IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 12, Issue 4 (Jul. - Aug. 2013), PP 89-93
www.iosrjournals.org

Context Based Web Indexing For Semantic Web

Anchal Jain[1] Lecturer(JPIEAS)
Nidhi Tyagi [2] Asst. Professor(SHOBHIT UNIVERSITY)

Abstract : *A context based focused crawler downloads web pages that are more relevant for user query in syntax of context. Wherein downloaded web pages are indexed for providing the speed to search engine. This paper purposes a new indexing technique based on B+ tree that indexed the context along with ontology's of keywords. These keywords are extracted from the web documents that are stored in web repository. This purposed indexing technique increases the speed of search engine for finding the more relevant documents from semantic web*

Keywords *- Architecture, B+ Tree, Context, Semantic web, Web repository*

Submitted Date 14 June 2013 **Accepted Date: 19 June 2013**

I. INTRODUCTION

With the rapid growth of the Internet, the World Wide Web (WWW) has become one of the most important resources for obtaining information and one of the most important media of communication. Currently there are huge amounts of documents existing in the *World Wide Web*. Finding information from *WWW* according the user interest becomes a critical task. Modern web search engines can cache, index and search several billion of web pages, which only includes a small part of all existing documents in the Web. And even for this small amount, the search quality could not meet a user's requirements in many cases. Many ideas have been proposed to improve the web search quality, which can be measured with the following two metrics:

(1) Precision rate: The ratio of the number of relevant documents retrieved to the total number of documents retrieved.

(2) Recall rate: The ratio of the number of relevant documents extracted to the total number of relevant documents in the Web.

The purpose of storing an index is to optimize speed and performance in finding relevant documents for a search query. Without an index, the search engine would scan every document in the corpus, which would require considerable time and computing power. For example, while an index of 10,000 documents can be queried within milliseconds, a sequential scan of every word in documents is a time consuming task. The additional computer storage required to store the index, as well as the considerable increase in the time required for an update to take place, are traded off for the time saved during information retrieval[1].

In B+ tree all paths from the root to the leaf nodes are equal length .So this tree is called balanced tree. All data is stored at the leaf nodes (*leaf pages).* Leaf pages are linked to each other .B+ tree reduces the number of I/O operations required to find an element in the tree. Finding a record requires $\mathbf{O\ (Log_b n)}$ operations. This strategy is more beneficial for search engine.

II. Related Work

Here many algorithm & technique all ready purposed for indexer to achieve the indexing on documents for information retrieval. But they are not more efficient for search.

Nidhi Tyagi, R.P Agarwal [1] This paper proposes a technique for indexing [1] the keyword extracted from the web documents along with their contexts wherein it uses a height balanced binary search (AVL) tree, for indexing purpose to enhance the performance of the retrieval system.

P. Gupta and A. K. Sharma [2], worked on context based indexing in search engines using ontology. The index construction is done on the basis of the context using ontology. The context repository, thesaurus and ontology repository are used by the indexer to identify the context of the document.

C. Zhou, W. Ding and Na Yang [5], the paper introduces a double indexing mechanism for search engines based on campus Net. The CNSE consists of crawl machine, Chinese automatic segmentation, index and search machine. The proposed mechanism has document index as well as word index. The document index is based on, where the documents do the clustering, and ordered by the position in each document. During 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, seems to be time consuming as the index exists at two levels.
The critical look at the available literature reveals that there is a requirement for a technique to organize the keyword and their contexts in a better fashion as storing in a linear fashion makes searching of a document a bit time consuming.

III. Purposed work

This paper proposes an algorithm for indexing the keyword extracted from the web documents along with their context & ontology. The purposed indexing technique is a B+ tree, in addition to improved performance in the retrieval of information; this data structure is able to support dynamic indexing, which is especially important for environments where documents are changed frequently. If the planning about the arrangement of the keywords is done then B+ tree can be achieved. B+ tree algorithm & technique improve the efficiency of indexer for searching the documents from semantic web. This paper purpose a ontology based context indexing architecture in **fig 1**

3.1 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.

