ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 9, September 2017

Two Stage Intelligent Focus Crawler Using JavaScript Parser

Revati Rajane, Prof. Pradnya Kasture

M. E Student, Department of Computer Engineering, RMDSSOE, Pune, Maharashtra, India

Professor, Department of Computer Engineering, RMDSSOE, Pune , Maharashtra, India

ABSTRACT: The World Wide Web is a massive assemblage of billions of web pages containing terabytes of data arranged in various servers using HTML. The all-purpose crawlers are challenged extensively at a fast pace from a scaling point of view because of the fast-paced evolution of the internet. A web crawler is a mechanized (automated) tool that traverses the web and extracts webpages for gathering information. In intelligent focus Web crawler, the crawler starts with a specific defined topic and crawls the relevant webpages based on the defined search criteria. In this project, a new intelligent focus crawler has been proposed. A. The goal of the focused crawler is to identify and notify pages based on the most relevance limiting the search scope to the boundaries of pages that are with the pre-decided relevance factors. This helps in reducing network and hardware resources, in turn leading to cost savings and improves the efficiency and accuracy of the crawl data stored. For this purpose, it uses" Reverse Searching Strategy". Keeping this aim in mind a two-level framework is used, for efficient searching and gathering of deep and hidden web interfaces. In the first stage, it uses search engines to identify main pages which avoid visiting irrelevant pages. After identifying the pages, the intelligent focus web crawler will prioritize the webpages to rank them to be more relevant than the other based on the search topic. In the second stage, the crawler searches the insides of the websites for relevant information based on the defined search criteria.HTML and JavaScript parser is developed to deal with the dynamic pages. Moreover, a report on crawled URLs is published after crawling which gives entries of all crawled URLs and errors found.

KEYWORDS: Intelligent Crawler, focused crawler, weight table, World-Wide Web,Search Engine, links ranking, HTML Parser, JavaScript Parser.

I. INTRODUCTION

This category of research falls under the domain data mining and information retrieval. Here are some important concepts regarding web crawling.

Web Crawler [1]–An automated package that systematically scans and downloads internet webpages that can be reached via hyperlinks. It starts with a list of web URL's which are also known as seeds and then it visits these URL's to further hyperlink the relevant URL' creating a list called crawler frontier. These URL's are revisited multiple times as per the defined procedure that the user defines.

Focused Crawler [2]–A package that is geared towards and most used by subject or domain specific required web portal users. This locally maintains a log and helps in efficient accessing of intricate and important information.

II. RELATED WORK

1. Smart Crawler [1]: For efficient and in-depth analysis of web pages, a Two-Stage Crawler concept is used. In this the crawler uses two strategies stepwise- a. site locating and b. in-site exploring. The focused web crawler downloads pages based on the relevancy of each other and the defined criteria. The performance of the crawler is

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 9, September 2017

defined by how relevant the gathered information is. This technique eliminates the problem of irrelevant results and hence most accurate results can be found.

2. Study of Web Crawler and Types [2]: The given paper studies distinct types of crawler's such as: 1. Focused WebCrawler, 2. Distributed Crawler, 3. Incremental Crawler, and 4. Parallel Crawler. Wastage of resources is minimized at the maximum by making sure the search criterion is specific. Another technique that is used called as 'Surfacing' is used to maximize the coverage while using very limited queries.
3. Novel Architecture of Web Crawler for URL Distribution[3]: This paper also goes into explaining how the Crawler balances the URL load for performance optimization. This technique is called load balancing.
4. Web Crawling Algorithms [4]: The paper also provides a detailed study of algorithms which work efficiently for static as well as dynamic searches and can be used as searching algorithms during crawling. Some of them are as follows: 1. Breadth First Search, 2. Best First Search and 3. Fish Search. Other searching algorithms such as, 1. Page Rank algorithm and 2. Batch Page Rank algorithm are also proposed which reduce cost and network traffic.
5. Domain-Specific Web Site Identification [5]: The CROSSMARC Focused Web Crawler In this paper a notion of crawler is proposed which utilizes different techniques like Site navigation, Page-filtering, Link-scoring. These methods increase the relevancy of subsequent URLs. This type of crawling is also called as Goal directed Crawling since the crawler considers the end user's preference and constraints.

