©2010 International Journal of Computer Applications (0975 – 8887)
Volume 1 – No. 14

Context based Indexing in Search Engines using Ontology

Parul Gupta
Lecturer
Y.M.C.A. Institute of Engg., Faridabad

Dr. A.K.Sharma
Prof. & Head
Y.M.C.A. Institute of Engg., Faridabad

ABSTRACT
Indexing in search engines has been an active area of current researches. The main aim of search engines is to provide most relevant documents to the users in minimum possible time. So granting efficient and fast accesses to the index is a major issue for performances of Web Search Engines. Indexing is performed on the web pages after they have been gathered into a repository by the crawler. The existing architecture of search engine shows that the index is built on the basis of the terms of the document and consists of an array of the posting lists where each posting list is associated with a term and contains the term as well as the identifiers of the documents containing the term. The current information retrieval systems use terms to describe documents and search engines. This paper proposes an indexing structure in which index is built on the basis of context of the document rather than on the terms basis using ontology. The ontology-based collection selection method presented in this paper uses context to describe collections and search engines. The context of the documents being collected by the crawler in the repository is being extracted by the indexer using the context repository, thesaurus and ontology repository and then documents are indexed according to their respective context.

General Terms
Algorithm, Theory, Performance.

Keywords
Context, indexing, posting list, context repository, ontology repository.

1. INTRODUCTION
The internet contains hundreds of thousands of electronic collections that often contain high quality information. The basic aim is to select the best collection of information for a particular information need. The indexing phase of search engines can be viewed as a Web Content Mining process. Starting from a collection of unstructured documents, the indexer extracts a large amount of information like the list of documents, which contain a given term. It also keeps account of number of all the occurrences of each term within every document. This information is maintained in an index, which is usually represented using an inverted file (IF). IF is the most widely adopted format for this index due to its relatively small size occupancy and the efficiency involved in resolution of the keywords based queries. The index consists of an array of the posting lists where each posting list is associated with a term and contains the term as well as the identifiers of the documents containing the term. The term based index seems to be less efficient due to two information retrieval problems: polysemy (means a word has multiple meanings) and synonymy (means that multiple words having the same meaning). Thus the significance of term for building the index is reduced and the emphasis is laid on the context of the document. Context provides extra information to help improve search result relevance. The context of a document cab be easily derived using the concept of ontology.

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.

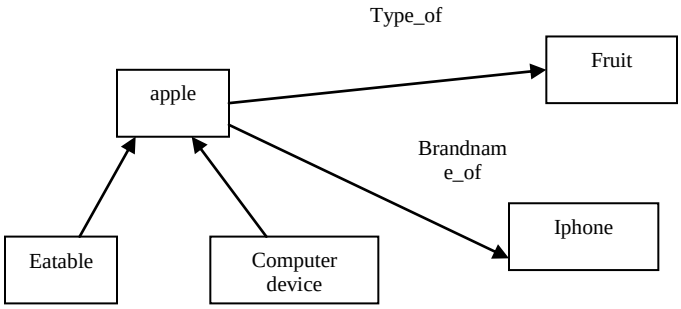

Figure 1 An Ontology Representation

2. RELATED WORK
In this paper, a review of previous work on index organization is given. In this field of index organization and maintenance, many algorithms and techniques have already been proposed but they seem to be less efficient in efficiently accessing the index.

In [12], the authors introduce a double indexing mechanism for search engines based on campus Net. The Campus Net Search

Engine (CNSE) is based on full-text search engine, but it is not general full-text search engine as it is basically a private net. The CNSE consists of crawl machine, Chinese automatic segmentation, index and search machine. They proposed double-indexing mechanism, which means, it has document index as well as word index. The so-called document index is based on the documents do the clustering, and ordered by the position in each document. In the retrieval, the search engine first gets the document id of the word in the word index, and then goes to the position of corresponding word in the document index. Because in the document index, the word in the same document is adjacent, the search engine directly compares the largest word matching assembly with the sentence that users submit. The mechanism proposed by them seems to be time consuming as the index exists at two levels.

Another work proposed was the reordering algorithm [2] which partitions the set of documents into k ordered clusters on the basis of similarity measure. According to this algorithm, the biggest document is selected as centroid of the first cluster and n/k1 most similar documents are assigned to this cluster. Then the biggest document is selected and the same process repeats. The process keeps on repeating until all the k clusters are formed and each cluster gets completed with n/k documents. This algorithm is not effective in clustering the most similar documents. The biggest document may not have similarity with any of the documents but still it is taken as the representative of the cluster.

Another proposed work was the threshold based clustering algorithm [6] in which the number of clusters is unknown. However, two documents are classified to the same cluster if the similarity between them is below a specified threshold. This threshold is defined by the user before the algorithm starts. It is easy to see that if the threshold is small, all the elements will get assigned to different clusters. If the threshold is large, the elements may get assigned to just one cluster. Thus the algorithm is sensitive to specification of threshold.

