A Novel Approach to Priority based Focused Crawler

Rishabh Dixit
Galgotia's College of Engineering and Technology
1, Knowledge Park, Phase II, Greater Noida, Uttar Pradesh 201306

Shiva Gupta
Galgotia's College of Engineering and Technology
1, Knowledge Park, Phase II, Greater Noida, Uttar Pradesh 201306

Shivesh Gupta
Galgotia's College of Engineering and Technology
1, Knowledge Park, Phase II, Greater Noida, Uttar Pradesh 201306

Rajkumar Singh Rathore (Assistant Professor)
Galgotia's College of Engineering and Technology
1, Knowledge Park, Phase II, Greater Noida, Uttar Pradesh 201306

ABSTRACT
The web continues to grow at an exponential rate so fetching relevant information about a specific topic is gaining importance. Web crawlers are programs that traverse the web and fetch the web documents in an automated manner. Focused crawlers search for a specific keyword in a web page. Link based focused crawlers focus on the anchor links of the page and seeks out the most relevant links without actually downloading the web page itself. This paper is based on assigning priorities to different links so that the most relevant links are displayed to the user first. The insignificant links are avoided which leads to significant savings in the computational costs involved in query processing, network, as well as the hardware resources.

General Terms
Web crawling; information retrieval; focused crawlers; search engines; link ranking.

Keywords
Visited URL Test, Content Matching Test.

1. INTRODUCTION
The new advancements in the computation and the networking technologies have made the World Wide Web the biggest database present at the current age. Every second some new information is added. Due to such huge size of the web, it may not be possible for the general crawler to fetch relevant information from the web and keep its index fresh. To counter this problem, focused crawler was proposed [1,2]. Compared to the standard web search engines, focused crawlers produce good accuracy as they restrict themselves to a smaller and specific domain. In this paper, another focused crawler is not being introduced, but another approach for focused crawling is being introduced. A general crawler starts with a seed URL and retrieves all the hyperlinks from that page and stores them in a queue. The crawler takes the first link from the queue, retrieves all the hyperlinks and stores the link again in the queue, and this process is further repeated on and on until the queue is empty. This kind of approach cannot be used further as the size of the web is enormous and a single crawler cannot crawl the whole web. More information about crawling can be found from [3]. Further, the websites are getting updated frequently so the crawler needs to revisit the web page in order to classify the web page again. A focused crawler uses the link structure of the web page. A focused crawler searches the web for a specific topic and retrieves the most relevant information thus creating an index of the relevant pages in a prioritized manner which can be used by search engines to process the user queries. In the crawling process, the focused crawler needs to determine the extent to which the page is relevant to a specific topic before actually downloading it. The focused crawler searches for relevant links and goes deeper into the page when more links to the same topic keywords are discovered. If a non-essential link is encountered, it simply ignores it and moves forward. Thus a focused crawler works with the greatest efficiency by utilizing the minimum resources which is the need of the hour. It also makes it easier to keep the index fresh as new information is added very frequently.

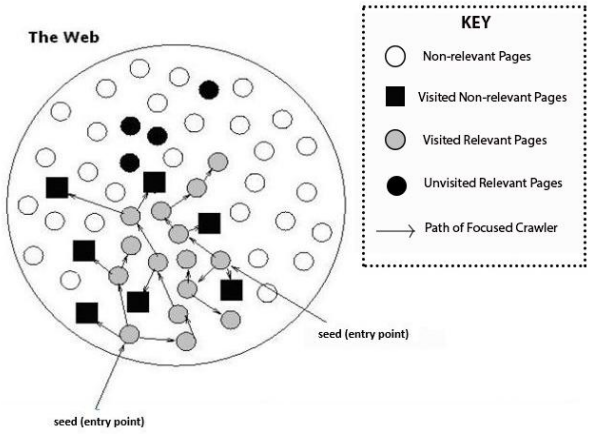

Figure 1: Working of a Focused Crawler [4]

