PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTERY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

جامعة محمد بوضياف - المسيلة
Université Mohamed Boudiaf - M'sila

UNIVERSITY OF M'SILA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Computer Science Department

Dissertation submitted in partial fulfilment of the requirements for the Degree of MASTER

Domain : Mathematics and Computer Science
Branch : Computer Science
Specialty : Network

By: MAHAMEDI Soundous

Title

Implementation of a Hidden Web crawler

Publicly defended: / /2015 before a Jury composed of:

Mrs.SAOUDI Lalia	**University of M'sila**	**Supervisor**
.....	**University of M'sila**	**Chair**
.....	**University of M'sila**	**Examiner**
.....	**University of M'sila**	**Examiner**

Academic Year : 2014 /2015

CONTENT TABLE

1 Context of the study

Since the early days of the Web, search has been the ubiquitous discovery tool for Web users to find the answers to their questions and solutions to their needs. And since the wide adoption of search, organizations have vied for position in search results using a variety of strategies and tactics to capitalize on this rich source of Web traffic.

The World Wide Web is a global information medium of interlinked hypertext documents accessed via computers connected to the internet. Most of the users rely on traditional search engines to search the information on the Web. These search engines deal with the Surface Web which is a set of Web pages directly accessible through hyperlinks and ignores a large part of the Web called hidden Web which is hidden to present-day search engines. It lies behind search forms and this part of the Web containing an almost endless amount of sources providing high quality information stored in specialized databases, can be found in the depths of the WWW. World Wide Web (WWW) is broadly divided into two categories:

- The surface Web contains 1% of information content of the Web. Search engine crawl along the Web to extract and index text from HTML documents on the Websites, then makes this information searchable through keywords.
- The hidden Web contains 99% of information content of the Web. Most of this information is contained in the databases and is not indexed by search engines.

This means if we are searching for information from surface Web only, we search through only 1% of WWW and miss 99% of it whereas 95% of hidden Web is free publicly accessible information. As the hidden Web information that is hidden behind the search query forms can only be accessed by interacting with these forms, development of automated system that interacts with the search forms and extracts the hidden Web content would be of great value to human users.**[23]**

2 Statement of the Problem

In fact all the information is available on the internet but buried behind search interfaces and stored inside the databases. Therefore, the required Web service has to dig, gather, normalize and store the information to make it searchable for the users.

Our Hidden Web crawler must implement some techniques which respond to the following challenges:

a. **Automatic Finding of Search Interfaces of a specific domain**: Automatic detection of search interfaces will also be useful for ranking of search engines' results.

b. **Querying Web Databases:** Retrieving information by filling out Web search forms is a typical activity for a Web user. The automation of querying and retrieving data behind search interfaces is desirable and essential.
c. **Extract and analyze response pages:** A crawler must extract the content lying behind the form interfaces of the selected content sources.
d. **Indexing response web pages:** permitting rapid identification of which crawled pages contains particular words or phrases, used to optimize speed and performance in finding relevant documents for a search query.

3 Motivation

Search engines are clearly among those that can benefit from the information behind search interfaces. Current searchers deal well with informational (give me the information I search for) and navigational (give me the URL of the site I want to reach) queries, but transactional (show me sites where I can perform a certain transaction, e.g., shop, access database, download a file, etc.) queries are satisfied only indirectly. It is needless to say that data in the deep Web is indispensable to use when answering transactional queries that constitute about 10% of web search engine queries.

The fact that the deep Web is still poorly indexed is another major issue for search engines, which are eager to improve their coverage of the Web.

4 Objective

4.1 General Objectives

In this work, we study how we can build a *Hidden-Web crawler* that can automatically download pages from the hidden Web, so that search engines can index them. We address the problem of building a hidden Web crawler; one that can crawl and extract content from these hidden databases. Such a crawler will enable indexing, analysis, and mining of hidden Web content, askin to what is currently being achieved with the PIW (Public Indexable Web).

4.2 Specific Objectives

The specific objectives of this research are the following:

a- *Form Analysis:* Parse and process the form relevant searchable forms to build a copy of them.

b- *Value assignment and ranking:* Use approximate string matching between the form labels and the labels in the predefined list to generate a set of candidate value, then use the adaptive approach to generate a new candidate values from the generated page.

c- *Response Analysis and Navigation:* Analyze the response pages (i.e., the pages received in response to form submissions) to check if the submission yielded valid search results. Use this feedback to tune the value assignments in Step b. Crawl the hypertext links in the response page to find new URLs.

d- Index the content behind many HTML forms.

