Available Challenges and Guidelines in the Field of Deep Web and Intensive Crawling

Yasin Ezatdoost
Department of Computer Engineering, Guilan University
Rasht, Iran

Ali Tourani
Department of Computer Engineering, Guilan University
Rasht, Iran

Amir Seyed Danesh
Faculty of Computer Science and Information Technology
University of Malaya
50603, Kuala Lumpur, Malaysia

ABSTRACT

Today, there is a great deal of information available in Web world and the only way to access them is through search relationships. Web crawler is an automated script that independently browses the web. Web crawler starts its task with a "seed URL" and then traces links available in each page. This encountered many available crawlers with essential difficulties. Identification of search intermediate and selection of a proper inquiry, on one hand, and retrieving documentaries returned by the web as the result, on the other hand, are issues that intensify challenges available for web crawlers. The aim of the present paper is to investigate available challenges and guidelines in the field of deep web and intensive crawling.

General Terms

Algorithms, Software Implementation

Keywords

Intensive crawler, search engine, genetic algorithm, deep web

1. INTRODUCTION

Development of web and the increase in users' needs to access information available in web clarify the necessity and importance of search engines to facilitate accessing required resources. When search is performed on a search engine and search results are presented, users, in fact, see the outcome of the work of different parts in the engine. The engine has already made its database ready and it does not search the whole web at the moment. Google and any other search engine are not able to do this. All of the search in their own database while responding to a user's search word. With the help of its various sections, a search engine already collects and analyzes required information, stores the information in its database and searches in this database when the user's types a topic or word to search about [31]. Separate parts of a search engine include:

- Spider
- Crawler
- Indexer
- Database
- Ranker

A search engine can be divided into two general sections: online and offline. The online section takes the inquiries from users and sends proper pages to him/her from previously indexed and stored pages. The offline section is responsible for collecting and indexing web pages. Crawlers are used to retrieve and store web pages [29].these two sections of Search Engines are shown in Figure 1.

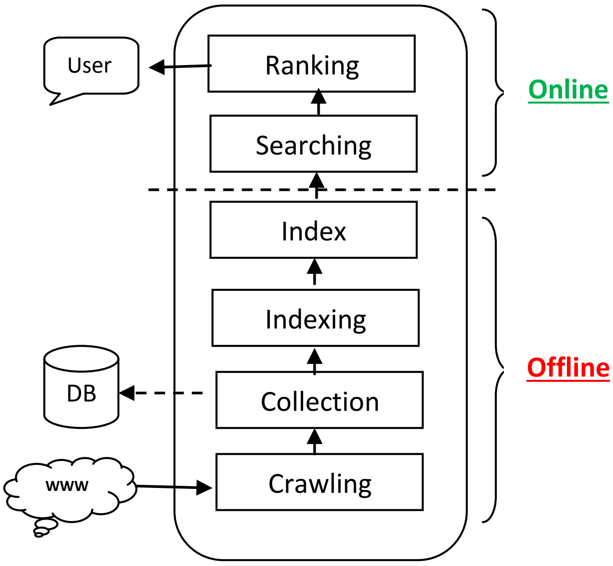

Figure1. Online and offline sections in Search Engines

According to studies in 2005, web contained more than 11.5 million pages at that time and every day approximately 7.5 million pages are adding [23]. Overall search engines cannot cover the great mass of pages with the high growth rate, and users demanding for some information see improper and out-of-date pages. Intensive crawlers are used to collect pages in a certain field. These crawlers face challenges such as local search difficulty and how to anticipate the quality of web pages before retrieving them. A highly efficient crawler possesses at least one of following tow specifications. First, it uses smart strategies in its decision-making (for example, choosing a link form non-selected links to retrieve related pages). Second, it has optimized supporting software and hardware structures to retrieve a great deal of pages per time unit [21, 22]. For the sake of this goal, resistance against errors and considerations related to web server resources must be added to the crawler. Research on the first specification includes strategies to detect important pages [8, 15], retrieval of documents related to a topic [4, 5, 9, 18], and re-crawling to maximize the novelty of web archive [6, 7]. But little work is done on the second specification because of difficulty of its implementation. In recent years, some search engines with certain domains are developed. These engines use intensive crawlers which only collect pages related to a certain topic. Intensive crawling was first proposed by "Chakrabati" in 1991[4].

