Search Optimization using Context based Search

Nidhi Jain
IIMT Engineering College
Meerut

Paramjeet Rawat, Ph.D
IIMT Engineering College
Meerut

ABSTRACT
Finding meaningful information among the billions of information resources on the web is a tedious task as the popularity of Internet is growing rapidly. The future of web is a structured semantic web in place of unstructured information present in the web nowadays. On semantic web, ontology is used to assign meaning to the content of the web. The main concern of focused crawling is to retrieve only the relevant pages rather to crawl all web pages. So the main issue is how to extract quality pages that is relevant to the topic. To overcome this problem, we have designed a focused crawler in which we first will construct ontology from the web repository then we will integrate this ontology with the semantic nets so that a focused document group can be created. After that will accept the keywords to be searched and make search more concise by pruning the unwanted data and display the results based upon that along with its related context with the help of tree structure that is created using the ontology designed by us.

General Terms
Focused crawler,ontology,semantic net,context seraching.

1. INTRODUCTION
Most of the web search engines of first generation crawlers [1] are heavily based on traditional graph algorithms, such as depth-first or breadth-first traversal, for web indexing. The recursive algorithm use a seed set URLs to crawl the whole web by using the hyperlinks down to other document.

The enormous growth of World Wide Web (WWW) is a continuously expanding large collection of hypertext documents [2] to represents a very huge distributed hypertext system which involves hundreds of thousands of individual sites. It is a client-server based architecture that allows a user to initiate search by providing keywords to a search engine, which in turn collects and returns the desired web pages from the web. Due to extremely huge amount of pages present on the web, the search engine depends upon crawlers to crawl required pages. To find several thousands of matches for an average query it has to maintain large number of web pages [3,4] by using current commercial search engines

Therefore, a search engine may present a list of thousands of web pages in response to user's particular keyword possibly consisting of irrelevant web pages also. The web search engines try to cover the whole web and serve queries concerning all possible topics [5]. In fact, from the user's point of view, it does not matter whether the search returned 10,000 or 50,000 hits because the number of matches becomes too large to sift, leading to the problem of information overkill.

Focused crawling [6,7,8] is used to improve the searching quality of web pages and aim to retrieve and search the subset of the WWW that pertains to a specific topic of relevance. Therefore, Focused crawler, provides a potential solution to the problem of information overkill. Different strategies are adopted by the existing focused crawlers [7, 4] to compute the words' frequency in the web documents. The document is considered to be relevant if higher frequency words match with the topic keyword otherwise document is irrelevant.

The focused crawling process consists of two interconnected cycles: First, the ontology cycle, and second, the crawling cycle. The human engineer mainly drives the first cycle. An instantiated ontology is targeted by the crawling form defines by the engineer. The output of the crawling process is provide by the cycle to the user in the form of a document list and proposals for enhancement of the already existing ontology to the user.

Ontologies are becoming the corner stone of the Semantic Web (SW).Ontologies aim at capturing domain knowledge in a generic way and provide a commonly agreed understanding of a domain. They are shared conceptualizations of a domain and they possibly include the representations of these conceptualizations [9]. They are used to facilitate efficient exchange of information among people. In an organization, the University Ontology is defined as the conceptualization of the Person which forms super class of either student class or staff class.