2. RELATED WORK
The earliest web crawling algorithms were based on algorithms like breadth first or depth first traversal. The main motive was to traverse the whole web[5] Chakrabarti first introduced focused Crawling in 1999[1]. Focused Crawler associates a link score to a web page and prioritizes accordingly[6,7].Then P.DeBra et al. proposed fish-search algorithm for collecting topic-specific pages [8].The shark-search algorithm was proposed by M.Hersovici et al. based on the improvement of fish-search algorithm [9].S.Ganesh et al. introduced an association metric[10].This matrix estimates the Semantic content on the domain, based on dependent ontology of the URL is, thus strengthening the metric which is used for prioritizing the URL queue. To evaluate the page value there is analysis of the reference-information among the pages in Link-Structure-Based. This introduced famous algorithms like HITS algorithm [11] and Page Rank algorithm [12].Some later experiments measure the similarity of page contents with a specific subject and reorder the downloaded URLs for the next crawl using special metrics [13]. Some later papers also suggested taking the context of the topic

keywords which increases the relevancy of the results according to the user interests. HAWK [14] is one such focused crawler which considers content and link analysis. Other focused crawling methods include Info Spiders and Best First proposed by X. Zhang [15].

The major problem faced by focused crawlers made till date is that the relevant links extracted by the focused crawler had a static priority that made a particular extracted link somewhat static in the Indexed List of Links. Thus the updating of the extracted list of Links is not efficient enough.

To solve this problem, in this paper another new crawling mechanism has not being introduced but updated former crawlers by assigning priorities to different links so that the most relevant links are displayed to the user first. The insignificant links are avoided which leads to significant savings in the computational costs involved in query processing, network, as well as the hardware resources.

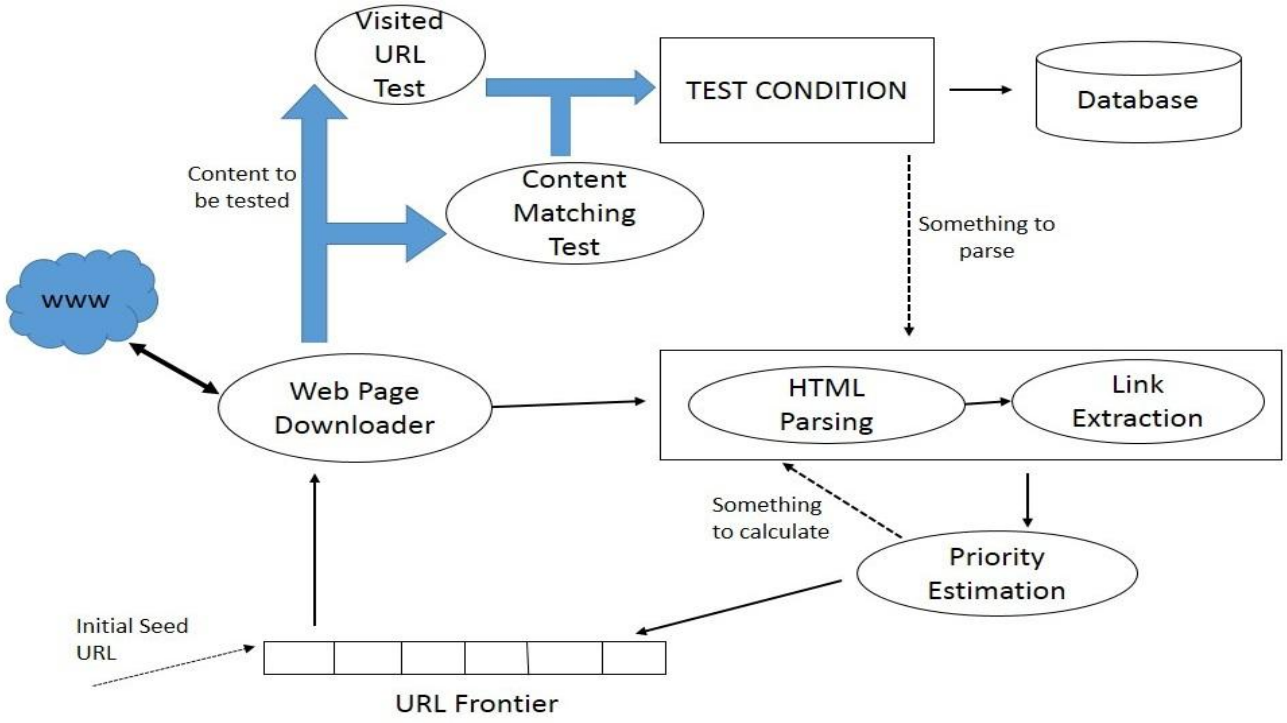

Figure 2: The proposed crawling process

3. PROPOSED ALGORITHM

This method mainly focuses on the assignment of priorities to the URls. Crawling of the URLs are done using First come first serve algorithm. In this algorithm we present a method to prioritize the web pages using its content, indexing, no of redirecting link and frequency of the keywords present in the page. This method uses a technique called visited URL test and Content matching Test not to revisit the URLs.

