ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (274-277), Month: January-April 2016, Available at: www.noveltyjournals.com

Survey Paper on Smart Web Crawler

[1]Rutuja Thite, [2]Bhagyashree Pawar, [3]Tejaswini mode, [4]Mitali Mete

[1,2,3,4] Computer Engineering, SKNSITS, Lonavala

Abstract: **In today scenario World Wide Web is flooded with huge amount of information. Finding useful information from the Web is quite challenging task. There are many search engines available in the market that will solve our purpose. However among all, selecting proper search engine with highly effective web crawler is quite necessary. There are many challenges in the design of high performances web crawler like it must be able to download pages at high rate, store it into the database efficiently and also crawl page rapidly. In this paper we present taxonomy of WebCrawler, various challenges and solutions of WebCrawler and various crawling algorithms.**

Keywords: **Deep web, hidden web, domain-specific search, query form discovery.**

1. INTRODUCTION

This topic introduces prior studies covering initiatives for the web search, search engines, search engine optimization techniques, limitations of existing search engines, meta-search engines, meta-search engine optimization techniques, difference between search engines and meta-search engines, limitations of existing meta-search engines, need of a new model of meta-search engine and scope of research in meta-search engine for specific information retrieval in an efficient manner.

As the web surfing is growing at a very fast pace, there has been increased focous in techniques that help improve efficiency to locate deep searches. However, due to the large amount of web resources and the dynamic nature of deep web, achieving wide coverage and high efficiency is a challenging issue. We propose a two-stage framework, namely *Smart Crawler*, for efficient harvesting deep web interfaces. In the first stage, *Smart Crawler* performs site-based searching for center pages with the help of search engines, avoiding visiting a large number of pages. To achieve more accurate results for a focused crawl, *Smart Crawler* ranks websites to prioritize highly relevant ones for a given topic. In the second stage, *Smart Crawler* achieves fast in-site searching by excavating most relevant links with an adaptive link-ranking. To eliminate bias on visiting some highly relevant links in hidden web directories, we design a link tree data structure to achieve wider coverage for a website. Our experimental results on a set of representative domains show the agility and accuracy of our proposed crawler framework, which efficiently retrieves deep-web interfaces from large-scale sites and achieves higher harvest rates than other crawlers.

2. HISTORY OF THE WEB SURFING FOR WEB SEARCH

The roots of web search engine technology are in Information Retrieval (IR) systems, which can be traced back to the work of Kuhn at IBM during the late 1950s. IR has been an active field within information science, and has been given a big boost since the 1990s with the new requirements that the Web has brought.

Many methods used by current search engines can be traced back to the developments in IR during the 1970s and 1980s. Especially influential is the SMART (System for the Mechanical Analysis and Retrieval of Text) retrieval system, initially developed by Gerard Salton and his collaborators at Cornell University during the early 1970s.

Prior to 1990, there was no approach to search the Web. At that time there were a small number of websites. Most sites contained collections of files that user could download. The only way user could find out that a file was on a specific site. Then came a tool which is called Archie. It was the first program to search the Web for the contents of all websites all over the world. It is not actually search engine but like Yahoo, it is to search list of files. Information seeker needed to know the exact name of the file for which he/she is looking for. Prepared with that information, Archie would advise from which website it is possible to download the file.

3. CRAWLER POLICIES

A Web crawler has various tasks and goals that must be handled carefully in spite of various contradictions amongst them. Also the various resources that are available must be used by web crawlers efficiently, also including network bandwidth which must exhibit a high degree of parallelism without affecting the web server by overloading. So policies are followed to maintain a crawling decorum.

a) A selection policy that states which pages to download.