5 Research Approach and Methodology

The proposed hidden web crawler's goal is to automate the process of searching, viewing, filling in and submitting the search forms and analyzing the response pages. It employs a suite of algorithms distributed into following three phases:

5.1 Domain Definitions

The process for creating the domain definitions was the following: we choose the book domain we automatically explored 7 sites at random via SQLI queries to extract their data to be inserted into database initial values and to define the attributes and their aliases. The score of each book, the specificity weight of each attribute and the relevance threshold were also manually chosen from our experience visiting these sites.

5.2 Processing Forms

The first stage of the extraction of the hidden Web information is the detection of deep Web search interface.

- The system tries to classify the found forms into two groups searchable and non searchable forms.
- the system tries to determines if the searchable form is relevant with respect to the domain by matching its attributes with the fields of the searchable forms, The method we use to determine if a form is relevant to a domain consists of adding the frequency of each label, pondered by its *weight*, and checking if the sum exceeds the *relevance threshold* μ.
- If the form is relevant, the crawler uses it to execute the queries defined in the database.

5.3 Response page analysis:

This phase analyzes response pages (the server response to the crawler Query), wherein the response analyzer distinguishes between the response pages containing search results, and pages containing error messages

6 Thesis Outline

This thesis consists of four chapters:

The first chapter considers the content of our work. Firstly we provide an overview of Web crawler and its different operations, its architecture and the different technical features which should and must provide in the Web crawler.

The second chapter provides a general overview of the hidden Web and their features; we explain the causes to have hidden Web pages, the techniques of hidden Web crawler and how to access in it.

The third chapter presents the conceptual aspect of our crawler prototype, and its different modules to implement the hidden Web crawler.

We present our prototype in the last chapter and discuss the results obtained from running the prototype, revealing the improvements resulting from the proposed method.

Finally, we conclude this project by a general conclusion, recommendations and different perspectives.

GENERAL CONCLUSION

Current-day crawlers are used to build repositories of Web pages that provide the input for systems that index, mine, and otherwise analyze pages (e.g., a Web search engine). However, these crawlers are restricted to the set of pages in the publicly indexable portion of the Web.

In this project, we addressed the problem of extending current-day crawlers to build repositories that include pages from the "hidden Web", the portion of the Web behind searchable HTML forms.

We proposed an application/task specific approach to hidden Web crawling. We argued that as with the PIW, the tremendous size and heterogeneity of the hidden Web makes comprehensive coverage very difficult, and possibly less useful, than task-specific crawling. A narrow application focus is also useful in designing a crawler that can benefit from knowledge of the particular application domain.

We presented a simple operational model of a hidden Web crawler that succinctly describes the steps that a crawler must take: relevant page extraction, form detection, form structure classification, form content classification, form submission and response page analysis.

We described the architecture and design techniques used in IHiWC, a prototype crawler implementation based on SQLI to get the initial keywords values and fill the repository database in the training phase.

In the test phase we choose a set of 12 sites and calculate different metrics to evaluate the performance of our crawler, the promising experimental results using IHiWC demonstrate the feasibility of hidden Web crawling and the effectiveness of our different techniques to implement this crawler.

We believe that our operational model sets the stage for designing a variety of hidden Web crawlers for different domain, ranging in complexity from the simple label matching approach of IHiWC, to the use of sophisticated natural language and knowledge representation techniques.

TERMINOLOGY

Adaptive website: An adaptive website is a website that builds a model of user activity and modifies the information and/or presentation of information to the user in order to better address the user's needs.

Backlinks: also known as *incoming links*, *inbound links*, *inlinks*, and *inward* links, are incoming links to a website or web page. In basic link terminology, a backlink is any link received by a web node (web page, directory, website, or top level domain) from another web node.

CAPTCHA: is a type of challenge-response test used in computing to determine whether or not the user is human.

Checksum: is a small-size datum from a block of digital data for the purpose of detecting errors which may have been introduced during its transmission or storage.

Continuous Crawl: in continuous crawl mode, the search appliance is crawling your enterprise content at all times, ensuring that newly added or updated content is added to the index as quickly as possible. After the Google Search Appliance is installed, it defaults to continuous crawl mode and establishes the default collection.

