NOVATEUR PUBLICATIONS
INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY [IJIERT]
ISSN: 2394-3696
VOLUME 3, ISSUE4, APR.-2016

SMART CRAWLER: A TWO-STAGE CRAWLER FOR EFFICIENTLY HARVESTING DEEP-WEB INTERFACES

Ms. Asmita D Rathod,
Department of Computer Engineering,
SRTM University, Hingoli, India.

ABSTRACT

Deep web growing at a very fast pace, lot of speculations in techniques this techniques has been added that help efficiently locate deep-web interfaces. However, due to the large volume of web resources and the dynamic nature of deep web, achieving wide coverage and high efficiency is a challenging issue. In this paper author has proposed a two-stage framework, namely Smart Crawler, for efficient harvesting deep web interfaces. Smart Crawler performs site-based searching for center pages by using search engines, avoiding visiting a large number of pages. To achieve more accurate results for a focused crawl, Smart Crawler techniques prioritize websites to highly relevant ones for a given topic. Smart Crawler achieves fast in-site searching by finding most relevant links with an adaptive link-ranking. To eliminate bias on visiting some relevant links in hidden web directories, author has designed a link tree data structure to achieve wider coverage for a website.

KEYWORDS: Deep web, two-stage crawler, feature selection, ranking, adaptive learning.

INTRODUCTION

The hidden web refers to the contents lie behind searchable web interfaces that cannot be indexed by searching engines. It is estimated that the deep web contains approximately 91,850 terabytes and the surface web is only about 167 terabytes in 2003 [1]. In recent years studies estimated that 1.9 zettabytes were reached and 0.3 zettabytes were consumed worldwide in 2007 [2], [3]. A significant portion of this huge amount of data is estimated to be stored as structured or Relational data in web databases — deep web makes up about 96% of all the content on the Internet, which is 500 to 550 times greater than the surface web [5], [6]. It is an challenge to locate the deep web databases, because they are not registered with any search engines, are usually sparsely distributed, and constantly changing. To address this problem, previous work has proposed two types of crawlers, generic crawlers and focused crawlers. Generic crawlers [10], [11], [12], [13], [14] fetch all searchable forms and cannot focus on a specific topic.

Focused crawlers such as Form-Focused Crawler (FFC) [15] and Adaptive Crawler for Hidden-web Entries (ACHE) [16] can automatically search online databases on a specific topic. Crawler must produce a large quantity of high-quality results from the most relevant content sources [15], [16], [18], [19], [20], [21]. For assessing quality source, Source ranks the results from the selected sources by computing the agreement between them [20], [21]. When selecting a relevant subset from the available content sources, the set of retrieved forms is very heterogeneous. Thus it is crucial to develop smart crawling strategies that are able to quickly discover relevant content sources from the deep web as much as possible. We propose an effective deep web harvesting framework, namely Smart Crawler, for achieving both wide coverage and high efficiency for a focused crawler. Our crawler is divided into two stages: site locating and in-site exploring.

RELATED WORK

Previous work has proposed a number of techniques and tools, including deep web understanding and integration [10], [24], [25], [26], [27], hidden web crawlers [18], [28], [29], and deep web samplers [30], [31], [32]. For all these approaches, the ability to crawl deep web is a key challenge. Crawling deep web has three steps: locating deep web content sources, selecting relevant sources and extracting underlying content [19].

LOCATING DEEP WEB CONTENT SOURCES

Generic crawlers are mainly developed for characterizing deep web and directory construction
of deep web resources that do not limit search on a specific topic, but attempt to fetch all searchable Forms [10], [11], [12], [13], [14]. The Database Crawler in the MetaQuerier [10] is designed for automatically discovering query interfaces. Database Crawler first finds root pages by an IP-based sampling, and then performs shallow crawling to crawl pages within a web server starting from a given root page. The IP based sampling ignores the fact that one IP address may have several virtual hosts [11], thus missing many websites. To overcome the drawback of IP based sampling in the Database Crawler, Denis et al. propose a stratified random sampling of hosts to characterize national deep web [13], using the Host graph provided by the Russian search engine Yandex. I-Crawler [14] combines prequery and post-query approaches for classification of searchable forms.