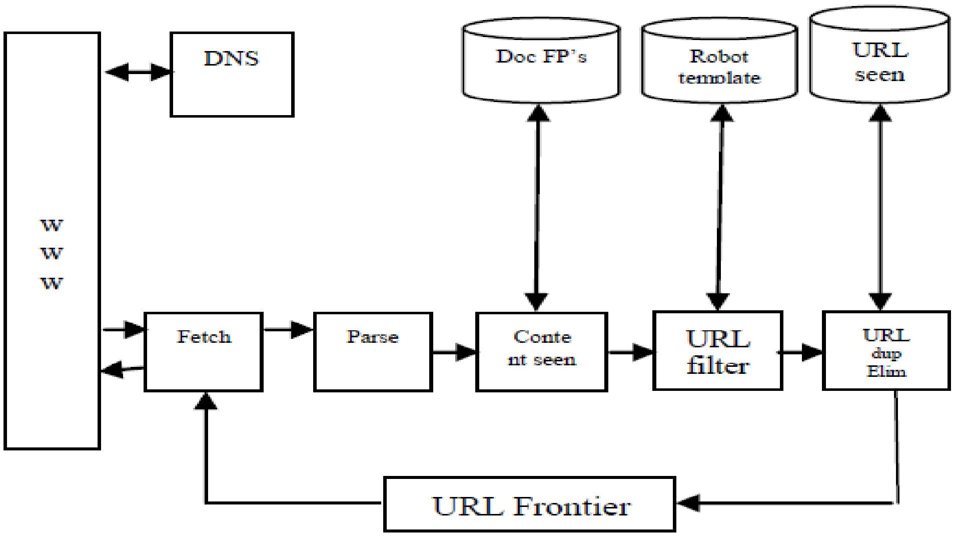
b) A re-visit policy that states when to check for changes to the pages.

c) A politeness policy that states how to avoid overloading Web sites.

d) A parallelization policy that states how to coordinate distributed Web crawlers.

4. WEB CRAWLER ARCHITECTURE

The web crawler architecture below shows how it crawls through a whole site on the Internet



Basic web crawler Architecture:

The above architecture outlines some of the modules that depict a very useful working scenario of the crawler. The URL frontiers, contains URLs yet to be fetched soon. A DNS resolution module converts the URLs to its equivalent IP address. A fetch module to retrieve the web pages associated with its URL. A Parsing module is used for extraction of text and set of links from a retrieved web page. Finally, a duplicate elimination module checks to see whether there are any duplicate URL in the frontier. A crawler fetches web pages associated with an URL by taking an URL from the frontier using http protocol. Once fetched, the web pages are stored temporarily, thereafter the page is parsed and the text as well as the links in it is being extracted.

After the URLs being parsed, the very next task is to check whether the web page with same content has being seen at another URL.

The *URL filter* has another promising task which filters out the URLs that it should be excluded from the frontier based on several tests. For instance it may exclude (say all .com URLs). The *Robot Exclusion protocol* is used by most of the host by placing certain portions of the websites off-limits to crawling.

This protocol works by placing a robot.txt file at the root of the URL. The final process is to check for any duplicate URLs, if the URL is already present in the frontier i.e. already crawled then we do not add it to the frontier.

5. DESIGN ISSUES

Various Data structures are required while considering design issues such as one data structure for maintaining the set of URLs that have been discovered (whether downloaded or not), and a second data structure for maintaining the set of URLs that are yet to be downloaded. The first the second data structure (usually called the frontier) must support adding URLs, and selecting a URL to fetch next, while the second data structure (sometimes called the URL-seen test‖ or the duplicated URL eliminator‖) must support set addition and set membership testing.

The basic web crawler algorithms are modified in many variations that give search engines different levels of coverage. Billions of websites are crawled and indexed using algorithms depending on a number of factors.

a) Breadth-First search Algorithm: This Algorithm starts its crawling process right from the root node and sweeps down searching for the related neighbouring node at the same level. While crawling, if at the very first level itself finds the relevant node or its objective then a success occurs and terminates else goes on finding its objective down the very next level.

