Content Relevance Prediction Algorithm in Web Crawlers to Enhance Web Search

Saurabh Pakhidde*1 , Jaya Rajurkar#2 , Prashant Dahiwale3**

***Abstract*- A Web crawler is a computer program that browses the World Wide Web in a methodical, automated manner or in an orderly fashion.. Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches. An efficient web crawler algorithm is required so as to extract required information in less time and with highest accuracy. As the number of Internet users and the number of accessible Web pages grows, it is becoming increasingly difficult for users to find documents that are relevant to their particular needs. Users must either browse through a large hierarchy of concepts to find the information for which they are looking or submit a query to a publicly available search engine and wade through hundreds of results, most of them irrelevant. Web crawlers are one of the most crucial components in search engines and their optimization would have a great effect on improving the searching efficiency. Generally web crawler rejects the page whose url does not contain the search keyword while searching information on World Wide Web. But it may so happen that these pages may contain information required. Our main emphasis will be to scan these pages and parse them check for their relevancy by assigning each page a page weight and arrange them in terms of most relevant page first and then second page and so on according to the page weight that we may gain more relevant information or site addresses at top of result..Thus we will get more accuracy while searching some information on network. Our main work will be based on architecture of web crawler and to modify it according to our need so as it will also scan pages along with page weight. We will then arrange result gathered to make result more efficient.**

***Keywords*- Web Crawler, Seed, Frontier, Content Prediction Algorithm, Web Search, Search Engine.**

I. INTRODUCTION

A Web Search Engine is a software that is used to search information on the World Wide Web [1-4]. The information may be a specialist in web pages, images, information and other types of files. Search Engines maintain real time information by running an algorithm on "*Web Crawlers"* [1,3,4,7]. A web crawler is a program that, given one or more seed(starting link) URLs, downloads the web pages associated with these URLs, extracts any hyperlinks contained in them, and recursively continues to download the web pages identified by these hyperlinks. Web crawlers are an important component of web search engines, where they are used to collect the corpus of web pages *indexed* by the search engine. The large size and the dynamic nature of the Web highlight the need for continuous support and updating of Web based information retrieval systems. Crawlers facilitate the process by following the hyperlinks in Web pages to automatically download a partial snapshot of the Web.

Some basic terms related to crawlers are:

- ***Seed***- It is the starting URL from where the where crawler starts traversing the World Wide Web recursively[1].
- ***Frontier***- It is the list of unvisited URLs. The choice of visiting or ignoring these URLs depends on the algorithm being followed[1].
- ***Relevant Page***- The pages that contain information pertaining to users need. The relevancy can be deciding either by checking the URL or the content or in some cases both[1].
- ***Irrelevant Page***- The pages that are useless to users. Irrelevant Pages are decided when the URLs and content do not match the users query[1].

Given the current size of the Web, even large search engines cover only a portion of the publicly-available Internet; a study by Lawrence and Giles (Lawrence and Giles, 2000) showed that no search engine indexes more than 16% of the Web. The crawler cannot download all the pages from the web. It has to decide which pages to download. Thus, the main aim of the crawler is to download only the "important" pages. The definition of important for us are the pages that contain some information regarding the users search query, others are irrelevant.

II. AIM AND OBJECTIVE

A 2005 study showed that large-scale search engines index no more than 40-70% of the index able Web. As a crawler always downloads just a fraction of the web pages, it is highly desirable that the downloaded fraction contains the most relevant and not just a random sample of the web. Today's Web crawler has the relevancy mechanism which checks the URL to decide whether the users search query matches the URL name. If not, then the page is deemed irrelevant [1]. But it is not necessary that if the URL is irrelevant then the page would not contain any information regarding the users search query. Thus the contents of the page have to be checked to know whether or not they contain the information that is of importance to the user by matching the users search string to the page contents. If the string occurs in the page, then that page is termed as relevant else it is irrelevant. Thus the "*Link based Relevancy check is moved to a Content based check"*

The contents of the page have to be matched alongside the users query to check whether that page is *Relevant* or *Irrelevant*. As a result we can retrieve links to get more accuracy in the results. We are thus extending the current functionality of the web crawlers to Content-based checking.