<UNIVERSITY_PERSON>

<UNIVERSITY NAME=XYZ>

<STAFF>

<ACADEMIC>

<PROFESSOR_NAME>NIDHI

<DEPT_NAME>CSE

<SUBJECT_TEACHING>DMS

</SUBJECT_TEACHING>

</DEPT_NAME>

</PROFESSOR_NAME>

<ASST_PROFESSOR></ASST_PROFESSOR>

<LECTURER> </LECTURER>

</ACADEMIC>

<NON_ACADEMIC>

<CLERK></CLERK>

<ACCOUNT_OFFICER></ACCOUNT_OFFICER>

</NON_ACADEMIC>

</STAFF>

<STUDENT>

<GRADUATE>

<DEGREE></DEGREE>

</GRADUATE>

<POST_GRADUATE>

<CLASS></CLASS>

</POST_GRADUATE>

</STUDENT>

</UNIVERSITY>

</UNIVERSITY_PERSON>

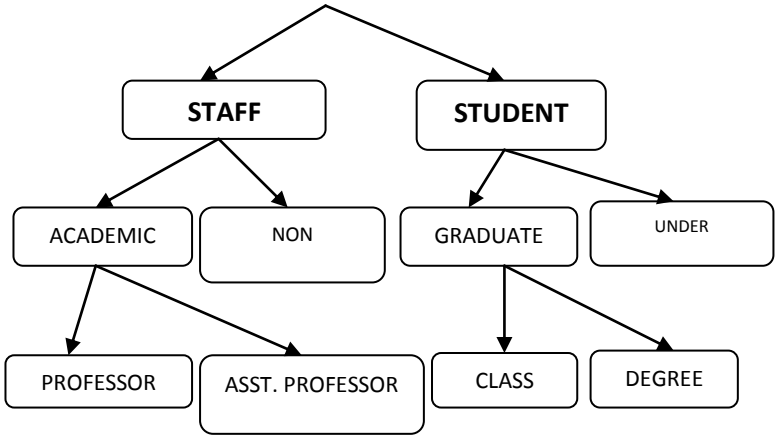

Figure 1. An example of ontology

The topical relevance is not the only issue for focused crawlers but context relevance should also be considered [10]. If the user issues one keyword then its relevant context must also be known. In this paper, the design of a *Context Driven Focused Crawler* (CDFC) is being proposed that provides the context of the keywords to the user in a flexible and interactive DOM tree [11]. DOM depicts a web page into a fine grained structure. Each node of DOM tree can be labeled as a block. The URL score is calculated based on topical relevancy of parent page block because we know that any user puts information about topics in parent page and all related information about topics refer to child pages.

The remainder section of the paper is organized as. In Section II we present the related work on the field of context based focused crawling and web document retrieval. In Section III we describe the architecture of the proposed framework . . In Section IV, we have presented our proposed approach. In Section V, our proposed algorithm has been represented and in Section VI, we have concluded our research paper.

2. RELATED WORK

Author et al.[13] proposed design of context driven determined crawler (CDFC) is based on the amplified hypertext document wherein the context of the keywords is stored in the form of TOC (Table of Contexts). The TOC coupled with a category tree provides context of the keywords. With the help of this design we can not only avoids the expensive complex computations for deriving the context of the user keywords but also reduces the network traffic appreciably. In addition, the quality of downloaded documents is in conformance with the topic and context of the user choice. Author et. al[14] recommend an effective approach——A focused web crawling based on web semantic analysis and web link analysis. This model is based on Formal Concept Analysis, using concept context graph analyze similarity between the web content and the users' interests. Using the content similarity forecast the similarity between the web links and the users' interests, the benefit of this methodology is that crawls the web pages which are related to users' interests, so that it can improves the efficiency and precision of the crawls greatly. An experiment illustrates that the new method is an valuable mechanism which has a huge result.[15] proposed a uniqueness .Focused Web Crawling strategy based on web semantics Analysis and web links analysis and considers the persuade about web pages' content and link relation to the priority of crawling ordering. It is make sure by Web semantics analysis that obtained web pages by crawler always belong to users' interested subjects, and solve topic shift problem capably. Web links analysis can really trim down the number of web pages that crawler need to download by pruning. of found url. In that way, it is possible to see enhancement of speed and efficiency of Focused Web Crawling. Investigational results show that Focused Web Crawling approach proposed to improve efficiency of precision metric and recall metric. This paper proposes [16] 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 collection selection method of ontology based uses context to describe collections and search engines. The crawler is used to collect the context based document in the repository which is being retrieved by the indexer using the context repository, ontology repository, thesaurus and then documents are indexed according to their respective context. It depicts the better performance of the existing system. Its main disadvantage is it is time consuming. David B. Leake[5]present a focused crawling algorithm that builds a model for the context which is based on focused crawling algorithm based on topically relevant pages occur on the web. Valuable pages of link hierarchies occur in this context model, as well as content on documents that frequently co-occur with relevant pages. The existing capability of large search engines is to provide partial reverse crawling. It shows significant performance to improve crawling efficiency over standard focused crawling. We found that such difficult categories are those where target content is not reliably co-located with pages from a different category, and where common hierarchies do not exist or are not implemented uniformly across different web-sites. It is therefore to be expected that the context graph provides less guidance in such cases. However, due to our architecture design, The major limitation of our approach is the requirement for reverse links to exist at a known search engine for a reasonable fraction of the seed set documents

3. PROPOSED MODEL

Frontier contains a list of unvisited URLs maintained by the crawler and is initialized with seed URLs. Web page downloader fetches URLs from frontier and downloads corresponding pages from the internet. We build a DOM tree. Each node of DOM tree can be labeled as a block. Our main objective is to find a most appropriate block. In order to facilitate analysis we have neglected those nodes which show the attribute of label and uses the cues provided by HTML mark-up tags such as tables, paragraphs, headings, lists, etc. Relevance calculator calculates relevance of a page with respect to the topic, and assigns score to URLs extracted from the block of the page. Topic filter analyzes whether the content of parsed block is related to topic or not. If the block is relevant, the URLs extracted from it will be added to the URL queue. Otherwise, the URLs will be ignored.