Breadth first is well suited for situations where the objective is found on the shallower parts in a deeper tree.

b) Depth-First search: A technique which systematically traverses through the search by starting at the root node and traverses beneath down the child nodes is a powerful Depth-First search. While visiting each child nodes the objective is searched and so on the process continues if not found. If there is more than one child, then which node to visit depends upon the priority (i.e. left most child) and traverses deep until no more child is available. A backtracking method is used for traversing to the next unvisited node and then continues in a similar manner.

c) PageRank Algorithm: In a PageRank, each of the pages on the web has its own measure which is independent of any informational needs. This algorithm ranks the web pages according to their importance or relevance. The page rank of a given page is calculated as:

$PR(A)= (1-d) + d(PR(T1)/C(T1) +...+ PR(Tn)/C(Tn))$

PR(A) Page Rank of a website,

d damping factor,

T1…..Tn Link

6. CRAWLING TECHNIQUES

The crawling method used by various search engines in order to download pages that have already been downloaded and those that are yet to be downloaded relies greatly on various techniques such as follows:

A. FOCUSED CRAWLER:

Focused crawling was coined by as an efficient resource discovery system. It has three components –crawler, classifier and distiller. The Focused crawling is a technique which is specifically designed to gather web pages on a specific topic by carefully prioritizing the crawl frontier hardware requirements, increases the reliability and download speed. Thus, the distribution of process to multiple processes makes the system scalable. domain or crawl pages with larger Page Rank), whereas a general crawler gathers as many pages as it can from a particular set of URLs.

B. DISTRIBUTED CRAWLER:

In distributed web crawler a URL server distributes individual URLs to multiple crawlers thereby downloading the web pages in parallel and then sends the downloaded pages to a central indexer on which links are extracted and sent via the URL server to the crawlers.

The crawling over network of workstations in parallel requires a reasonable amount of time to complete the crawl process which ultimately leads to reduction of hardware requirements, increases the reliability and download speed. Thus, the distribution of process to multiple processes makes the system scalable. It basically uses Page rank algorithm for its increased efficiency and quality search.

C. INCREMENTAL CRAWLER:

An Incremental crawler continuously crawl the entire web and updates the existing downloaded pages instead of restarting. A data from previous cycles is taken to decide which pages should be checked for updates resulting in high freshness and low peak load is achieved.

D. HIDDEN WEB CRAWLER:

Hidden web is the term used to describe the information available on the web that is hidden behind the search query interfaces that act as entrance to backend databases.

Many web pages cannot be crawled without filling up the forms as such they are inherently hidden behind search forms and are termed as ―Deep web‖. A study describes that only publicly indexable web (PIW) i.e. set of pages which are accessible by following the hyperlinks. A further study by Raghavan and Garcia-Molina observed that crawling of ―Deep web or Hidden web‖ has revealed

7. CONCLUSION

In this paper we have studied how to build an effective web crawler. The study carried out based on crawl ordering reveals that the incremental crawler performs better and is more powerful because it allows re-visitation of pages at different rates. Crawling at other environment, such as peer-to-peer has been a future issue to be dealt.

REFERENCES

[1] Alex Wallar, Erion Plaku, and Donald A. Sofge, "Reactive Motion Planning for Unmanned Aerial Surveillance of Risk-Sensitive Areas" IEEE Transaction on automation science and engineering, vol. 12, no. 3, july 2015.

[2] Samer Hawayek, Claude Hargrove and Nabila A. BouSaba,"Real-Time Bluetooth Communication Between anFPGA Based Embedded System and an AndroidPhone" 978-1-4799-0053-4/13/$31.00 ©2013 IEEE.

[3] Daniel Herrera, Javier Gimenez, Ricardo Carelli,"Human-Robot Interaction in Precision Agriculture: Sharing the Workspace with Service Units" 978-1-4799-7800-7/15/$31.00 ©2015 IEEE.

[4] M Muhammad, D Swarnaker, M Arifuzzaman,"Autonomous Quadcopter for Product Home Delivery", International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT) 2014.

[5] AhmetTabanlıoglu, Adil Çagatay Yucedag Bilgisayar Muhendisligi Bolumu,"Multicopter Usage for Analysis Productivity in Agriculture on GAP Region", 978-1-4799-4874-1/14/$31.00 ©2014 IEEE.