III. MOTIVATION

Most of the earlier work on Web Crawlers was dedicated to increasing efficiency by finding out more relevant pages in less time i.e. to give more page-hit in less time. We needed a mechanism to check as to '*What degree is a page relevant to the user'*. Thus it is needed that we should generate more result and more relevant information in given time. If the Link Based checking is performed, it is seen that sometimes results are found but there may be times that the results are not found on the pages on which the links match to the users query and thus, such pages have to be ignored. Along with link checking content of the page has to be checked so as to determine the degree of content relevancy in a particular page. For this, the tags of the page are to be assigned weights (the mechanism that we call *parsing*). The results of the link based search and content based search has to be combined so as to find the total weight of the page and this will thus decide the relevancy degree of that page. The pages which will not have any weight will thus be useless for users and thus are deemed Irrelevant.

IV. LITERATURE SURVEY

Web crawlers are almost as old as the web itself and in the spring of 1993, just months after the release of NCSA Mosaic, the first web crawler was written, the *World Wide Web Wanderer*, which was used from 1993 to 1996 to compile statistics about the growth of the web.

In later years it was proposed that the "*first research paper containing a short description of a web crawler, the RBSE spider*" has to be researched and implemented.

A CSE student at the University of Washington, in 1994 started *WebCrawler in his spare time. At first, WebCrawler was a desktop application, not a Web service as it is today.* WebCrawler spat out its first Top 25 list on March 15, 1994. WebCrawler goes live on the Web with a database containing pages from just over 4000 different Web sites.

After a few years *the first detailed description of the architecture of a web crawler* was proposed, namely the original Internet Archive crawler.

Brin and Page, proposed a *seminal paper on the early architecture of the Google search engine contained a brief description of the Google crawler*, which used a distributed system of page-fetching processes and a central database for coordinating the crawl.

Heydon and Najork, proposed and described ***Mercator***, *a distributed and extensible web crawler that was to become the blueprint for a number of other crawlers*. Other distributed crawling systems described in the literature include PolyBot, UbiCrawler, C-proc and Dominos.

V. ARCHITECTURE

This architecture only worked with:

- Relevant pages based on URL i.e. Link based Crawling.
- Irrelevant page information checking was discarded.
- Important components of the original crawler architecture are as follows:

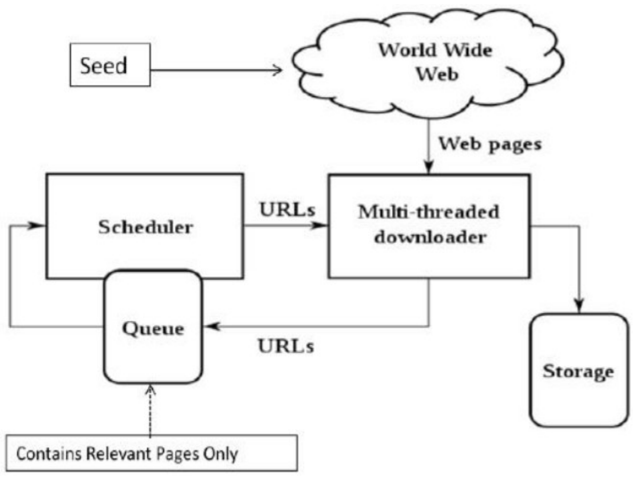


Fig 1:Original archictecture of web crawler

a) Seed URL
b) Search String
c) Downloader
d) Queue
e) Storage (To store text and metadata)
f) World Wide Web (database to crawl)

VI. MATERIAL AND METHODOLOGY

In this architecture, the crawler connects to the web using the seed URL and the search string to find the relevant pages based on the URL and downloads those pages. The irrelevant pages are ignored or discarded. This is "Link based Crawler".