1. Start by entering the Seed URL.

2. Web page downloader retrieves the web pages found in the given seed URL.

3. Now, it finds the keywords and new URLs present in downloaded pages.

4. Content Matching and Visited URLs test is being taken so that the same URLs can't be visited again.

5. Again, crawler downloads all the web pages corresponding to all new URLs.

6. Now, it calculates priority of all the URLs using its content and indexing.

7. Assign the priority of the URLs as follows:-

i) Initial priority of all the newly visited URLs is 0.

ii) If the Keywords are present in the <Head> tag of the newly visited webpage, then the priority of the URLs is increased by 10 otherwise it remain same.

iii) If keywords are present in the <Href> tag of the newly visited Webpage, then the priority of the URL is increased by 8, else priority remains unchanged.

iv) If the keywords are present in the <Body/Text> tag of the newly visited Webpage, then the priority of the URL is increased by 6, else priority remains unchanged.

v) If the keyword found is related in context to the keyword of the seed URL then the priority of the URL is increased by 4, else priority remains unchanged.

8. The Final priority of the URLs is the cumulative priority of the above.

The seed URL is the initial URL from which the crawling process begins. It must contain good and quality links and keywords so that the classification of keywords and their corresponding links could produce better results. The seed

URL is downloaded by the Web Page Downloader. For the seed URL, the Visited URL test and Content Matching Tests are not taken as it is not required. The downloaded web page is then fed to the HTML Parser and Link Extractor. This module searches the web page for specific topic keywords and other links which are stored in the queue to be crawled later. If the new URL is not being revisited, then the HTML Parser and Link Extractor again extracts keywords and links and this process is followed till the queue is empty. The priority estimation is done according to the contents of the web page.

3.1 Visited URL Test

The final URL obtained will be stored from the web page instead of the short links or the redirecting links. The URLs are generally long and contain the topic of the webpage which will increase the size of the database. The matching process will take a lot of time and space. The complexity of the system will increase as well. To counter this problem we will convert this link into a CRC32 checksum and store it in the database. This will save significant amount of space and will speed up the matching process as well. Then we will use this database to match it for already visited URLs so that it does not revisits the URLs once visited. The problem occurs when the contents of the webpage are updated then the crawler will not re-visit the web page. In this case the crawler will not visit the web page although the contents of the web page have been updated. To counter this problem, we propose a Content Matching Test.

3.2 Content Matching Test

This test matches for existing content in the database. It prevents mirrored pages from being downloaded. The matching of whole document is a very lengthy and complex task. It will also consume a lot of storage and processing time. To counter this problem, we propose to create a checksum of the content of the whole document and store it in the database. This will significantly reduce the storage and processing time while matching the contents of the web page.

3.3 Test Cases

The Visited URL Test and Content Matching Test work together to determine the authenticity of a web document. Let us assume Visited URL Test is VUT and Content Matching Test is CMT then four distinct cases arrive while matching the document. The URL will be sent along with the test cases (Boolean values) to be stored in the database. The checksum of links will be matched for the Visited URL Test and whole page checksum will be matched for Content Matching Test. A table showing the test cases is mentioned below. The output from these cases decides the further processing of the current URL. A detailed explanation of the cases is discussed the next section.

Table 1. Test Cases for Visited Url Test and Content Matching Test

VUT	CMT	Output
0	0	Move to next step
0	1	Irrelevant page
1	0	Move to next step
1	1	Irrelevant page

3.3.1 Test cases terminology

VUT

0: The URL has not been visited

1: The URL has been visited.

CMT

0: The content did not match

1: The contents match.

