A Study on Web Crawlers and Crawling Algorithms

Nay Chi Lynn
University of Computer Studies, Mandalay
naychelynn@gmail.com

Su Su Aung
University of Computer Studies, Mandalay
susuaung87@gmail.com

Abstract

Making use of search engines is most popular Internet task apart from email. Currently, all major search engines employ web crawlers because effective web crawling is a key to the success of modern search engines. Web crawlers can give vast amounts of web information possible to explore the web entirely by humans. Therefore, crawling algorithms are crucial in selecting the pages that satisfy the users' needs. Crawling cultural and/or linguistic specific resources from the borderless Web raises many challenging issues. This paper will review various web crawlers used for searching the web while also exploring the use of various algorithms to retrieve web pages.

Keyword: Web Search Engine, Web Crawlers, Web Crawling Algorithms.

1. Introduction

Professional search engines act as a search middleware for end users or customers and try to figure out in an interactive dialogue with the system and the customer, what the customer needs, and how this information should be used in a successful search. There are two major categories of searching tools on the Web: directories (Yahoo, Netscape, etc.) and search engines (Lycos, Google, etc.). It is hard to use directories with the increase of web sites. Web crawling is the process by which we gather pages from the Web in order to index them and support a search engine. The objective of web crawling is to gather quickly and efficiently as many useful web pages as possible, together with the link structure that interconnects them. Search engines cannot meet every search requirement. The search engine techniques may become useless or junky if the information it draws are not attracting users, especially if the malicious user who are trying to attract more traffic in to their site by embedding the most used keywords invisibly in to their site. The challenges are relevancy, robustness and the ability to download large number of pages. A web crawler (also known as a web spider or web robot) is a program which automatically traverses websites, downloads documents and follows links to other pages. It keeps a copy of all visited pages for later uses. Many web search engines use web crawlers to create entries for indexing. In general, the crawler starts with a list of web sites to visit (URLs), called the seeds. As it visits them, it identifies all the links to other pages hyperlinks) in the page and adds them to a list of other sites it needs to visit, called the crawl frontier. When done with one site, it continues to visit the entire next site on the list, choosing which is "next" by a set of policies - it may not be just the one last it saw, or the first on the list.

The later part of the paper deals with describing challenges of web crawlers, different types of web crawlers, various crawling

algorithms and the summary of web crawlers and crawling algorithms.

2. Related Work

Different algorithms with different metrics have been suggested to lead a crawl towards high quality pages.

Shaojie Qiao [10] proposed a new page rank algorithm based on similarity measure from the vector space model, called SimRank, to score web pages. They pro-posed a new similarity measure to compute the similarity of pages and apply it to partition a web database into several web social networks (WSNs).

Tian Chong [11] proposed a new type of algorithm of page ranking by combining classified tree with static algorithm of PageRank, which enables the classified tree to be constructed according to a large number of users' similar searching results, and can obviously reduce the problem of Theme-Drift, caused by using PageRank only, and problem of outdated web pages and increase the efficiency and effectiveness of search.

In Pann Yu Mon, Chew Yew Choong and Yoshiki Mikami [6], a language focused crawler designed for Myanmar language is proposed and its architecture and performance are presented. The main feature of LSC is that it can identify the various encodings of Myanmar Web pages. Through experiment, the LSC has successfully downloaded a satisfactory number of Myanmar Web pages. Several measures, such as accuracy rate of language identifier, the performance and the recall rate of the LSC, are presented.

In [5], Saeko Nomura, Satoshi Oyama and Tetsuo Hayamizu proposed the integration of the two improvement methods. This method gives good results. Not only can relevant pages be extracted but also those pages are loaded on principal eigenvectors or the non principal vectors in low computation cost for broader types of topics.

Wenxian Wang, Xingshu Chen, Yongbin Zou, Haixhou Wang and Zongkun Dai [12] proposed an efficient crawler based on Naïve Bayes to gather many relevant pages for hierarchical website layouts.

3. Challenges

Although the area of web crawlers and crawling algorithm is a mature research area, there are still rapid changes in web technology and the usages of web crawler become vary so much that web crawling faces new challenges today. There are many open questions and issues such as

- How to select of appropriate level of greediness for a scoped crawler?
- How to select the method to discard or retire unwanted pages?
- Integration of theory and systems work, and
- Deep web, the science and practice of deep web crawling is in its infancy.

There are also remaining several research areas such as vertical crawling, crawling scripts, personalized content and Collaboration between content providers and crawlers.

