridabad in April 2002. He obtained his second Ph.D. in IT from IIIT & M, Gwalior in the year 2004. His research interests include Fuzzy Systems, Object Oriented Programming, Knowledge representation and Internet Technologies.