III. EXISTING METHODOLOGY

Following techniques are used for current web crawling systems:
The number of Web pages is increasing in a very fast proportion. This growth has urged the development of retrieval tools like search engines to get the information from WWW. For retrieving information of the web, web crawling plays an important role. A package that systematically and automatically goes over the World Wide Web (www) to maintain a database of the visited pages for later handling by a given search engines is called Web Crawling. Topic specific crawlers are used for specialized web portals and specialized applications. These crawlers are designed to retrieve pages that are relevant to the triggering topic.

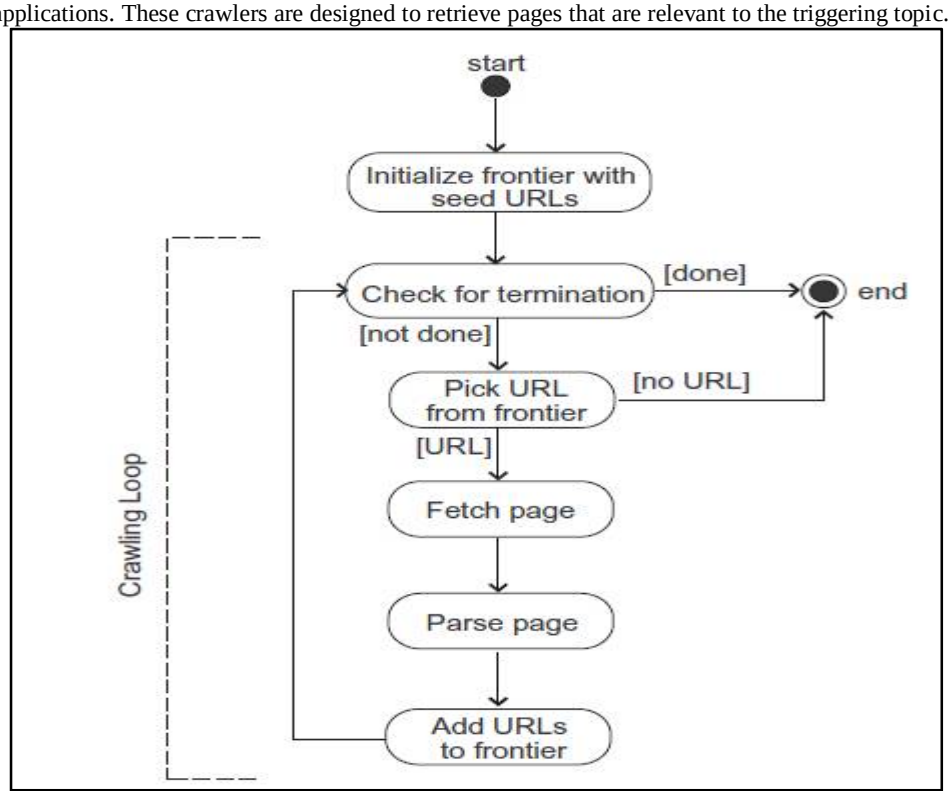

Figure 1. Flow of a basic sequential crawler

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 9, September 2017

Figure 1, shows the flow of a general-purpose WebCrawler. The crawler maintains a list of unvisited URLs called the frontier. The list is initialized with seed URLs, which may be provided by a user or another program. This crawler internally preserves a list of unvisited URLs in either database or Text file. This is called as frontier [6].

Given list contains the seed URLs. Seed URLs are used by the crawler as a starting point that may be given by the user. Each crawling step contains selecting the URL from the list i.e. frontier and then fetch the corresponding page through hyperlink, parse6 HTML contents from the fetched page to extract all the hyperlinks and application specific data. Treat these hyperlinks as frontier and recursively follow the same process. Each frontier is assigned to a score which gives the projected advantage of visiting the equivalent URL. After several pages have been crawled the process can be terminated based on the defined counter. In case no more frontier is available and the crawler is ready to crawl then the state indicates a dead end. Once it reaches this dead end it stops. When awebsite is crawled, it can be viewed as a graph where nodes are represented by pages and edges between them are represented by the hyperlinks. This problem can be considered as a graph search problem7. Root node of a graph that is starting point of a crawler is obtained by referring seed URLs and then follows the edges to reach next node. As the new web page is fetched further hyperlinks are extracted which is equivalent to expanding a node in a graph. When the frontier is empty, leaf node is obtained and the crawler stops.

Drawbacks:

Every time the crawler takes the starting URL from the seed URLs and Frontier but when that is empty, it is a dead end for the crawler. If the number of seed URLs is increased, then it will affect the speed of the crawling process. Also, a web crawler may face a spider trap5 problem which means that many different URLs redirect to the same page.