SELECTING RELEVANT SOURCES

Available hidden web directories [34], [8], [7] tends to have low coverage for relevant online databases [23], which restricts their ability in satisfying data access needs [35]. Focused crawler is developed to visit links to pages of interest. The classifier learns to classify pages as search-relevant or not and gives priority to links in search relevant pages. The baseline classifier gives its choice as feedback so that the apprentice can learn the features of relevant links and prioritize links in the frontier. The FFC [15] and ACHE [16] are focused crawlers used for searching interested deep web interfaces. The FFC comprises of three classifiers, a page classifier that scores the relevance of retrieved pages with a specific topic, a link classifier that prioritizes the links that may lead to pages with searchable forms, and a form classifier that filters out non-searchable forms. ACHE improves FFC with an adaptive link learner and automatic feature selection. Source Rank [20], [21] assesses the relevance of deep web sources during retrieval. Based on an agreement graph, Source Rank calculates the stationary visit probability of a random walk to rank results. Smart Crawler is a domain-specific crawler for locating relevant deep web content sources. Instead of simply classifying links as relevant or not, Smart Crawler first ranks sites and then prioritizes links within a site with another ranker.

SYSTEM ARCHITECTURE

PROPOSED SYSTEM:

Author has proposed a two-stage framework, namely Smart Crawler, for efficient harvesting deep web interfaces and second time smart crawler is used as a fast in site searching. Mainly in the first stage, Smart Crawler performs site-based searching for main pages with the help of search engines. In the second stage, Smart Crawler achieves fast in-site searching by digging most relevant links with an adaptive link-ranking. To eliminate bias on visiting some highly relevant links in hidden web directories. Author has designed, a link tree data structure to achieve wider coverage for a website.

TWO STAGE ARCHITECTURE OF SMART CRAWLER:

For effectively discovering deep web data sources, Smart Crawler is designed with a two stage architecture, *site* locating and in-site exploring, as shown in Fig. No. 1. The first site locating stage finds the most relevant site for a given topic, and then in next phase second in-site explores searchable forms from the site. Seeds sites plays an important role of finding candidate sites can be given for Smart Crawler to start crawling, which begins by different URLs from chosen seed sites to explore other pages and domains. Smart crawler comes with a capability of “reverse searching” when the number of unvisited URLs in the database is less than a threshold during the crawling process. Site Frontier is designed to fetch homepage of different URLs from the site database, which are ranked and prioritize by Site Ranker on basis of relevant sites. The Site Ranker comes with a ability of an Adaptive Site Learner, which adaptively learns from features of deep-web sites. To achieve more accurate results for a focused crawl, Site Classifier categorizes URLs into relevant or irrelevant for a given topic based on the homepage content.