Crawlytics: Social Crawlytics is a tool that crawls a website to see how many times pages from that website have been shared on social media pages, hence the name, Social Crawlytics.

CrawlTrack: is a free and open source audience statistics of a website.

Data records: in computer science, a **record** is a basic, it is a collection of *elements*, typically in fixed number and sequence and typically indexed by serial numbers or identity numbers. The elements of records may also be called *fields* or *members*.

Darknet: is a private network where connections are made only between trusted peers sometimes called "friends" (F2F) using non-standard protocols and ports.

Free-form text: words and sentences, such as input to a word processor or e-mail program. Since text is already free form, the term is redundant; however, it is used to emphasize its unstructured nature. See free-form database, unstructured data and text mining.

Fingerprint: In computer science, a fingerprinting algorithm is a procedure that maps an arbitrarily large data item (such as a computer file) to a much shorter bit string, its fingerprint, that uniquely identifies the original data for all practical purposes just as human fingerprints uniquely identify people for practical purposes. This fingerprint may be used for data deduplication purposes.

Giga Alert: formerly known as Google Alert, is the web's leading automated search and web intelligence solution for monitoring your professional interests online. It tracks the entire web for your personalized topics and sends you a new results by daily email.

I2P: is an anonymous overlay network - a network within a network. It is intended to protect communication from dragnet surveillance and monitoring by third parties such as ISPs.

MIME: An internet media type is a standard identifier used on the internet to indicate the type of data that a file contains. Common uses include the following:

- email clients use them to identify attachment files,
- web browsers use them to determine how to display or output files that are not in HTML format,
- search engines use them to classify data files on the web.

PageRank: is an algorithm used by Google Search to rank websites in their search engine results. It is a way of measuring the importance of website pages. According to Google: PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is.

Robots exclusion standard: also known as the *robots exclusion protocol* or *robots.txt protocol*, is a standard used by websites to communicate with web crawlers and other web robots. The standard specifies the instruction format to be used to inform the robot about which areas of the website should not be processed or scanned. Robots are often used by search engines to categorize and archive web sites, or by webmasters to proofread source code.

Search engine spamming: spamdexing (also known as search engine spam, search engine poisoning, Black-Hat SEO, search spam or web spam) is the deliberate manipulation of search engine indexes. It involves a number of methods, such as repeating unrelated phrases, to manipulate the relevance or prominence of resources indexed in a manner inconsistent with the purpose of the indexing system.

Page Selection: Dynamic page selection is enabled when the web content viewer is configured to use the Dynamically select a web content page link broadcasting option. To render a content item, the dynamically selected target page must contain at least one web content viewer. The viewer must be configured to receive links from other portlets.

Snapshot: is the state of a system at a particular point in time.

Spambot: is an automated computer program designed to assist in the sending of spam. Spambots usually create fake accounts and send spam using them, although in many cases it would be obvious that a spambot is sending it. This has led to the development of password-cracking spambots that are able to send spam using other people's accounts.

Spider trap: or (*crawler trap*) is a set of web pages that may intentionally or unintentionally be used to cause a web crawler or search bot to make an infinite number of requests or cause a poorly constructed crawler to crash.

Tor: is short for The Onion Router (thus the logo) and was initially a worldwide network of servers developed with the U.S. Navy that enabled people to browse the internet anonymously.

User-agent: In computing, a user agent is software (a software agent) that is acting on behalf of a user. For example, an email reader is a mail user agent, and in the Session Initiation Protocol (SIP), the term *user agent* refers to both end points of a communications session.

Vertical search engine: as distinct from a general web search engine, focuses on a specific segment of online content. They are also called specialty or topical search engines. The vertical content area may be based on topicality, media type, or genre of content. Common verticals include shopping, the automotive industry, legal information, medical information, scholarly literature, and travel.

Web search engine: is a software system that is designed to search for information on the World Wide Web.

Web archiving: is the process of collecting portions of the World Wide Web to ensure the information is preserved in an archive for future researchers, historians, and the public.

Web data mining: is a technique used to crawl through various web resources to collect required information, which enables an individual or a company to promote business, understanding marketing dynamics, new promotions floating on the internet, etc. There is a growing trend among companies, organizations and individuals alike to gather information through web data mining to utilize that information in their best interest.