Today, available crawlers use local search. To start the task first they take the address of considered pages related to a certain topic and retrieve them. Then, there is a set of repetitive steps which is considered as a score for each new retrieved page in each step and irrelevant pages (pages with lower score) will be omitted from the list of stored pages. Then links of remaining pages are educed and pages referred to by these links are retrieved. These tasks permanently continue and new pages are added to the set of stored pages in each step since the crawler uses a set of seed URLs in this way and only stores those pages which related to the seed URL through a direct or indirect link. This method is called "local search" and has three problems [17]:

1- Some sites relating to a similar topic may not have links to one another (for competitive reasons). Commercial and business sites are among these kinds of sites.

2- Links may be sideway. This means that there may be a link from first page to the second one, but there is no link from the second to the first. Thus, starting from the second site does not lead to the first one.

3- A group of web pages be separated from some pages relevant to a certain topic through some pages irrelevant to that topic. In this situation the crawler stops its task when it faces irrelevant pages and does not review next pages.

Most studies in the field of crawling related to determining the quality of retrieved pages and little work is done about the quality determination of links (anticipation of a page's quality before its retrieval), while efficiency of an intensive crawler depends on richness of links relevant to searched topic.

In the next section, related works on intensive crawling are reviewed and then a genetic algorithm-based method is explained to solve the problem of local search. Section 4 related to deep web and section 5 reviews challenges available for deep web search engines. Also, solutions proposed so far in the field are summarized. Section 6 is the conclusion of the present paper.

2. RELATED WORK

After proposing intensive crawling, the first question was that how one anticipate the quality of a web page before its retrieval. “Rungsawang"[19] et al proposed their own method as "learnable crawler". This is a crawler based on a knowledgebase and includes three parts: keywords, seed URLs and URLs' anticipator. First, keywords are entered by the user and are put in the knowledge base. Then, a common search engine, like Google, searches the words. Among found pages some (which have the highest priority) are selected and their address is recorded in the knowledgebase as the seed URL. The pages are retrieved and their links are traced. In each step, links of retrieved pages in the previous step are educed and a score is calculated for each of them. Links with higher score are traced earlier. Although this method is called "learnable crawling", but it functions like other intensive crawlers. The only difference is that primary and seed pages are not given to it and it finds them to start crawling. In method suggested by "Diligenti"[9] the second problem related to local search (sideway links) is solved to some extent. This method uses context graph and for each page given to crawler a context graph is developed the first node of which is the page itself (that is in zero layer). Then, pages having link to the first page are found using common search engines and here layer 1 is formed. Regression also repeats for these pages and the task continues until a predetermined number of layers are developed. Topics directly or indirectly related to the considered topic are gained using context graph, the relevant topics are then classified and every retrieved page goes into its own class. The main problem of this method is its high dependence to common search engines.

3. USING GENETIC ALGORITHMS

Usually when we face a very large search area and have little information about it, genetic algorithms are good methods to find an optimized solution. Since web is, also, a very large environment, these algorithms could be applied in intensive crawling. Genetic algorithm in an optimization calculative algorithm that effectively searches different parts of the answer zone by considering a set of answer zone points in each calculative repetition. Although the value of target function in search mechanism is not calculated in the whole points of answer zone, but the calculated value of target function for each point is involved in statistical average of the function (in all sub-spaces to which the point was related). The statistical average of target function is calculated in these sub-spaces. This leads the search toward zones having higher statistical average of target function in which it is more likely for the absolute optimized point to exist. Since in this method, despite all single-procedure methods, the entire answer zone is searched, there is little possibility convergence to an optimized local point.