While this is a drawback, this can be improved by defining the suggested number of pages that the crawler should crawl to extract information specific to the defined criteria before it terminated. The web crawler can make sure that the frontier should contain the URL which is a fully qualified domain name (e.g. www.testweb.com). As a result, the crawler will not access the same URL again and hence the problem of spider trap is solved.

.

IV. PROBLEM ANALYSIS

When the user provides a set of domain keys, a crawling technique is to be developed such that it makes a list of relevant pages based on the specific domain keys for achieving higher accuracy of the results.

The proposed crawler is divided into 2 levels: Site Locating and In-Site Locating. The Site Locating Level is used to achieve wider focus of the crawler while the In-Site Level is used for more focused domains [6]. The two levels are as below:

1. Site-Locating Technique is used to search for the higher-level webpages using the reverse search technique. In this, the crawler uses search engines such as google to direct towards relevant links called reverse searching technique. The links are then separated into more relevant sources using a two-level site prioritizing technique.
2. In-Site Technique is used to prioritized and search through relevant sites depending on the topic domain. The technique searches through the insides of links resulting in more accurate and efficient result findings.
3. Once Crawling activity is completed, a consolidated report is being generated to give the entire crawled URLs specific to a domain and details of the errors found while crawling.

V. PROPOSED SYSTEM

There are two major functions of the intelligent focus crawler: 1) Site Tracing and 2) Inter-site Traversing.

In the first step, the crawler depending on the defined criteria identifies the relevant web pages and then in the second stage it crawls these identified web pages to discover and explore the frontier [8] found earlier. Precisely, the first stage begins with the given topic from user which is a specific topic. After that it achieves converse searching of frontier which is obtained by search engine.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 9, September 2017

