A review of web crawling approaches

Elda Xhumari[a], Izaura Xhumari[b]

[a]Universtity of Tirana, Department of Informatics, Boulevard "Zogu I", Tirana, 1001, Albania
[b]Universtity of Tirana, Department of Informatics, Boulevard "Zogu I", Tirana, 1001, Albania

Abstract

Websites are getting richer and richer with information in different formats. The data that such sites possess today goes through millions of terabytes of data, but not every information that is on the net is useful. To enable the most efficient internet browsing for the user, one methodology is to use web crawler. This study presents web crawler methodology, the first steps of development, how it works, the different types of web crawlers, the benefits of using and comparing their operating methods which are the advantages and disadvantages of each algorithm used by them.

Keywords

Web crawler, Algorithms, Types of web crawlers

1. Introduction

The world wide web is a large collection of data. Data that continue to grow day by day. Nowadays it has become an important part of human life to use the internet to gain access to information on the World Wide Web. Due to bandwidth, storage capacity, limited computer resources and the rapid growth of the World Wide Web, unforeseen scaling challenges have arisen for search engines. The two most important features of the web such as the large volume of data and the speed of their change pose a difficulty for web crawling, as there are a large number of pages which are added, changed and deleted every day. Although search engine technology has dramatically scaled up to keep pace with the rise of the Web, these general-purpose search engines and crawlers have encountered some limitations as follows: [1]

- It is impossible for them to index and analyze all pages and keep these search indexes up to date.
- They may return hundreds or more links to a user's query, due to misunderstanding of the query pages run by these links may not be closely related to the user query.
- They may not meet query requirements with different backgrounds, purposes and periods.
- Dynamic content, such as news and financial data, on the Web is growing and changing frequently. Many search engines can take up to a month to refresh their indexes across the Web. Therefore, query results are probably not valid at the time the request is made.

Therefore, it is necessary for a technology which enables fast crawling search to assemble web pages with the highest possible content and quality and keeping these pages up to date. This "problem" can be solved using indexes which are built by a web crawler. A web crawler, otherwise known as a "web spider", is a program that browses the World Wide Web by "clicking" on any link they find and collects information found automatically. However, building a large index for web pages is not the only web crawler application [1].

2. How it works?

The crawler maintains a list of unvisited URLs called frontier. The list is first initialized with URLs provided by a user or other program. Each crawl cycle involves selecting a URL from the list and

Proccedings of RTA-CSIT 2021, May 2021, Tirana, Albania
EMAIL: elda.xhumari@fshn.edu.al (E. Xhumari); izaura.xhumari@fshn.edu.al (I. Xhumari)

retrieving the corresponding page for that URL via HTTP, analyzing it to extract URLs and specific information, and finally adding these unvisited URLs to the frontier list. Before being added to the list these URLs may be marked a point depending on the benefit achieved if the page with the corresponding URL is visited. The crawl process may end when a certain number of crawled pages are accessed. If the crawler is ready to visit another page and the frontier list is empty then the situation signals a dead end for the crawler and since the crawler no longer has new pages to visit it stops.

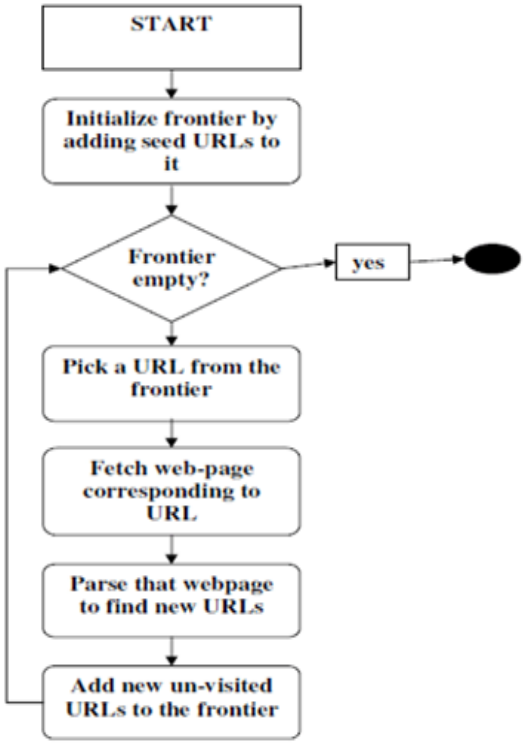

Figure 1: The Data Flow of a Crawler

3. Types of web crawler

Different types of web crawlers are available depending on how the web pages are crawled and how the future web pages are retrieved and accessed. Some of which are as follows.