4. Web Crawlers

Web crawlers work in a recursive or loop fashion. Specifically the crawler iteratively performs the following process:

1. Download the Web page.

2. Parse through the downloaded page and retrieve all the links.
3. For each link retrieved, repeat the process.

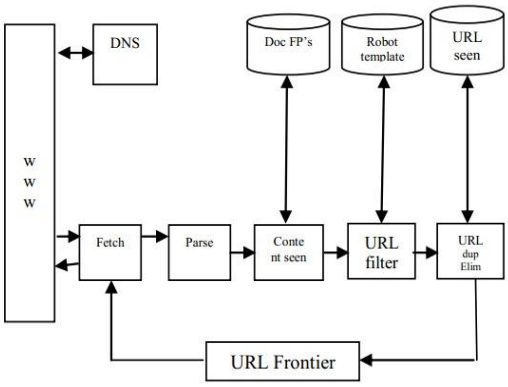

Figure 1. Architecture of a Basic Web Crawler [3]

The crawlers 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 as follows:

4.1. Focused Crawler

A *focused crawler* is a web crawler that collects Web pages that satisfy some specific property by carefully prioritizing crawl frontier and managing the hyperlink exploration process e.g., "crawl pages with large PageRank", or "crawl pages about baseball". A focused crawler must predict the probability that an unvisited page will be relevant before actually downloading the page. In contrast to traditional approaches, a focused crawler [2] efficiently seeks out documents about a specific topic and guides the search based on both the content and link structure of the web. It implements a strategy that associates a score with each link in the pages it has downloaded.

4.2. Incremental Crawler

Incremental Crawlers are search crawlers that gather the changes made since the last crawl. An incremental crawler visits the web repeatedly after a specific interval for updating its collection. For each incremental crawl run that happens, the freshness of content decreases and decreases because the crawler needs to catch up, but the content is getting updating faster than the crawler can crawl them. Thus will eventually lead to incremental crawl and will be updating content that was made during incremental crawl, as long as the content output continues. Once the content output slows, then the incremental crawl will catch up. The interval doesn't matter there because the incremental crawl will take the time it needs to finish.

4.3. Continuous Crawler

A *continuous crawler* is created to try to help keeping your search index more up to date. While the frequencies at which requests are made have increased, the maximum number of simultaneous requests on one repository/host will still be controlled by "Crawl Impact Rules" (which define the maximum number of simultaneous threads that can make requests). Continuous crawl foot print is similar to incremental crawl. Multiple continuous crawls can run at the same time (for the same content source) and they update the index continuously.

4.4. Parallel Crawler

A *parallel crawler* is a crawler that runs several processes in parallel. The goal is to maximize the download rate while minimizing the overhead from parallelization and to avoid frequent downloads. A parallel crawler consists of multiple crawling processes. Each

process performs the basic tasks that a single process crawler conducts. It downloads pages from the Web, stores the pages locally, extracts URLs from the downloaded pages and follows links.

4.5. Distributed Crawler

Distributed web crawling is a distributed computing technique where by Internet search engines employ many computers to index the Internet via web crawling. Such systems may allow for users to voluntarily offer their own computing and bandwidth resources towards crawling web pages. By spreading the load of these tasks across many computers, costs that would otherwise be spent on maintaining large computing clusters are avoided [14].

5. Web Crawling Algorithms

One of the main desirable features a crawler should have is the ability to download important pages first. The crawler should fetch not just relevant pages, but high quality relevant pages. It is important to consider the algorithm by which web sites are crawled. These are common algorithms for creating web crawlers.

5.1. Breadth First Search Algorithm

Breadth first algorithm starts its crawling process right from the root node and sweeps down searching for the related neighboring 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 [7].

5.2. Depth First Search Algorithm

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 [7].

5.3. Page Rank Algorithm

PageRank algorithm determines the importance of the web pages by counting citations or backlinks to a given page [13].

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: [3]

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

Where

PR (A)→ Page Rank of a website,

d→ damping factor,

T1…..Tn→ Link

5.4. HITS Algorithm

This algorithm uses scores to calculate the relevance of a page. This method retrieves a set of results for a search and calculates the authority and hub score within that set of results. Because of these reasons this method is not often used [1].

Joel C. Miller et al [4] proposed a modification on adjacency matrix input to *HITS algorithm* which gave intuitive results.

5.5. Genetic Algorithm

Genetic algorithm is based on biological evolution where by the fittest offspring is obtained by crossing over of the selection of some best individuals in the population by means of fitness function. In a search algorithm solutions to the problem exists but the technique is to find the best solution within specified time [8].

The genetic algorithm is presented as best suited algorithm when the user has literally no or less time to spend in searching a huge database and also very efficient in multimedia results. While almost all conventional methods search from a single point,

Genetic Algorithms always operates on a whole population. This contributes much to the robustness of genetic algorithms.