The given paper discusses the context based indexing structure in which the index is being created on the basis of the context of the terms in the documents rather than on the basis of terms itself.

3. PROPOSED WORK

3.1 Architecture of Context based Indexing System

Search Engines processing steps are carried out by three distinct and cooperating modules. The *Crawler* gathers Web documents and stores them into a huge repository after being compressed. The spider follows hyperlinks across the web collecting information from HTML web pages. Every Web page has an associated ID number called document identifier, which is assigned whenever a new URL is parsed out of a web page. The *indexer* takes the web pages collected by the spiders and parses them into a highly efficient index. The current paper proposes an architecture for building context based index and hence performing context based searching as shown in figure2.

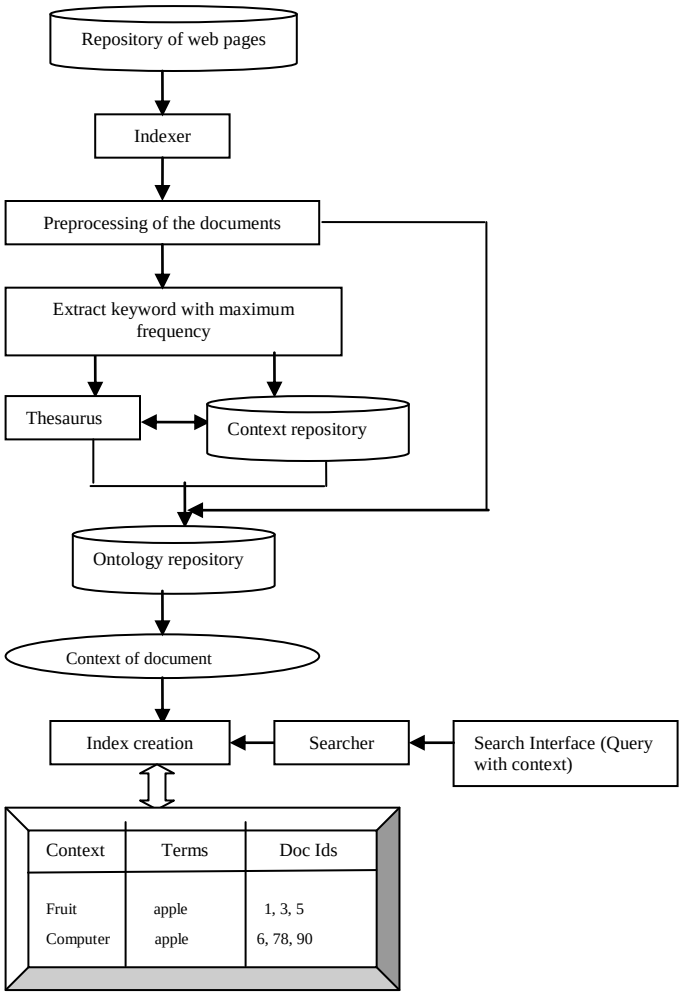

Figure 2 Architecture of Context based Indexing

3.2 Description of various Components

1. *Repository of web page.* This is the database which contains the set of documents that have been collected by the crawler.

2. *Indexer*. After the documents have been gathered by the crawler, the indexer maintains an index of the documents which is in the form of posting lists that contain the term as well the document identifiers of the documents which contain the given term and also other related information.

3. *Preprocessing of document.* The preprocessing step involves stemming as well as removal of stop words. A stop word is any word which has no semantic content. Common stop words are prepositions and articles, as well as high frequency words that do not help retrieval

4. *Thesaurus.* It is a dictionary of words available on the world wide web from thesaurus.com which contains the words as well as their multiple meanings.

5. *Context Repository*. This is a database which contains the various contexts. Also the new contexts derived from thesaurus are stored in this repository. The context repository maintains a database of several types of context data

6. *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.

7. *Context of the document.* This context represents the theme of the document that has been extracted using context repository, thesaurus and ontology repository.

8. *Index.* 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, term as next field and finally the document identifiers of the relevant documents.

9. *Searcher.* It is that module of the search engine that receives user queries via the user interface and hence after searching the results in the index provides them to the user.

10. *Search Interface.* It is that user interface through which user types the query along with the context specified.

3.3 Algorithm for Index Construction

The algorithm depicted in figure 3 shows the various steps in the construction of the context based index and hence context based searching.

1. The preprocessing of the documents which includes stemming and removing stop words is performed by indexer.
2. Once the document preprocessing is complete, the term with the maximum frequency matched with the title is extracted from the document
3. Now the maximum frequency keyword is being searched in the thesaurus (thesaurus can be taken online from thesaurus.com) and the context repository.
4. This step helps in extracting the context of the document but a keyword may have multiple contexts. So multiple contexts are extracted.
5. Now the next step is to extract the specific context of the document from these multiple contexts.
6. The multiple contexts and the terms of the document are compared with the ontology repository. Thus by matching the keywords of the document and the multiple contexts with the concepts and the relationship terms in the ontology repository, the context of the document gets extracted.
7. Now the posting list in the index consist of three columns, the one containing the context, the second one containing terms related to the context and the third one contains the lists of documents that contain the term with that specific context.
8. Now when the user fires a query with the context explicitly specified, then the index is being searched first on the context basis rather than on the term basis.
9. After the context is matched, the keywords in the query are matched with the terms related to that context in the index.
10. The document identifiers of the relevant documents are being picked up and the user is provided with best matching documents.
11. Thus this index provides a fast access to document contents and structure.