A. Incremental Crawler

An incremental web crawler is one of the traditional crawlers, which constantly updates an existing set of downloaded pages instead of restarting the crawling process from scratch each time. This includes some way to determine if a page has changed since it was last downloaded. Pages can appear multiple times in the crawler order, and crawling is an ongoing process that conceptually never ends. To have an updated content of downloaded web pages, an incremental web crawler links the review of previously downloaded pages to the first visit to new pages [2]. The goal is to achieve updating and coverage at the same time. The advantage of an incremental web crawler is that only valuable data is provided to the user, thus the network bandwidth is stored and data enrichment is achieved.

B. Form Focused Crawler

Form Focused Crawler deals with the rare distribution of forms on the Web. Form Crawler avoids crawling through unproductive links by restricting search to a specific topic, learning the characteristics of links and pages that lead to pages containing searchable forms, and using appropriate stopping criteria. Web crawler uses two rankings: site and links to guide its search. Later, a third classifier: the shape classifier is used to filter out useless forms.

C. Focused Crawler

A Focused Crawler collects documents which are specific and related to the given topic. Sometimes this crawler is also known as Topic Crawler to approach how it works. Focused Crawler is a web crawler that tends to transfer pages that are related to each other. Determines if the given page has similarities to the specific topic. One of the advantages of the focused crawler is the economic flexibility in hardware and network resources. It reduces the amount of network traffic, logging and downloads [3]. Focused Crawler searches, acquires, indexes, and maintains pages for specific groups of topics that represent a relatively narrow segment of the network. This crawler is run by a classifier that learns to recognize the importance of taxonomy embedded examples, and a distiller that identifies current online priority points.

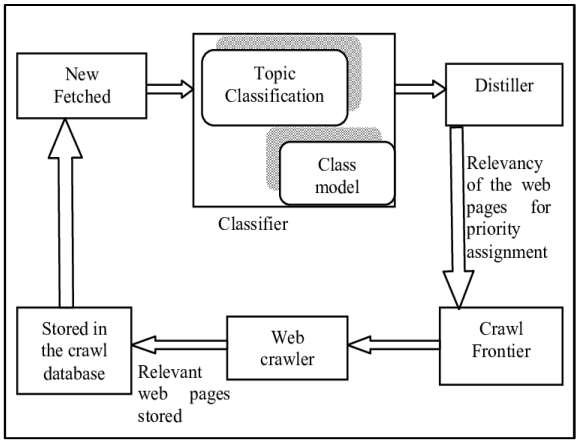

Figure 2: General architecture of Focused Web Crawler

Focused Crawler is a new approach to increasing accuracy and expert internet search. An ideal focused crawler could only download those pages that are related to the topic while ignoring other pages and would anticipate the possibility of a link to a specific topic related to the topic before downloading it. Focused Crawler has three main components: a classifier that makes important judgments on crawled pages to decide on the extension of downloaded links, a distiller that sets a crawl center measure to determine visit preferences, and a crawler which has dynamically reconfigurable priority controls dominated by the classifier and distiller.

Focused crawler aims to provide a simpler alternative to overcoming the issue that instant pages which are low ranking related to the topic in question. The idea is to recursively execute an exhaustive search to a certain depth, starting with the relatives of a highly ranked page [4].

D. Parallel Crawler

As the size of the internet grows, it becomes difficult to retrieve the entire or a major portion of the web employing a single method. Therefore, several search engines typically run multiple processes in parallel to perform the above task, so download rate is maximized. This kind of crawler is known as a parallel crawler [5]. We can also say that when multiple crawlers are usually run in parallel, it's referred as Parallel crawlers. A parallel crawler consists of multiple crawling processes referred to as C-procs which can run on network of workstations [6]. The Parallel crawlers rely on Page freshness and Page selection. A Parallel crawler may be on local network or be distributed at geographically different locations. Parallelization of the crawling system is extremely important from the purpose of read of downloading documents in an affordable quantity of time.

E. Distributed Crawler

Distributed Web Crawler is a distributed computing technique. Many crawlers are operating for distribution within the web crawling method to master as much web coverage as possible [7]. A central server manages the communication and synchronization of nodes, as it is geographically distributed. Mainly uses PageRank algorithm to increase its efficiency and quality search. The advantage of distributed web crawler is that it is not affected by system crashes or various events and can be adapted by many crawling applications.
To design an efficient web crawler, it is required to create the distribution task between multiple machines in a synchronous process. Large websites should be distributed individually on the network and they should provide the right chance and rationality for synchronous access. Meanwhile synchronous distribution saves network bandwidth resources [4].