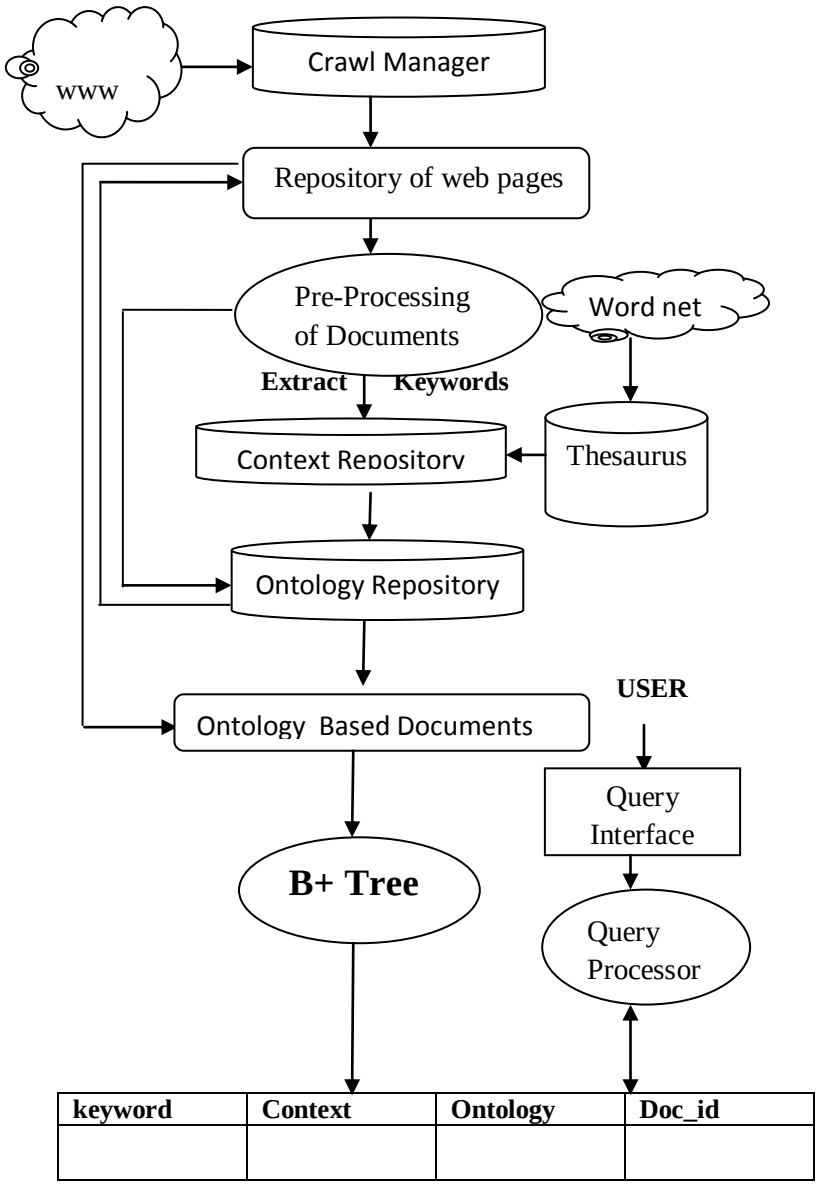


Fig 1 Architecture of Context Based Indexing.

2. 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
3. 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.
4. 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
5. Ontology Repository: This is a database of ontology's which contains the various relationships among objects in various domains. Ontology repository contains various concepts with their relationships.
6. Ontology based document: This context represents the theme of the document that has been extracted using context repository, thesaurus and ontology repository.
7. B+ Tree: this is the indexing technique that is constructed after extracting the context of the document on the basis of ontology.
8. Query Interface & query processor: It is that module of the search engine that receives user queries and hence after searching the results through query processor in the index provides relevant information to the user.

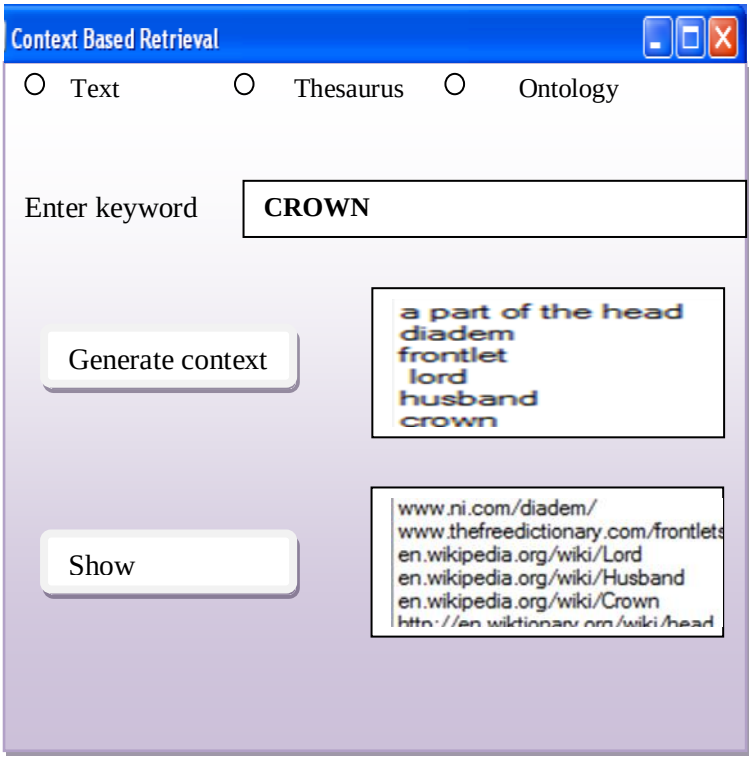
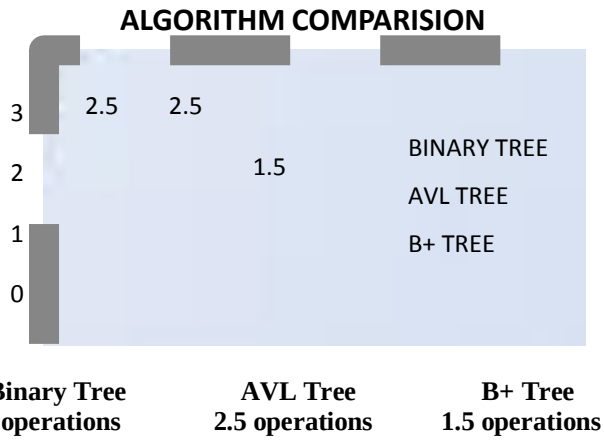