Figure 3 Proposed algorithm for construction of Context based Index

4. PERFORMANCE ANALYSIS

4.1 Individual Performance Analysis of Proposed and the Existing Indexing System

The performance of the context based indexing can be measured by computing the support value that can be calculated after the simulation of the current index structure. A rough estimate of the expected performance is shown in the figures 4 and 5.

Support = No. of relevant documents / No. of retrieved documents

Figure 4 shows the performance of the current indexing system when user puts the query "apple".

Figure 4 Existing System Performance

As per the above results, five out of the ten results seems to be relevant. So as per the above estimated results, for the existing system i.e. for term based indexing, the support can be calculated using formula given above as follows:

- Support = 5/10 = 0.5

Figure 5 shows the performance of the proposed context based indexing system when user puts the query "apple (iphone)".

Figure 5 Proposed System Performance

As per above results, for the proposed context based indexing system, eight out of ten results seems to be relevant, so the support value can be calculated using the formula given above as follows:

- Support = 8/10 = 0.8

4.2 Comparison of Performance of Proposed and Existing Indexing Systems

The performance of both current and the existing systems can be graphically visualized as in figure 6.

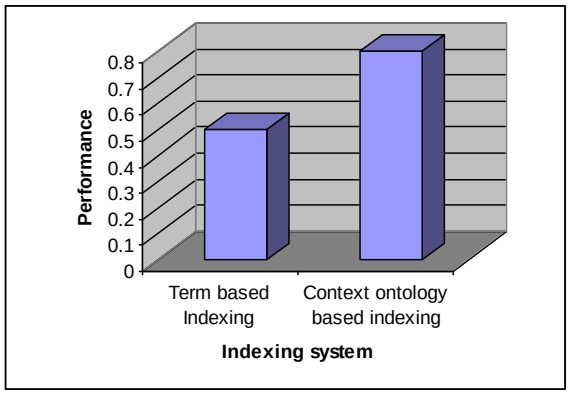

Figure 6 Graph showing performance of current and proposed Indexing system

5. CONCLUSION

This paper presents an indexing structure that can be constructed on the basis of the context of the document. The context of the document can be extracted by using thesaurus and ontology repository. So this paper uses ontology for context based index building. The context based index enables retrieval from index on the basis of context rather than keywords. This aids in improving the quality of the retrieved results. A rough estimate of support values for the existing and the proposed system clearly depicts the better performance of the existing system.

6. REFERENCES

[1] Parul Gupta, Dr. A.K. Sharma, A framework for multilevel Indexing in Search Engines, accepted in International Journal of Applied Engineering Research..

[2] Fabrizio Silvestri, Raffaele Perego and Salvatore Orlando. Assigning Document Identifiers to Enhance Compressibility of Web Search Engines Indexes. In the proceedings of SAC, 2004.

[3] S.Brin and L.Page.The Anatomy of a Large-Scale Hypertextual Web Search Engine. In the 7th International WWW Conference, (WWW7), pp. 107-117, Brisbane, Australia.

[4] Van Rijsbergen C.J. Information Retrieval. Butterworth 1979.

[5] Soumen Chakrabarti. Mining the Web Discovering Knowledge from Hypertext Data. Morgan Kaufmann Publications, 2003.

[6] Oren Zamir and Oren Etzioni. Web Document Clustering: A feasibility demonstration. In the proceedings of SIGIR, 1998.

[7] A. Jain and R. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988

[8] Berners-Lee, T., Hendler, J. and Lassila, O., "The Semantic Web," Scientific American.284(5):35-43, 2001.

[9] O. Zamir, O. Etzioni, O. Madanim, and R.M. Karp, "Fast andIntuitive Clustering of Web Documents," Proc. Third Int'l Conf. Knowledge Discovery and Data Mining, pp. 287-290, Aug. 1997.

[10] Wang Jicheng, Huang Yuan, Wu Gangshan and Zhang Fuyan, 'Web Mining: Knowledge Discovery on the Web' ,IEEE (1999).

[11] Frawley, W., Piatetsky-Shapiro, G., and Matheus, C., Knowledge Discovery in Databases: An Overview. Ai Magazine, Vol. 13 (1992), pp.57-70.

[12] Changshang Zhou, Wei Ding, Na Yang, Double Indexing Mechanism of Search Engine based on Campus Net, Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06)

[13] Quan, T. T., Hui, S. C., Fong, A. C. M., and Cao, T. H. (2004). Automatic generation of ontology for scholarly semantic Web. In: Lecture Notes in Computer Science. Vol. 3298. (pp. 726–740).