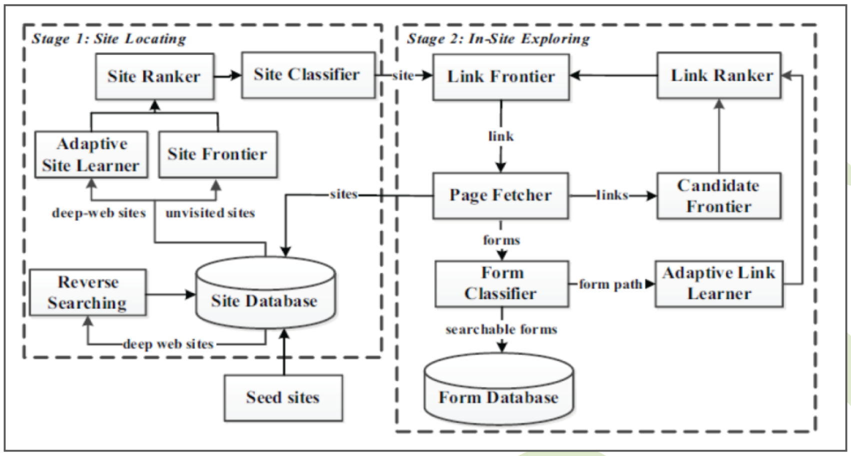

Fig. No.1 Two stage architecture of smart crawler

After finishing the work of a first stage i.e. relevant site searching, the second stage does the work of exploring and excavating searchable forms. In this case links of a most relevant sites are stored in link frontier and it's been used to fetch the corresponding pages. In addition to this links present in the link pages are been fed to candidate frontier, for prioritize links in candidate frontier and then smart crawler ranks them with the help of link ranker. Important point to notice here is site locating stage and in-site exploring stage are mutually intertwined. The Link Ranker is adaptively improved by an Adaptive Link Learner, which learns from the URL path leading to relevant forms. In-site exploring adopts two crawling strategies for high efficiency and coverage. Links within a site are prioritized with Link Ranker and Form Classifier classifies searchable forms.

FEATURE SELECTION & RANKING

Smart Crawler encounters a variety of web pages during a crawling process and the key to efficiently crawling and wide coverage is ranking different sites and prioritizing links within a site.

ADAPTIVE LEARNING

Smart crawler uses an adaptive learning strategy that enhances the learning capacity during crawling. As shown in fig. no.1 site ranker and link ranker are given by adaptive learning. Given Fig. No.2. shows the process for adaptive learning which is invoked periodically.

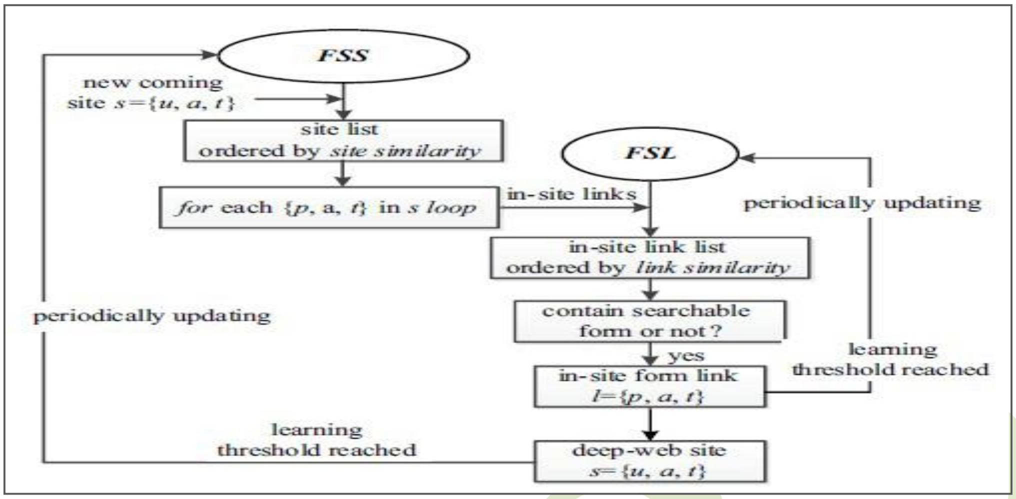

Fig. No.2. Adaptive learning process in smart crawler

RANKING MECHANISM:

Smart Crawler uses a site URLs for prioritizing potential deep sites for a relevant topic, while doing ranking two main aspects are taken into consideration are site similarity and site visit frequency. Site similarity assesses the topic similarity between a new site and known deep web sites. Site frequency assessment parameter depends on the frequency site to appear in other sites, which are available again assesses the popularity and authority of the site.
For prioritizing links of a site, the link similarity is computed based on a similarly to the site similarity described above.

CONCLUSION & FUTURE WORK

In this paper author had proposed, an effective harvesting framework for deep-web interfaces, namely Smart- Crawler. Author had proven that approach achieves both wide coverage for deep web interfaces and maintains highly efficient crawling. Experimental results on a representative set of domains shows the effectiveness of the proposed two-stage crawler, which achieves higher harvest rates than other crawlers. In future, we plan to combine pre-query and post-query approaches for classifying deep-web forms to improve the accuracy of the form classifier.