3.1 Strengths of the Genetic Algorithm

- It is parallel.
- Search zone is searched in several different directions.
- It is possible to break down search zones into smaller zones.

"Qin" and "Chen"[17] proposed a method based on genetic algorithm to solve problems of local search. In their method, address of some pages are given to crawler to start, related pages are retrieved and form the first generation. In "selection" phase the extent to which pages are relevant to the considered topic is investigated and each page takes a score. Pages having scores higher than a specific value are stored and the remaining pages are discarded. In "multiplying" phase links of pages in present generation are educed and the link takes score based on the page in which it is available. Then, a predetermined number of links are selected randomly, their pages are retrieved and they form the new generation. An infra-search is performed in mutation in certain time periods. Some keywords (at last 4 keywords) are selected and searched by some common search engines (like Google). Then, a predetermined number of found pages with highest priority are retrieved and added to the new generation of pages. To achieve more pages, previous tasks repeat continuously.

4. DEEP WEB

Hidden (deep) web is a part of World Wide Web content that users don't see them because search engines don't find them. Today, since the network is composed of several pages general, public and commercial search engines can't access all the pages available in this huge resource [30]. Many pages are created or removed every day; hence updating this large amount is almost impossible for these engines. Meanwhile, there are a lot of web pages that are placed in databases. These are called dynamic page or on-the-fly-pages. They form a large part of the network and are available to those users who have username and password. Naturally, these pages are out of reach of search engines and are a part of deep web pages.

Administrators of this kind of web sited (deep web) worry about their invisibility. According to a report, many deep web sites created a simple and superficial script of their pages with high costs [14]. Amazon is an example of these sites. Respecting the importance of this issue, finding a solution to improve present conditions and match search engines with today's expanding web is the most important research topic in the field of search engines. Since the nature of deep web is completely different from that of superficial web, strategies used by search engines to mark these web sites differ from common methods.

5. CHALLENGES AVAILABLE IN THE FIELD OF DEEP WEB SEARCH ENGINES

5.1 Analyzing the Inquiry Intermediate

The first challenge facing all search engines is to analyze and identify web site's inquiry intermediate. Since inquiry intermediate in the only way to access information in deep web sites, thus identification of components of this intermediate is the first step toward accessing information behind it.

The first step to analyze an inquiry intermediate is detection of these pages' nature. Generally, there are three reasons for developing dynamic pages: time sensitive information, consistency with users' needs, and various entries by the user. On the other hand, there are various mechanisms to develop dynamic pages. These techniques are summarized in three following classes:

- Server-side applications.
- Codes devised in server-side running [1].
- Codes devised in client-side running (like Java Applets and ActiveX Controls).

Server-side web applications are those which completely run on the server and only the result of their implementation is sent to the receiver device for display. Client-side applications are those which run on the device of client user and their only difference with a desktop application is that they are limited in the web browser environment. With this definition of a dynamic page, a crawler must search specific component in a web page. After retrieving the whole page, a set of processes are performed to gain the main form and its relevant fields. Modeling inquiry intermediates is a very difficult task since it is really hard to related features and performances of each element, so that when it is done by human mistakes occur in selections. Applying apocalypse principles to reduce field tags is one of the most common used methods [12]. Research [13] has proved the effectiveness of this method. Observations of thousands of inquiry intermediates reveal that all of these pages have shared blocks. According to this observation, a theory is developed that says there is a hidden grammar based on which inquiry intermediates is formed.