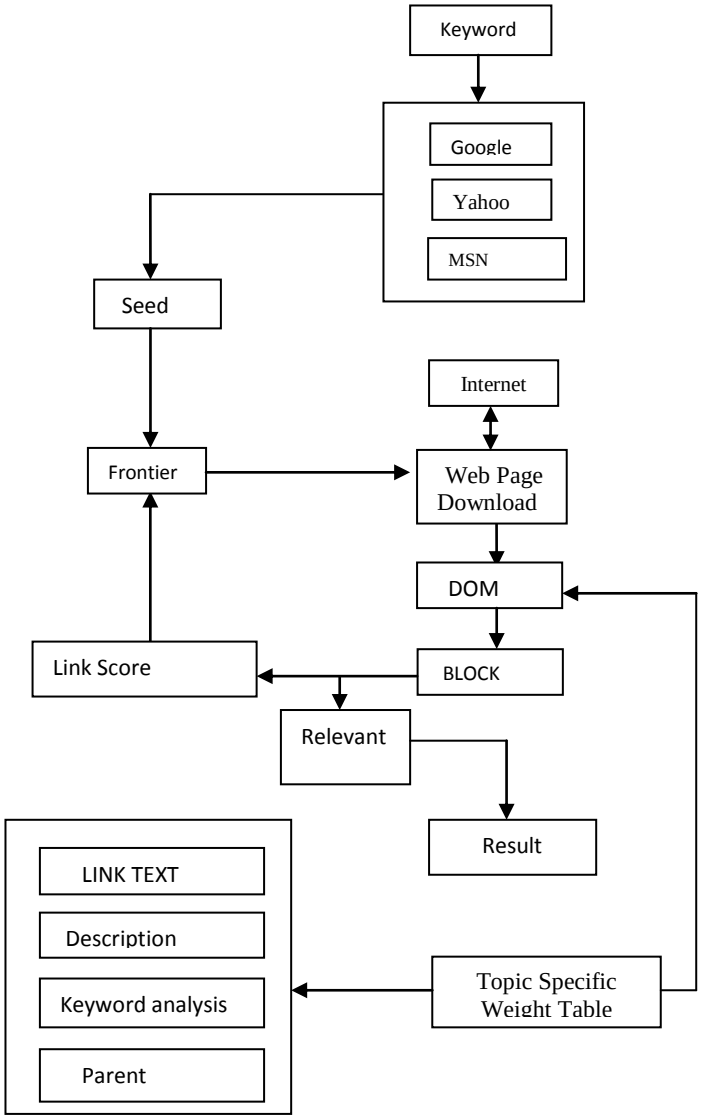

Figure 2. The architecture of focused crawler

4. PRPOSED APPROACH

A**. Seed URL Extraction :**Seed URLs are extracted by one search engine known aswww.threesearches.com. We put a query in this search engine and it shows the result of three most popular search engines like Google, Yahoo, and MSN search. We take resulting URLs which are common in all the three search engines. We assume that this common search result URLs are most relevant for this query and thus these URLs are the seed URLs.

B**. Frontier:** It is initialized by seed URLs. It contains only unvisited URLs. It uses the priority queue. A URL which has higher URLs score is given higher priority. The higher priority URL is input to the web page downloader.

C. **Web Page Downloader** :In our approach, web page downloader is used to take input URLs which has higher priority from frontier and downloads the web page from internet.

D**. Page Partition:** In a web page, texts and links about the same topic are often gathered into one region, which is called a content block. Content block partition is the process of partitioning web pages into blocks.

We define page partitioning algorithm in the following way:

```
WEB_P_g_BLOCK_PARTITION(P_g)
{
H_p= Parse TREE(P_g)
Initialize(H_P)
Q=root(H_P);
While(Q!=null)
{
x=Q[Front]        // where Front= 0

yHB= H_B(root,0) *d        //H_B is a block height
if(x[child]<=d && HB (x,0)>=y)
      Q=x[child];
else
      printf("x is a block");
}
}
H_B(Node root, int h)        // h is a height.
{
       count=h;
       If (root[child]!=NULL)
   child=root[child];
   l=l[child]                // l is a length of node
   htemp=0;
   ftemp=count;
 if(child==block[content]
   htemp=H_B(child, count+1)
 else
   htemp=H_B(child, count)
          if( htemp> ftemp)
                ftemp=htemp
          if( ftemp>count)
             count=ftemp;
       return count
}
```

E. Relevance Calculator It calculates relevance score of block with respect to topic and stores relevance score of blocks in relevance block database. When relevance calculation of all blocks is famished, then it goes to relevance calculation of page step. Otherwise, it again returns to relevance calculation step to calculate the relevance score of rest of the blocks in particular page.