4. Web crawling algorithms

i. Breadth First Search

It starts with a small set of pages and then explores other pages following the links in the first width. Indeed, websites are not strictly traversed at first glance, but can use a variety of policies. For example, it may crawl the most important pages first. This method is used by many search engines. This crawler balances the load between servers. Breadth first algorithm work on a level by level, i.e. algorithm starts at the root URL and searches the all the neighbors URL at the same level. If the desired URL is found, then the search terminates. If it is not, then search proceeds down to the next level and repeat the processes until the goal is reached. When all the URLs are scanned, but the objective is not found, then the failure reported is generated. Breadth first Search algorithm is generally used where the objective lies in the depthless parts in a deeper tree [8].

ii. Depth First Search

This is an algorithm for traversing or searching tree or graph data structures. It is a technique of systematically examining the search starting from the root node and penetrating deeper through the child node. If there is more than one child, then priority is given to the child on the left and penetrates deeply until there are no more children available. Returns to the other unexplored node and then proceeds in a similar manner. This algorithm ensures that all edges are visited at once. It is suitable for search problems, but when the branches are large, then this algorithm can end up in an endless loop.

iii. Best First Search

Best first algorithms are often used to find search paths. Best First Search is a search algorithm that roams a graph starting from the most promising node selected according to a specified rule. The basic idea is that having a URL limit, the best URL according to some evaluation criteria such as accuracy, recall, accuracy, and points (F-Score). In this algorithm, the URL selection process is driven by lexical similarity between the topic keywords and the URL source page. Thus, the similarity between the page and the topic keywords is used to evaluate the fit with all the outbound links of the page.

```
insert in ready queue(seeds)
while true do
if more links in ready queue then
link := dequeue best
doc := fetch(link)
score := apply rule(doc)
out links := extract links(doc)
save score(out links, score)
else
sorted links := sort(non processed queue)
insert in ready queue(sorted links)
end if
end while
```

Figure 3: The best-first algorithm pseudo-code

iv. Fish Search Algorithm

The main principle of the algorithm is: it takes as input an initial URL and a search query, and dynamically builds a list of priorities (initialized with the initial URL) of the next URLs (referred to as nodes) to be explored. In each step the first node is removed from the list and processed. As the text of each document becomes available, it is analyzed by a scoring component assessing whether it is relevant or irrelevant to the search query (value 1-0) and, based on that result, a heuristic decides to pursue the search in that direction or not: Whenever a document source is retrieved, it is scanned for links. Nodes run by these links are assigned a depth value. If the parent is important, the depth of the children is set to a predetermined value. Otherwise, the depth of the children is set to be one less than the depth of the parent. When the depth reaches zero, the direction is interrupted and none of his children are included in the list [9].

v. Shark-Search Algorithm

Fish-Search algorithm's main flaw is that the interrelated computation is too simple, it only has 0 and 1, interrelated and irrelevant respectively. Secondly, every
node's potential score has a low precision which only has three situations (0,0.5, and l). Aimed at these disadvantages, Michael Hersovici [10] brought forward an improved Shark-Search algorithm which mainly ameliorates page, interrelated query computation and potential score' s computing method. The following process is its detail:

- Import vector space model to compute the page and user query's relativity.
- Consider the information given by anchor text near the hyperlink and compute the relativity between it and user's query.
- Calculate both of the above two factors with child node's potential score computing formula.

Through these betterments, Shark-Search algorithm's efficiency is much better than Fish-Search's [2].

vi. Page Rank Algorithm

In Page rank algorithm web crawler decides on the importance of web pages in each web site through the total number of links or citations per page. Page rank is calculated according to the

Relatedness between web pages by the Page Rank algorithm. Website ranking calculation:

$$PR\,(A) = (1-d) + d\left(\frac{PR(T_1)}{C(T_1)} + \cdots + \frac{PR(T_n)}{C(T_n)}\right), \quad (1)$$

where *PR(A) - Page Rank of a given Page,D – Dumping factor*
T_1 – links.

To find the Page Rank for a page, called PR (A), you must first find all the pages that link to page A and Out Link from A. If we find a page T_1 that links with A then page C (T_1) will give the number of outbound links on page A. The same procedure is done for pages T_2, T_3 and all other pages that can be linked to the main page A - and the sum of their values will provide the Page Rank of the website [11].