5.2 Auto Completion of Forms

Auto completion of forms and creation of proper inquiry words is another studied issue in the field of analyzing inquiry intermediate. After the available formed is analyzed in the intermediate, it must be possible to complete the form automatically and send it to the related site to be able to access information behind the intermediate. In other words, appointing proper values to educed fields in the previous step is responsible for creating inquiry to detect content of the database. There are some commercial services on the Internet that automatically fill in certain fields. Microsoft passport is among these services. "ShopBot" is a proper method to fill in forms automatically [11]. This factor is used for comparative shopping using detection methods in certain domains. But this is not a general guideline for all domains. "Gravano" et al presented a method called "2ps"[24]. In this two-step algorithm method the crawler first educes some words from the intermediate page and sends it to the database as an inquiry. Then, it creates more proper inquiries in retrieving resulted documents and educing new keywords. Another technique used to develop a proper inquiry is "machine learning" method [2]. Here, some documentaries first teach a document classifier. Base on this training, some principles transform to a setoff inquiries. In some cases, inquiry applying is client-side [10]. The present paper proposes a client-side crawler.

5.3 Analyzing Returned Results

After applying the inquiry and gaining the results from studied database, result should be investigated for following reasons:

5.3.1. To gain new keywords

Considering the form auto completion step and used algorithm, form completion phase may be repeated. Keywords are one of the best resources to develop proper inquiries. These keywords are educed from results gained from the database [24, 28].

5.3.2. Approximation of database documents number in a certain field [27].

A great deal of work is done in the field of clustering results of a search engine that can be used as an auxiliary tool to develop new inquiries [16, 20].

5.4 Identification of Site's Subject

So far, many sites tried to classify databases available on the web manually. But since the number of these databases is large (and they are increasing daily), achieving a complete hierarchy of databases is unseemly [2]. But, on the other hand, we hope that this task to result in a precise hierarchy. Respecting this issue, the most is important part of a deep web search engine is responsible. In this part of the engine, the database is placed in a predetermined classification (based on gained results). Of course, in some cases tagging is used to cluster instead of classification. "Change"[3] et al showed that database topics are distributed widely on the web but a limited number of topics cover a significant number of databases.

From one viewpoint, different classification methods of deep web sites can be divided into two general groups [24]:

5.4.1. Inquiry-based classification

There is no need to retrieve a document form the database in this method but the classification is based on the number of results gained from the related database [2, 25, and 26].

5.4.2. Crawling-based classification

Here, classification differs from the previous method. Classification method is not based on results but result documents are retrieved and every document goes to its own class using a classifier. Then, classed relevant to the database are educed considering the density of documents in each class [27].

5.4.3. Comparison

considering the explanations about performance of these two methods of classifying deep web sites it is indicted that inquiry-based classification is prior to crawling-based classification in terms of precision and speed of classification[24].

6. CONCLUSION

The present paper tried to present methods for correct determination of link quality or, in other words, correct anticipation of web page quality (those pages which have not been retrieved so far but there is a link to them). Some points were noted on problems available in local search method used in intensive crawling, problems such as sites having no links to one another or sideway links. Also, guidelines were proposed to dominate these problems the most important of which is to use genetic algorithm. Then, problems of deep web search engines were investigated and using marks and signs of search intermediate of deep web to reduce its topic was studied.

7. REFERENCES

[1] See http://java.sun.com/products/servlet/ 2006 Java Servlet TM Technology

[2] Gravano L., Iperirotis P.G, Sahami M. 2003 QProber: A system for automatic classification Web databases. In Proceedings of the ACM Trans. Information System pp.1-14

[3] Change K. C.C., He B., Li C., Patel M., Zhang Z. 2004 Structured databases on the web: Observations and implications. SIGMOD Record

[4] Chakrabarti S., Berg M.V.D., Dom B. 1999 Focused Crawling: a New Approach to Topic-Specific Web Resource Discovery. In 31th Computer Networks Conference, pp. 1623-1640

[5] Chakrabarti S., Berg M.V.D., Dom B. 1997 Distributed Hypertext Resource Discovery through Example". In 25th International Conference on Very Large Data Base, USA

[6] Cho J., Garcia-Molina H. 2000 the Evolution of the Web and Implications for an Incremental Crawler. In 26th International Conference on Very Large Data Bases, USA, pp. 200-209

[7] Cho J., Garcia-Molina H. 2000 Synchronizing a Database to Improve Freshness. In ACM SIGMOD International Conference on Management of Data, USA, pp.117-128