We will now move to a proposed architecture of the web crawler that parses the content along with checking for URL relevancy.

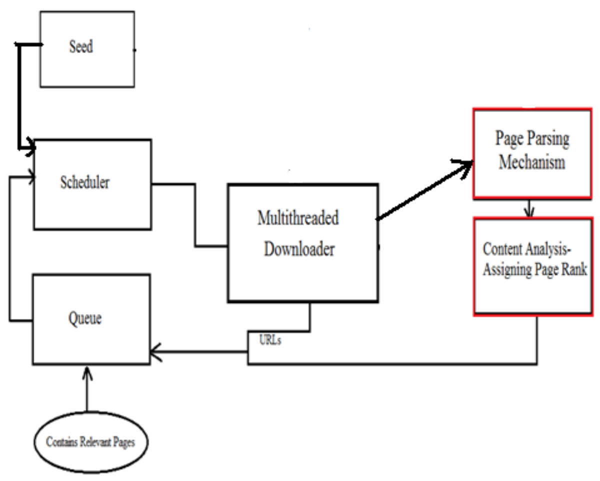


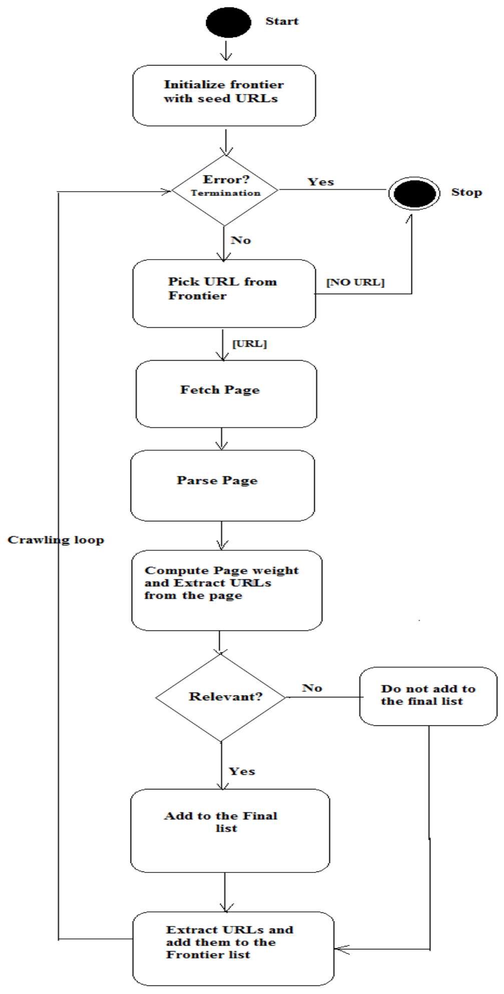
Fig. 2 Proposed Architecture of Web Crawler

The proposed crawler follows these steps:

1. The seed URL is used to connect to the World Wide Web. Along with the seed, a search string is to be provided by the user. The search string will act as the query that the user is searching for. If the URL is valid and the string is valid, then the crawler proceeds else it stops.
2. Every Web page has got tags that may contain information. Weights in the form of integers have been assigned to the tags of the page. If the content is found on the page (once or multiple times), the weight is to be multiplied by the number of occurrences in each tag and the total weight is added up to get the total page weight.
3. The Seed is downloaded and all the URLs on the seed page is extracted. The seed page is then checked to see whether it contains any relevant information. If yes, then the weight is calculated and it is set aside for further use, else it is discarded.
4. The URLs extracted are then scheduled to be downloaded one by one and the above process is followed (i.e. download a page and check its contents). If it returns a positive page weight then set it aside and recursively do the same for every page.
5. Sort the pages in descending order of page weight and display the results.