According to the above four cases, the decision is made whether to visit the page. If a new URL is being visited (VUT=0 and CMT=0) then the crawling process goes on as proposed. If the URL has been visited but the contents of the page did not match (VUT=1 and CMT=0) then it means that the web page has been updated since the last time the page was visited by the crawler. So again the crawling process will go on. If the URL has not been visited but the contents of the page match (VUT=0 and CMT=1) then it does not crawl the page as the page may be duplicate or mirrored. So it categorizes the webpage as irrelevant page and aborts the process. If the page has been visited and the contents also match (VUT=1 and CMT=1) then it means that the page is already present in the crawler's database. So it does not follow the process in this case as well.

4. CONCLUSION

In this research paper, minimum resources have been used as large amount of space and processing power is needed for crawler mechanism. Due to the large and ever increasing size of the web, the crawler needs to be simple and accurate. The proposed crawling technique has minimal complexity and is fast as well. The proposed technique also avoids duplicate/mirrored links or same content over the web which saves significant amount of bandwidth. The storage of the web pages is done using checksum which reduces the storage space and also reduces the complexity during the Visited URL/Content Matching test as compared to the text form of links and web documents.

5. FUTURE WORK

Focused Crawlers are going to be an important tool in the future. Classification of documents is essential to provide the best results in the least amount of time to the user. Although the proposed focused crawler is simple and fast but it does not considers the words in context to the specified keywords. So the words which have similar meaning or are used in a similar context are considered to be different keywords and separate records are maintained in the database. Keywords which are homonyms also create redundant entries in the database. Code optimization is also desired to be done from time to time to improve the performance of the crawler.

6. REFERENCES

[1] S. Chakrabarti, M. van den Berg, B. Dom, 1999 on "Focused crawling: a new approach to topic-specific Web resource discovery".In "8th International WWWConference on WWW",Toronto,Canada ,pp.1623.

[2] J. Cho, H. Garcia-Molina, and L. Page, "Efficient crawling through URL ordering," Seventh WWW Conference, 1998.

[3] C. Olston, M. Najorl, 2010 on "Web Crawling", Foundations and Trends in Information Retrieval, Vol. 4, (3), . pp – 175-246.

[4] B. Ganguly and D. Raich, 2014 on "Performance Optimization of Focused Web Crawling Using Content Block Segmentation" at International Conference on Electronic-Systems, Signal-Processing and ComputingTech.

[5] O. Heinonen, K. Hatonen, and K. Klemettinen, 1996 on "WWW robots andsearch engines." Seminar on Mobile Code, Report TKO-C79, HUT, Department of CS.

[6] K. Bharat and M. Henzinger, 1998 on "Improved algorithms for topic distillation in hyperlinked environments,"at Twenty first Int'l ACM SIGIR Conference.

[7] J. Kleinberg, 1997 on "Authoritative sources in a hyperlinked environment." Report RJ 10076, IBM.

[8] De Bra,P. and Post, R. , 1994 on "Information Retrieval in the World-Wide Web: Making Client-based searching feasible".

[9] M. Hersovici, A. Heydon, M. Mitzenmacher, D.pelleg, 1998 on "The Sharksearch Algorithm-An application: Tailored Website Mapping." At World Wide Conference, held in Australia, 317-326.

[10] S. Ganesh, M. Jayaraj, V. Kalyan, S. Murthy and G. Aghila.,2004 on "Ontologybased Web Crawler", IEEE Computer Society, Las Vegas – Nevada – USA, pp. 337-341.

[11] Jon M. Kleinberg, 1999 on "Authoritative Sources in a Hyperlinked Environment", Journal of the 9th ACM-SIAM Symposium on Discrete Algorithm, 46(5), 604-632.

[12] S. Bri, L. Page, 1998 on "The anatomy of large-scale hypertext Web search-engine",suggested at 7th World-Wide Web Conference, Australia, 107-117.

[13] J. Cho, H. Garcia-Molina, and L. Page, 1998 on "Efficient crawling through URL-ordering,"at Seventh World-Wide Web Conference.

[14] X.Chen and X. Zhang, 2008 on "HAWK: A Focused Crawler with Content and LinkAnalysis", presented at ICEBE, China.

[15] Zhang X.,Zhou T. ,Yu Z. and Chen D., 2008 on "URL Rule Based Focused-Crawlers",conference IEEE-ICEBE,China .