Fig 2 Query Retrieval Interface

In figure 2 the user entered keyword Crown & desired context of the keyword displayed through the generate context button, the corresponding related web page URLs are listed (available in the repository) displayed by pressing the show document button. This can help the user to directly access more related and relevant information.

3.2 Comparison of Performance of Proposed and Existing Indexing Algorithm



The purposed algorithm for indexing provides a fast access to document context and structure along with an optimized searching.

3.3 Proposed algorithm for the indexing scheme

Step1: Preprocess the crawled web documents and extract the keyword along with their frequency of occurrence.

Step 2: Input the keywords to the context generator which extracts the multiple contextual Sense of the word. Context is being searched in the thesaurus (a dictionary of words available on WWW from thesaurus.com, which contains the words as well their multiple meanings).

Step3: The keywords along with the context are indexed using the B+ tree.

Step4: Compare the entered keyword with the node's keyword field of tree, until a similar word is found. Corresponding document_id is stored? Context is being searched in the thesaurus (a dictionary of words available on from thesaurus.com, which contains the Words as well their multiple meanings).

Step5: If search is not a success, create a node containing the following fields (Left child, Keyword, right child, and link) .The link is pointer variable which points to the Database where the context of keyword stored along with its ontology based document_id.

Step6: Arrange the node in the B+ tree, according to the height BF.

Step7: Repeat step 4, 5 and 6 until all the extract keywords are arranged.

Step8: Now when the user fires the query with context explicitly specified, then the index is being searched, reducing its search time to half of the linear search.

Step9: Thus, B+ indexing technique provides a fast access to document context and structure.

IV. 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.

Future Scopes: Future scope of this system is that the B+ tree based indexing technique, is able to support dynamic indexing and improves the performance in terms of accuracy and efficiency for retrieving more, relevant documents as per the user's requirements since the context of the various keywords is also stored along with them. Thus, the indexing technique provides a fast access to document context and structure along with an optimized searching

References

[1]. Nidhi tyagi, Rahul Rishi ,R.P. Agarwal "**Context based Web Indexing for Storage of Relevant Web Pages"** *International Journal of Computer Applications (0975 – 8887) Volume 40– No.3, February 2012*

[2]. Parul Gupta and A.K.Sharma **"Context based Indexing in Search Engines using Ontology**", International Journal of Computer Applications, Volume 1 No. 14, pp 49-52, 2010.

[3]. Pooja Gupta , Dr. A K Sharma, J. P.Gupta, Komal Bhatia " **A Novel Framework for Context Based Distributed Focus Crawler (CBDFC)"** *Int. J. Computer and Communication Technology, Vol. 1, No. 1, 2009*

[4]. Naresh Chauhan and A. K. Sharma," **Design of an Agent Based Context Driven Focused Crawler"**, BVICAM'S International Journal of Information Technology, pp 61-66, 2008.

[5]. Changshang Zhou, Wei Ding and 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), 2006.

[6]. O. Zamir, O. Etzioni, O. Madanim, and R.M. Karp **"Fast and Intuitive Clustering of Web Documents,"** Proceeding Third International Conference Knowledge Discovery and Data Mining, pp. 287-290, Aug. 1997.

[7]. S. Chakrabarti, K. Punera, Mallena Subramanyam, **"Accelerated Focused Crawling through Online relevance feedback"**, paper presented at *WWW conference* December 2002.

[8]. Steve Lawrence, **"Context in Web Search**", IEEE Data Engineering Bulletin, 2000.

[9]. S. Chakrabarti, M. van den Berg, and B. Dom. **"Focused crawling: a new approach to topic-specific web resource discovery"**. In *WWW-8*, 1999.

[10]. **Word Net-Online dictionary** and hierarchical thesaurus Obtained through the Internet http://www.wordnetonline.com [accessed 28/12/2009].

[11]. Sajendra Kumar, Ram Kumar Rana ,Pawan Singh " **Ontology based Semantic Indexing Approach for Information Retrieval System"** *International Journal of Computer Applications (0975 – 8887) Volume 49– No.12, July 2012*