VII. IMPLEMENTATION

The Flowchart of the implementation of the project is as follows:


Start
Initialize frontier with seed URLs
Error? Termination
Yes
Stop
No
Pick URL from Frontier
[NO URL]
[URL]
Fetch Page
Parse Page
Crawling loop
Compute Page weight and Extract URLs from the page
Relevant?
No
Do not add to the final list
Yes
Add to the Final list
Extract URLs and add them to the Frontier list

[

- Enter the Seed URL and the search string.
- Checking the validity of the Seed.
- Connecting to the World Wide Web and downloading the Seed page.
- Parsing the source code of the seed page so as to check with the search String and match its existence with it.

•Calculate the page weight. If weight > 1, then page is relevant else it's irrelevant. Extract the URLs and add them to the frontier list.

- Follow the crawling loop on the Frontier.

VIII. CONCLUSION

Using this concept we have implemented relevance prediction mechanism which is link based and it has been extended to being *content based* as well.this content prediction mechanism increases the overall results as it scan and outputs the pages which will be most useful to the users first based on page weight. We

believe this would increase the efficiency since the function of crawler is to provide efficient results to the search query. This will be an important tool to the search engines and thus will facilitate the newer versions of the search engines

IX.ACKNOWLEDGEMENT

I would like to thank my guide Prof. Prashant Dahiwale for providing me with valuable resources and insights needed for my project. He has been a guideline for me throughout the project. I really appreciate his valuable time spent for the betterment of this project. I would like to thank all my professors from the Information Technology Department who have helped enrich my knowledge and for all their support and encouragement.

REFERENCES

[1] Prashant Dahiwale, Anil Mokhade, M.M. Raghuwanshi, *Intelligent Web Crawlers,* ICWET, ACM New York, NY, USA, pp. 613-617, 2010.

[2] Brian Pinkerton, *Finding what people want: Experiences with the Web Crawler,* Proceedings of first World Wide Web conference, Geneva, Switzerland, 1994.

[3] Gautam Pant, Padmini Srinivasan, Filippo Menczer, *Crawling the Web*, pp. 153-178, Mark Levene, Alexandra Poulovassilis (Ed.), *Web Dynamics: Adapting to Change in Content, Size, Topology and Use,* Springer-Verlag, Berlin, Germany, November
2004.

[4] Christopher Olston, Marc Najork, *Web Crawler Architecture,* Journal Foundations and Trends in Information Retrieval archive, Volume 4 Issue 3, pp.
175-246, March 2010.

[5] Sriram Raghavan, Hector Garcia-Molina, *Crawling the Hidden Web, Computer* Science Department, Stanford University, Stanford, CA 94305, USA.

[6] Junghoo Cho, Hector Garcia-Molina*, The Evolution of the Web and Implications for an Incremental Crawler,* Department of Computer Science, Stanford, CA 94305.

[7] *Web crawling and indexes,* WEB CRAWLER SPIDER, Chapter 20, pp. 443-459, 2009.

[8] The Dev Article website. [Online]. Available: *http://www.devartical.com/*

[9] The Stack Overflow website. [Online]. Available: *http://www.stackoverflow.com/*

[10] Java Tutorial website. [Online]. Available: *http://www.mkyong.com/*

[11] Java Tutorials (Javadocs). [Online]. Available: http://docs.oracle.com/javase/tutorial/

[12][WordNet] Web page of WordNet project at Cognitive Science Laboratory of Princeton University http://wordnet.princeton.edu/obtain

AUTHORS PROFILE

Prof. Prashant Dahiwale is a lecturer of Computer Science & Engineering Department at Rajiv Gandhi College of Engineering & Research, Nagpur, MS, India and is currently pursuing his PhD from Visvesvaraya National Institute of Technology, Nagpur. He is a published author in journals such as ICWET, ACM New York.

Ms. Jaya Rajurkar is currently in her Final Year and pursuing her Bachelor in Engineering (B.E) Degree from Rajiv Gandhi College of Engineering and Research, Nagpur.

Mr. Saurabh Pakhidde is currently in his Final Year and pursuing his Bachelor in Engineering (B.E) Degree from Rajiv Gandhi College of Engineering and Research, Nagpur.