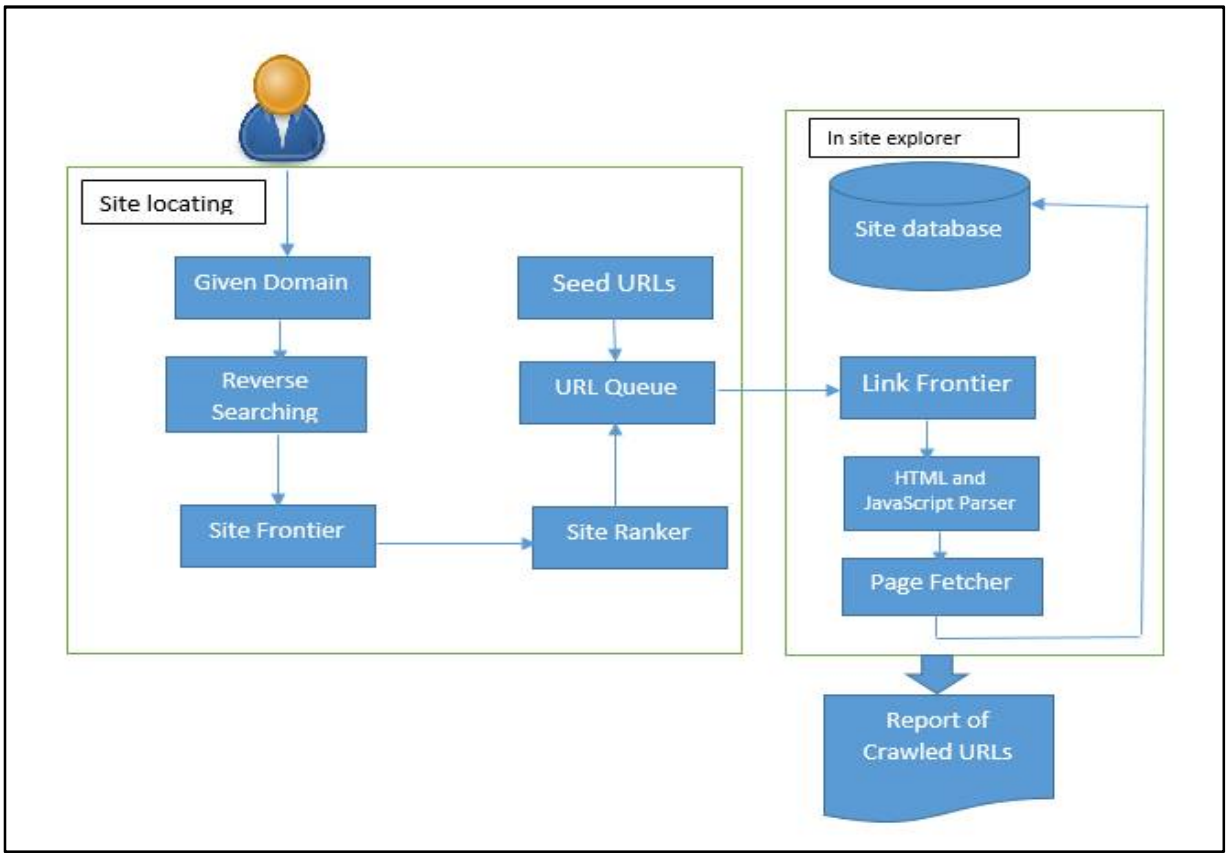

Figure 2. Process Flow of Site Locating and In-Site Locating

A. Site Locating

As shown in figure 2, in this step the crawler identifies the most relevant sites depending on the defined user criteria's. For this purpose, it uses a general-purpose search engine. It contains site identification, then ranking of the sites based on relevance and then classifying them for future easy access.

Web Site Assembling: While the traditional crawler surveys all recently found list of URLs, the intelligent Focus Crawler minimizes the number of visited URLs by using converse searching approach. To accomplish this objective, it is not sufficient to only use the hyperlinks in a downloaded page because a website contains a small number of hyperlinks to external websites. Even if the website is large there are maximum 5% hyperlinks redirecting to an external website.

Thus, finding external hyperlinks [9] from already visited and downloaded webpages is not sufficient for the Site Frontier. In the given project to identify additional webpages, two crawling strategies i.e. incremental two-level site prioritizing and reverse searching are used.

B. Converse Searching

The concept1 is to use existing search engines, such as Google, Baidu, Bing etc., to find frontier of unvisited sites. Search engine rank websites, according to relevancy and hence most relevant pages are displayed. Algorithm 1 defines the process of converse searching.

A converse search is activated:

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 9, September 2017

1. When the crawler bootstraps.
2. When user enters a specific topic.

For applying this strategy, the user enters a specific topic as an input. The system will then use a generic search engine facility such as google.com, bing.com, etc. to find relevant web page. For example, when an input is entered the crawler will list all the web pages that are directed to the Google Homepage by searching Google (google.com) for relevant information. While doing this is the first page of the search engine is parsed to identify and extract the relevant hyperlinks2.

C. HTML Parsing

After retrieving HTML code, it should be parsed. It uses cstring to store data, to not waste time to copy it into a string, the program use cstring functions directly on the Crawler structure: The algorithm search every occurrences of href tags (href=) in the cstring, then chars are stored until the first non-URL char: ; ; ; and space. The? and chars are added to this list to not store anchors and to ignore cookies set for a same page. To be the most efficient, a regular expression (regex) could have been used, but no function provides regex searches [10] in string nor in string.h, so is another library being used. With my way, links who are not included in href tags of which are using a non-strict syntax (like with simple quotations, of with spaces before or after the equal) will just not be crawled.

D. URL Parsing

The parsed URL's[2] are stored in C strings. They are parsed in turn, removing the 'http://' part, which is of no use and does not need to be stored, and eventually the 'www.' part is also removed. URL parts, bounded by / chars, are then stored in the tree for which string manipulation functions are used. While this is happening, the URL parts are stored and the URL parsing is initiated using a vector of strings. Storing parsed URLs, the program uses a vector iterator [11] to browse the various parts stored in the parsed URL. Each part is the parent of the deeper one, using the tree function to avoid the problem of duplicated elements, a temporary tree for each URL is been chosen, and then to use the merge function, with the option duplicate leaves to false. Using this pre-made function was the simplest and the most optimized way. All the URLs are stored in dictionary structure [12] to maintain the uniqueness of the URLs.

E. Recursive Browsing [13]

The URL are parsed and stored according to the process of HTML parsing. To be strictly recursive, it would have been required to retrieve and parse HTML data at every new URL found in a page, but it would also have caused memory problems, because of HTML data accumulation in memory. My solution is to make a first iteration to store URLs in the URL the user entered, and after to browse the tree, reconstructing URLs and crawling those one. It's working because the new elements stored in the tree are appended at the end, so the next iteration will consider new URLs at the end. After finishing recursive browsing the sort function is called, to be able to display the tree or search results in alphabetical sorting.