6. Comparison of Web Crawlers and Crawling Algorithms

Crawling process can improve the quality of services provided by search engine. Optimal crawlers and crawling algorithm play vital role in determining the quality and freshness of web pages. I have studied on the types of web crawlers and web crawling algorithms and also made comparisons with their pros and cons, and also strengths and weaknesses respectively.

Table 1. Pros and Cons of Crawlers

Web Crawler	Pros	Cons
Focused Crawler	-Spends less response time and effort for processing web pages	-The problem of zero probability and find out the relevancy of unvisited URLs
Incremental Crawler	-Allow re-visitation of pages at different rates	- Cannot run in parallel - Allow Deep change will not result in degraded freshness
Continuous Crawler	- Allow Changes will continue to be processed in parallel	- Increase load marginally on the host
Parallel Crawler	-Scalability -Network load dispersion -Network load reduction	-Require redundancy storage
Distributed Crawler	-Reduce hardware necessities and increases overall download speed and reliability	-Web partitioning/ repartitioning and data center placement are required

Table 2. Strengths and Weakness of Crawling Algorithms

Crawling Algorithms	Strengths	Weakness
Breadth First Search	- Suited for situations where objective is found on shallower parts in a deeper tree	- Not perform so well when branches are so many in a game tree - Needs more space to store all visited in each node level
Depth First Search	-Only needs to store visited pages in one	-When the branches are large then it

	web graph	might end up in an infinite loop
Page Rank	-In the very limited time, important pages are downloaded - In high Page Rank, pages are always good in quality	- Prone to adversarial manipulation. - Create a large number of auxiliary pages and hyperlinks.
HITS	-Relevance scoring method	-Does not consider the content of the pages at all; generates topic drift phenomena -Leans to old web pages and ignores the new ones
Genetic	-Optimization in search - Scheduling and timetabling	-Weighted selection of attributes

7. Conclusion

Most of challenges in these architectures are minimizing network bandwidth consumption rate, keeping the up-to-date database and improving the quality of search pages. As overall, I would like to recommend for Focused Crawler due to its smallest response time and effort, and Genetic algorithm because of its overall download speed and reliability to produce effective and relevant results. I hope that all of the algorithms reviewed in this paper are effective and supportive for future web search.

References

[1] Alessio Signorini, "A Survey of Ranking Algorithms" retrieved from http://www.divms.uiowa.edu/~asignori/phd/report/a -survey -of -ranking-algorithms.pdf 9/11/2011.

[2] Chakrabarti, S., van den Berg, M. & Dom, B., 1999a Focused crawling: a new approach to topic-specific Web resource discovery. In Proceeding of the 8th International conference on World Wide Web, Toronto, Canada, pp.1623.

[3] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze, "Introduction to Information Retrieval" Cambridge University Press, 2008, Ch 20, pg.405-416.

[4] Joel C. Miller, Gregory Rae, Fred Schaefer "Modifications of Kleinberg's HITS Algorithm Using Matrix Exponentiation and Web Log Records" Proc. SIGIR'01, ACM 2001.

[5] Nomura, S., Oyama, S., Hayamizu, T.: Analysis and Improvement of HITS Algorithm for Detecting Web Communities. Journal of Systems and Computers 35(13) (2004).

[6] Pann Yu Mon, Chew Yew Choong, Yoshiki Mikami, "Language Specific Crawler for Myanmar Pages". In Proceedings of the 11th International Conference on Humans and Computers (HC 2008), Nagaoka, Japan, November 2008.

[7] Pavalam S M, S V Kashmir Raja, Felix K Akorli and Jawahar. A Survey of Web Crawler Algorithms, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 1, November 2011 ISSN (Online): 1694-0814.

[8] S. N. Sivanandam, S. N. Deepa "Introduction to Genetic Algorithms" Springer, 2008, pg 20.

[9] S.N. Palod, Dr S.K.Shrivastav,Dr P.K.Purohit "Review of Genetic Algorithm based face recognition" International Journal of Engineering Science and Technology (IJEST) Vol. 3 No. 2 Feb 2011.

[10] Shaojie Qiao, Tianrui Li, Hong Li and Yan Zhu, Jing Peng, Jiangtao Qiu "SimRank: A Page Rank Approach based on similarity measure" 2010 IEEE.

[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).

[12] Wenxian Wang, Xingshu Chen, Yongbin Zou, Haizhou Wang, Zongkun Dai "A Focused Crawler Based on Naive Bayes Classifier" Third International Symposium on Intelligent Information Technology and Security Informatics, 2010.
[13] Yongbin Qin and Daoyun Xu "A Balanced Rank Algorithm Based on PageRank and Page Belief recommendation"
[14]http://en.wikipedia.org/wiki/Distributed_web_crawling