BIBLIOGRAPHY

[1] Christopher Olston and Marc Najork, Web Crawling, Foundations and Trends in Information Retrieval, Vol. 4, No. 3, 2010.

[2] Monica Peshave, how search engines work and a web crawler application, Department of Computer Science University of Illinois at Springfield, 2005.

[3] Carlos Castillo, Effective Web Crawling Submitted to the University of Chile in fulfillment of the thesis requirement to obtain the degree of Ph.D. in Computer Science, November 2004.

[4] Junghoo Cho, crawling the web: discovery and maintenance of large-scale web data, a dissertation submitted to the department of computer science and the committee on graduate studies of stanford university in partial fulfillment of the requirements for the degree of doctor of philosophy, November 2001.

[5] Marcnajork, Web Crawler Architecture, Microsoft Research, Mountain View, CA, USA.

[6] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, An Introduction To Information Retrieval, Draft of April 1, 2009.

[7] Vladislav Shkapenyuk, T orsten Suel, Design and Implementation of a High Performance Distributed Web Crawler, Department Polytechnic University Brooklyn, NY 1120, 2002.

[8] Steve Lawrence and C. Lee Giles. Accessibility of information on the web. Nature, 400(6740):107–109, July 1999.

[9] Krishna Bharat and Andrei Broder. Mirror, mirror on the web: A study of host pairs with replicated content. In Proceedings of the Eighth International World-Wide Web Conference, Toronto, Canada, May 1999.

[10] Junghoo Cho and Hector Garcia-Molina. The evolution of the web and implications for an incremental crawler. In Proceedings of the Twenty-sixth International Conference on Very Large Databases, Cairo, Egypt, September 2000.

[11] Craig E. Wills and Mikhail Mikhailov. Towards a better understanding of web resources and server responses for improved caching. In Proceedings of the Eighth International World-Wide Web Conference, Toronto, Canada, May 1999.

[12] Martijn Koster. Robots in the web: threat or treat? ConneXions, 4(4), April 1995.

[13] Robots exclusion protocol. http://info.webcrawler.com/mak/projects/robots/exclusion.html.

[14] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In Proceedings of the Seventh International World-Wide Web Conference, Brisbane, Australia, April 1998.

[15] Sonali Gupta, Komal Kumar Bhatia, A Comparative Study of Hidden Web Crawlers, International Journal of Computer Trends and Technology (IJCTT) – volume 12 number 3 – Jun 2014.

[16] Internet Archive, http://archive.org/.

[17] Panagiotis G. Ipeirotis, Classifying and Searching Hidden-Web Text Databases, Graduate School of Arts and Sciences, Columbia University, 2004.

[18] 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.

[19] Sachin Chirgaiya & Jayendra Barua, Multi-Threaded Web Crawler based on Multi keyword Web Crawling using Ontology, Department of Computer Science And Engineering Malwa Institute of Technology, Indore, India, Volume-2, Issue-4, 2013.

[20] Deep web white paper. July 2000.

[21] Dhiraj Khurana, Satish Kumar, Web Crawler Web Crawler: A Review, Department, University Institute of Engineering & Technology, Maharshi Dayanand University, Rohtak(Haryana), International Journal of Computer Science & Management Studies, Vol. 12, Issue 01, January 2012.

[22] K.F. Bharati, Prof. P. Premchand and Prof. A Govardhan, HIGWGET-A Model for Crawling Secure Hidden WebPages, International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.3, No.2, March 2013.

[23] Sriram Raghavan Hector Garcia-Molina, Crawling the Hidden Web, Computer Science Department, Stanford University, Proceedings of the 27th VLDB Conference, Roma, Italy, 2001.

[24] Moumie Soulemane, Mohammad Rafiuzzaman, Hasan Mahmud, Crawling the Hidden Web: An Approach to Dynamic Web Indexing, International Journal of Computer Applications (0975 – 8887), Volume 55– No.1, October 2012.

[25] Xu Jia, Design, Implementation and Evaluation of an Automated Testing Tool for Cross-Site Scripting Vulnerabilities, Darmstadt University of Technology (TUD), Computer Science Department, Darmstadt, in July 2006.

[26] Xin Wang, Luhua Wang, Gengyu Wei, Dongmei Zhang, Yixian Yang, hidden web crawling for sql injection detection, Proceedings of IC-BNMT 2010.

[27] William G.J. Halfond, Jeremy Viegas, and Alessandro Orso College of Computing Georgia Institute of Technology, A Classification of SQL Injection Attacks and Countermeasures, 2006 .