[8] Cho J., Garcia-Molina H. and Page L. 1998 Efficient Crawling through URL Ordering In 7th In World Wide Web Conference, Australia. pp. 161-172

[9] Diligenti M., Coetzee F., Lawrence S. 2000 Focused Crawling Using Context Graphs. In 26th International Conference on Very Large Databases (VLDB), Cairo, Egypt, pp. 527-534

[10] Alvarez M., Pan A., Raposo J. and Vina A. 2006 Crawling the client-side hidden web

[11] Doorenbos R. B., Etzioni O., Weld D. S. 1997 A scalable comparison-shopping agent for the World-Wide Web. In First International Conference on Autonomouse Agent, pp. 39-48

[12] Lage J.P., da Silva A., Golgher P.B., Laender A.H. 2004 Automatic generation of agent for collecting hidden web pages for data extraction. Data Knowledge Eng. pp. 177-196

[13] Zhang Z., He B., Chang K. 2004 Understanding Web query interfaces: best- effort parsing with hidden syntax. In Proceeding of the 2004 ACM SIGMOD international Conference on Management of Data, Paris, France

[14] Article on New York Times 2006 Old Search Engine, the Library Tries to Fit Into a Google World. See http://www.nytimes.com/2004/06/21/technology/21LIBR.html

[15] Najork M., Wiener J. 2011 Breadth-First Search Crawling Yields High-Quality Pages. In 10th Conference on Word Wide Web, Hong-Kong. pp. 114- 118

[16] Broder A., Carnel D. 2005 Sampling search-engine results. In 14th international Conference on world Wide Web, Chiba, Japan

[17] Qin J., Chen H. 2005 Using Genetic Algorithm in Building Domain-Specific Collections: An Experiment in the Nanotechnology Domain. In 38th Annual Hawaii International Conference on System Sciences, USA

[18] Rennie J., McCallum A. 1999 Using Reinforcement Learning to Spider the Web Efficiently. In 16th International Conference on Machine Learning, USA, pp. 335-343

[19] Rungsawang A., Angkawattanawit N. 2005 Learnable Topic-Specific WebCrawler. Journal of Network and Computer Applications, UK, pp. 97-114

[20] Koster M. 1993 Guidelines for robot writers, http://www.robotstxt.org/guidelines.html,

[21] Shkapenyuk V., Suel T. 2001 Design and Implementation of a High-Performance Distributed Web Crawler. In 18th International Conference on Data Engineering, USA, pp. 357- 368

[22] Younes H., Chabane D. 2004 High Performance Crawling System. In 6th ACM SIGMM International Workshop on Multimedia Information Retrieval, New York, USA, pp. 299-306

[23] Gulli A., Signorini A. 2005 The Index able Web is More than 11.5 billion pages. In 14th International World Wide Web Conference, Chiba, Japan

[24] Gravano L., Ipeirotis P.G., Sahami M. 2002 Query- vs. Crawling-based Classification of Searchable Web Databases. IEEE Data Engineering Bulletin

[25] Gravano L., Garcia-Molina H., Tomasic A. 1999 GIOSS: Text source discovery over the Internet. ACM TODS

[26] Ipeirotis P.G., Gravano L., Sahami M. 2001 Probe, count, and classify: categorizing hidden web databases. In Proceeding of 2001 ACM SIGMOD, international Conference on Management of Data, Santa Barbara, California, U.S.

[27] Ipeirotis P. G., Gravano L. 2002 Distributed Search over the Hidden web: Hierarchical Database Sampling and Selection. In 28th VLDB Conference, Hong Kong, China

[28] Barbosa L., Freire J. 2004 Siphoning Hidden-Web Data through Keyword-Base Interfaces. In SBBD

[29] Castillo C. 2004 Effective Web Crawling. In ACM SIGIR. Vo.39, Issue 1

[30] Kumar Sharma D. 2011 A Novel Architecture for Deep Web Crawler. International Journal of Information Technology and Web Engineering