Table 1
Advantages and limitations of web crawling algorithm

Algorithm	Advantages	Limitations
Breadth First Search	Suitable for situations where the solution is located at the beginning in a deep tree.	If a solution is far away then it consumes time. Consumes a large amount of memory.
Depth First Search	Suitable for in-depth search problems. Consumes very little memory.	If the edges are large then this algorithm can end in an endless cycle.
Fish Search	The algorithm is helpful in forming the priority table.	The usage of resources of network is high. Fish search crawlers significantly load not only network, also web servers.
Shark Search	The algorithm mainly ameliorates page, interrelated query computations and potential score's computing method.	The usage of resources of network is high.
Page Rank	In a short time, the most important pages are returned as Rank is calculated on the basis of the popularity of a page.	Favors older pages, because a new page, even a very good one, will not have many links unless it is part of an existing web site.

5. Approach

A traditional crawler worked simply by extracting static data from HTML code and most websites until recently would undergo the same crawler process. The crawling process is no longer as simple as it was a few years ago, due to the increasing use of JavaScript frameworks such as Angular, React, Meteor. Many of the websites are JavaScript heavy and generates content by doing asynchronous JavaScript calls after page is loaded. The use of these frameworks makes developer life simpler and provides many benefits for creating dynamic sites. To crawl this type of web sites Web Crawlers, use Selenium.

Selenium is a Web Browser Automation Tool originally designed to automate web applications for testing purposes. It is now used for many other applications such as automating web-based admin tasks, interact with platforms which do not provide API, as well as for Web Crawling.

Building a focused web crawler using selenium tool is good way to collect useful information. Focused Crawler is an approach to increase accuracy and expert internet search. An ideal focused crawler could only download those related pages by ignoring other pages and would anticipate the possibility of a link to a specific topic related site before downloading it.

One use case of a focused web crawler is extracting financial data. Financial market is a place of risks and instability. It's hard to predict how the curve will go and sometimes, for investors, one decision could be a make-or-break move. That's why experienced practitioners never lose track of the financial data. Financial data, when extracted and analyzed in real time, can provide wealthy information for investments and trading. And people in different positions scrape financial data for varied purposes.

6. Conclusions

Web Crawler is the essential source of information retrieval which roams the Web and downloads web documents that suit the user's need. Web crawler is used by search engines and other users to regularly ensure that their database is up to date. In this article has been presented a review of different types crawling technologies and algorithms, why "focused crawling" technology is being used. The crawling algorithm is the most important part of any search engine. Focused Crawlers uses more complex systems and

techniques to define the information of high relevance and quality. Searching algorithm is the heart of the search engine system. The choice of the algorithm has a significant impact on the work and effectiveness of focused crawler and search engine. In conclusion the focused crawler compared to different crawlers is intended for advanced web users focuses on specific topic and it does not waste the resources on irrelevant material.

7. References

[1] Mahmud, Hasan & Soulemane, Moumie & Rafiuzzaman, Mohammad. (2011). A framework for dynamic indexing from hidden web. International Journal of Computer Science Issues. 8.

[2] Su Guiyang, Li Jianhua, Ma Yinghua, Li Shenghong, Song Juping Department of Electronic Engineering, Slanghai Jiaotong University, Shanghai 200030, P. R. China (Received April 10, 2004) New Focused Crawling Algorithm

[3] Christopher Olston, Marc Najork, Web Crawling

[4] Gautam Pant, Padmini Srinivasan, Filippo Menczer, Department of Management Sciences School of Library and Information Science, The University of Iowa,2004 Crawling the Web (4-6), Web Dynamics

[5] Dhiraj Khurana, Satish Kumar, "Web Crawler: A Review", IJCSMS International Journal of Computer Science & Management Studies, Vol. 12, Issue 01, January 2012.

[6] Trupti V. Udapure, Ravindra D. Kale, Rajesh C. Dharmik, "Study of Web Crawler and its Different Types", IOSR Journal of Computer Engineering (IOSR-JCE), Volume 16, Issue 1, Ver. VI (Feb. 2014), PP 01-05.

[7] Yugandhara Patil, Sonal Patil, Janar 2016 Review of Web Crawlers with Specification and Working Vol. 5, Issue 1, January 2016

[8] Junghoo Cho and Hector Garcia-Molina ―Effective Page Refresh Policies for Web Crawlers‖ ACM Transactions on Database Systems, 2003.

[9] Andas Amrin*, C. X. (2015). Focused Web Crawling Algorithms. Shanghai, China.

[10] Michael Hersovici, Michal Jaoov, Maarek Yoelle S, et al.The Shark-Search algorithm. An applicatication: Taibred Web Site Mapping, Computer Networks and ISDN Systems 30, 1998. 317-326

[11] TIAN Chong "A Kind of Algorithm For Page Ranking Based on Classified Tree In Search Engine" Proc International Conference on Computer Application and System Modeling (ICCASM 2010)