VI. SYSTEM ANALYSIS AND PROPOSED ALGORITHM

In intelligent focus crawling system below steps are carried out:

Algorithm

Step 1: Select Domain of user's interest
Step 2: Apply reverse searching strategy
Step 3: Site Locating and In-Site exploring
Step 4: Parse HTML and JavaScript code
Step 5: Generate a consolidate report of all the fetched URLs and maintain error log.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 9, September 2017

Proposed Algorithm

The algorithms used in intelligent Focus Crawling technique and the implementation process details are described in this paper,

A. Converse Searching

Input: Seed Sites and Domain specific topic:

1. Enter specific topic
2. Use search Engine and get relevant webpages
3. do
4. pick a relevant webpage;
5. site getExtractedPages(siteDatabase, seedSites);
6. resultPageconverseSearch(site);
7. links GetAllHyperlinks(resultPage);
8. for each Hyperlink in links do
9. PageDownload(link);
10. CheckRelevance(page);
11. if relevant, Then;
12. extractUnvisitedSiterelevantSites(page)
13. Output:relevantSites;

End

B. In-Site Exploration

1. input:siteFrontier
2. output: searchable forms and out-of-site links
3. BFS (topic, starting-urls)
4. foreach link (starting-urls)
5. enqueue (frontier, link, 1);
6. while (visited MAX-PAGES)
7. link := dequeuetop link(frontier);
8. doc := fetch(link);
9. score := sim(topic; doc);
10. enqueue(frontier; extract links(doc); score);
11. if (frontier > MAX BUFFER)Then
12. dequeue bottom links(frontier);
13. end if
14. end while
15. End For

VII. CONCLUSION

Proposed System had demonstrated that goal-directed crawling is a powerful means for topical resource discovery. Reverse searching algorithm helped to provide more accuracy. HTML Parser and JavaScript parser helped to handle dynamic contents. Generated consolidated report helped user to get more information about crawled URLs and Errors if any.

ACKNOWLEDGMENT

I would like to take this opportunity to express my heartfelt gratitude to my guide, Prof. Pradnya Kasture and Head of Department, Prof. Vina M. Lomte, Department of Computer Engineering, RMDSSOE, Savitribai Phule Pune

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 9, September 2017

University, for her kind cooperation, constant encouragement, suggestions and capable guidance during the research, without which it would have been difficult to proceed with.

REFERENCES

[1] Zhao F, Zhou J, Nie C, Huang H, Jin H. Smart Crawler: A Two-stage Crawler for Efficiently Harvesting Deep-Web Interfaces. in IEEE Transactions On Services Computing. 2016, 9 (4), pp.
[2] Chakrabarti S, Berg MVD, Dom B. Focused crawling: A new approach to topic-specific web resource discovery. Computer Networks. 1999, 31 (11), pp. 1623-1640.
[3] Denis S. on building a search interface discovery system. International Workshop on Resource Discovery. 2009, pp. 81-93.
[4] Barbosa L, Freire J. an adaptive crawler for locating hidden web entry points. Proceedings of the 16th international conference on World Wide Web. 2007, pp. 441-450.
[5] Stamatakis K, Karkaletsis V, Paliouras G, Horlock J, Grover C, Curran JR, Dingare S. Domain-Specific Web Site Identification: The CROSSMARC Focused Web Crawler. Second International Workshop on Web Document Analysis.GB:United Kingdom,Edinburgh. 2003, pp. 75-78.
[6] Dong H. Hussain FK. Self-: Adaptive Semantic Focused Crawler for Mining Services Information Discovery. IEEE Transactions on Industrial Informatics. 2014, 10 (2), pp. 1616-1626.
[7] He Y, Xin D, Ganti V, Rajaraman S, Shah N. Crawling deep web entity pages. Proceedings of the sixth ACM international conference on Web search and data mining. 2013, pp. 355364.
[8] Dincturk ME, Jourdan GV, Bochmann GV, Onut IV. A model-based approach for crawling rich internet applications. ACM Transactions on the Web (TWEB). 2014, 8 (3), pp. 139.
[9] Acevedo J, Agosto G, Zuniga MOD. Professor Morley Mao, Web Crawler. University of Michigan EECS 489. 2005, pp. 1-13.
[10] Susan D, Hao C. Hierarchical classification ofWeb content, Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. 2000, pp. 256-263.
[11] Open directory project. https://www.resource-zone.com/. Date accessed: 17/03/2017.
[12] Cope J, Craswell N, Hawking D. Automated discovery of search interfaces on the web. Proceedings of the 14th Australasian database conference. 2003, 17, pp. 181-189.
[13] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The weka datamining software: An update. SIGKDD Explorations Newsletter. 2009, 11 (1), pp. 10-18.