F. Relevance Analyzer It analyses the relevance score of total blocks and calculates the summation of relevance score of all blocks which is the relevance score of page. A page which has relevance score greater than user specified limit, only that page is stored in Relevant page DB. Otherwise, the page is discarded. From Block DB, the block's URLs extract

only that block which has relevance score greater than 0.9. By experiment it has been proved that if the relevancy score between two pages is more than 0.9, then both the pages are similar to each other. Now, the URL score is calculated based on topical relevancy of parent page's block.

5. PROPOSED ALOGRITHM

STEP1: Extract seed pages from threesearches.com

STEP2:Extract all terms and links from the seed page ;

STEP3 :Form DOM tree for each web page.

STEP 4: First extract all the suitable blocks from the html DOM tree.

STEP5: Identify blocks in each parent page in which specific link exist.

STEP 6: Calculate the relevance score of parent page block with respect to topic.

/*calculate the relevancy score R(t,p) of each block with respect to relevant block*/

STEP 7:Calculate the weight of each topic table in terms block.

STEP8: Calculate relevancy _score _URL_description.

STEP9:Calculate Anchor_Relevance_Score

Step10:Calculate link_score(i).

6. CONCLUSION AND FUTURE WORK

Our methodology is mainly composed of three phases such as defining ontology, integrating the ontology with semantic networks and pruning the ontology for practically usage. This ontology can be updated and generalized using much easier process and is less time consuming and has specific definition of each word in the form of attributes.

Looking into the future perspective of this project, we can extend this research by building the concept of learning (Supervised as well as Un-Supervised) in the semantic networks so that any new word which does not have any entry in the existing ontology can be added. The architecture can be updated so that if user enters any new word (non-existing), it is been recorded by the model in a separate table (un-supervised learning) and whenever the developer of the architecture is looking for updating in ontology that word has been retrieved by the developer and a new entry must be created in the existing ontology related to that word.

7. REFERENCES

[1] O. Heinonen, K. Hatonen, and K. Klemettinen, "WWW robots and search engines." Seminar on Mobile Code, Report TKO-C79, Helsinki University of Technology, Department of Computer Science, 1996.

[2] Raj Kamal -- *Internet and Web Technologies*; Tata McGraw Hill, 2003

[3] Mike Burner. , "Crawling towards eterneity: Building an archive of the worldwide web ", *Web Techniques Magazin*e, 2(5), May 1998.

[4] Yang Yongsheng, Wang Hui, "Implementation of Focused Crawler", *COMP630D Course Project Report*

[5] Martin Ester, Matthias Grob, Hans-Peter Kriegel, "Focused Web Crawling: A Generic Framework for specifying the user interest and for Adaptive crawling strategies*", Proc. of 27th International Conference on Very Large databases(VLDB '01*), 2001

[6] S. Chakrabarti, M. Van Den Berg, B. Dom, "Focused Crawling: A New Approach to Topic specific web resource discovery", *Proc. Of 8th International WWW conference*, Toronto, Canada, May,1999

[7] Diligenti M., Coetzee F.M., Lawrence S., Giles C.L., Gori M., "Focused Crawling using context graphs", *Proc. International Conference on Very Large Databases (VLDB '00*), 2000,pp. 527-534

[8] F. Crimmins, "Focused Crawling review", 2001 Design of an Agent Based Context Driven Focused Crawler.

[9] D. Bergmark, C. Lagoze, and A. Sbityakov. Focused crawls, tunneling, and digital libraries. In *ACM European Conference on Digital Libraries*, Rome, September 2002.

[10] Steve Lawrence, "Context in Web Search", *IEEE Data Engineering Bulletin*, Volume 23, Number 3, pp. 25-32,2000

[11] S. Chakrabarti, M. Van Den Berg, B. Dom, "Focused Crawling: A New Approach to Topic specific web resource discovery", *Proc. Of 8th International WWW conference*, Toronto, Canada, May,1999

[12] S. Brin, L. Page, " The Anatomy of a Large-Scale Hypertextual Web Search Engine", *Proc. of the 7th International World wide web Conference*, Brisbane, Australia, 1998.

[13] Naresh Chauhan --Design of an Agent Based Context Driven Focused Crawler; BIJIT - BVICAM's International Journal of Information Technology , 2009

[14] Hui LI1,Qiangqiang PENG2, Focused Web Crawling Strategy Based on Web Semantic Analysis and Web Link Analysis,Journal of Computational Information Systems5:6(2009)

[15] Parul Gupta, Context based Indexing in Search Engines using Ontology, International Journal of Computer Applications,2010

[16] M. Diligenti, Focused Crawling Using Context Graphs, Proceedings of the 26th VLDB Conference,Cairo, Egypt, 2000.

[17] David B. Leake, Towards ContextBased Search Engine Selection, *IUI'01,* January 1417,2001, Santa Fe, New Mexico, USA.