[28] Alexandros Ntoulas, Crawling and Searching the Hidden Web, university of california Los Angeles, A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy in Computer Science, 2006.

[29] Crawling the Hidden Web, Sriram Raghavan, Hector Garcia-Molina Computer Science Department, Stanford University Stanford, CA 94305, USA, 2001.

[30] Michael k. Bergman, the deep web: surfacing hidden value, white paper, september 24, 2001.

[31] Harish Saini , Kirti Nagpal, Algorithm for Merging Search Interfaces over Hidden Web, International Journal Of Computational Engineering Research (ijceronline.com) Vol. 2 Issue. 4. Augest 2012.

Links and websites

[32] http://www.cellopoint.com/media_resources/blogs/2011/03/Web_Crawlers

[33] http://www.crawltrack.net/

[34] http://en.wikipedia.org/wiki/Web_crawler

[35] http://fr.slideshare.net/poonamkenkre/web-crawler-14590800

[36] http://en.wikipedia.org/wiki/Deep_Web

[37] http://en.wikipedia.org/wiki/Adaptive_website

[38] http://ugweb.cs.ualberta.ca/~c391/manual/chapt1.html

[39] http://www.vogella.com/tutorials/JavaIntroduction/article.html

[40] http://jsoup.org/

[41] http://www.tic.udc.es/~mad/publications/cchiddenbwf_extended.pdf

ملخص

إنّ روبوتات الفهرسة في الوقت الحاضر تستطيع فقط استرجاع محتوى صفحات الويب المفهرسة عامة، علما أنّ مجموعة صفحات الويب المتاحة فقط عن طريق اتباع روابطها، مهملين استمارات البحث والصفحات التي تحتاج إلى تصريح أو تسجيل أولي. ويهملون خاصة الكمية الكبيرة المخفية وراء استمارات البحث الموجودة في قاعدة البيانات الالكترونية الكبيرة الخاصة بالبحث. في هذا البحث اقترحنا إطارا لتنبؤ مشكل استخراج المحتوى المخفي. و من أجل ذلك أنشأنا روبوت معين في مجال خاص و هو: IHWC و وصفنا بنية الـ IHWC ثم قدمنا عددا من للتقنيات الجديدة التي استعملت في تصميمه، في مقاربته و في تطبيقه، كما قدمنا أيضا نتائج التجارب المستعملة بهدف اختبار و تأكيد تقنياتنا.

كلمات مفاتيح: روبوت عميق، روبوت الفهرسة المخفية، تصنيف الإستمارات، تسليم الإستمارات.

Abstract

Current-day crawlers retrieve content only from the publicly indexable Web, i.e., the set of web pages reachable purely by following hypertext links, ignoring search forms and pages that require authorization or prior registration. In particular, they ignore the tremendous amount of high quality content "hidden" behind search forms, in large searchable electronic databases. In this work, we provide a framework for addressing the problem of extracting content from this hidden Web, that is why we have built a task-specific hidden Web crawler called the Intelligent Hidden Web Crawler (IHiWC). We describe the architecture of IHiWC and present a number of new techniques that went into its design, approach and implementation. We also present results from experiments we conducted to test and validate our techniques.

Keywords: Deep crawler, Hidden Web Crawling, forms classification, forms submission

Résumé

Les robots d'indexation de nos jours peuvent seulement récupérer le contenu des pages Web publiquement indexables, à savoir, l'ensemble des pages Web accessibles uniquement en suivant les liens hypertextes, ignorant les formulaires de recherche et les pages qui nécessitent une autorisation ou un enregistrement préalable. En particulier, ils ignorent l'énorme contenu de haute qualité "caché" derrière des formulaires de recherche, dans les grandes bases de données électroniques. Dans ce travail, nous proposons un cadre pour prédire le problème de l'extraction du contenu de ce Web caché. Pour cette raison, nous avons construit un robot spécifique dans un domaine particulier : Intelligent Hidden Web Crawler. Nous décrivons l'architecture de IHiWC et présentons un certain nombre de nouvelles techniques qui ont servi à sa conception, à son approche et à son implémentation. Nous avons également présenté les résultats des essais utilisées afin de tester et valider nos techniques.

Mots-clés: robot profond, robot d'indexation caché, classement des formulaires, soumission des formulaires.