ACKNOWLEDGEMENTS

We take this opportunity to thank our project guide Prof.P.N.Mundhare and Head of the Department Prof.C.S. Nimodia from ZCOER SPPU Pune for their valuable guidance and for providing all the necessary facilities, & their stimulating suggestions in this paper.

REFERENCES

[1] Peter Lyman and Hal R. Varian. *How much information? 2003. Technical report, UC Berkeley, 2003.*

[2] Roger E. Bohn and James E. Short. *How much information? 2009 report on American consumers. Technical report, University of California, San Diego, 2009.*

[3] Martin Hilbert. *How much information is there in the "information society"? Significance, 9(4):8–12, 2012.*

[4] Idc worldwide predictions 2014: *Battles for dominance – and survival – on the 3rd platform. http://www.idc.com/ research/Predictions14/index.jsp, 2014.*

[5] Michael K. Bergman. *White paper: The deep web: Surfacing hidden value. Journal of electronic publishing, 7(1), 2001.*

[6] Yeye He, Dong Xin, Venkatesh Ganti, Sriram Rajaraman, and Nirav Shah. *Crawling deep web entity pages. In Proceedings of the sixth ACM international conference on Web search and data mining, pages 355–364. ACM, 2013.*

[7] *Infomine. UC Riverside library. http://lib-www.ucr.edu/, 2014.*

[8] *Clusty's searchable database dirctory. http://www.clusty. com/, 2009.*

[9] Booksinprint. *Books in print and global books in print access. http://booksinprint.com/, 2015.*

[10] Kevin Chen-Chuan Chang, Bin He, and Zhen Zhang. Toward large scale integration: *Building a metaquerier over databases on the web. In CIDR, pages 44–55, 2005.*

[11] Denis Shestakov. Databases on the web: *national web domain survey. In Proceedings of the 15th Symposium on International Database Engineering & Applications, pages 179–184. ACM, 2011.*

[12] Denis Shestakov and Tapio Salakoski. Host-ip *clustering technique for deep web characterization. In Proceedings of the 12th International Asia-Pacific Web Conference (APWEB), pages 378–380. IEEE, 2010.*

[13] Denis Shestakov and Tapio Salakoski. *On estimating the scale of national deep web. In Database and Expert Systems Applications, pages 780–789. Springer, 2007.*

[14] Shestakov Denis. *On building a search interface discovery system. In Proceedings of the 2nd international conference on Resource discovery, pages 81–93, Lyon France, 2010. Springer.*

[15] Luciano Barbosa and Juliana Freire. *Searching for hidden-web databases. In WebDB, pages 1– 6, 2005.*

[16] Luciano Barbosa and Juliana Freire. *An adaptive crawler for locating hidden-web entry points. In Proceedings of the 16th international conference on World Wide Web, pages 441–450. ACM, 2007.*

[17] Soumen Chakrabarti, Martin Van den Berg, and Byron Dom. *Focused crawling: a new approach to topic-specific web resource discovery. Computer Networks, 31(11):1623–1640, 1999.*

[18] Jayant Madhavan, David Ko, Łucja Kot, Vignesh Ganapathy, Alex Rasmussen, and Alon Halevy. *Google's deep web crawl. Proceedings of the VLDB Endowment, 1(2):1241–1252, 2008.*

[19] Olston Christopher and Najork Marc. *Web crawling. Foundations and Trends in Information
Retrieval, 4(3):175–246, 2010.*

[20] Balakrishnan Raju and Kambhampati Subbarao. Sourcerank: *Relevance and trust assessment for deep web sources based on inter-source agreement. In Proceedings of the 20th international conference on World Wide Web, pages 227–236, 2011.*

[21] Balakrishnan Raju, Kambhampati Subbarao, and Jha Manishkumar. *Assessing relevance and trust of the deep web sources and results based on inter-source agreement. ACM Transactions on the Web, 7(2):Article 11, 1–32, 2013.*

[22] Mustafa Emmre Dincturk, Guy vincent Jourdan, Gregor V. Bochmann, and Iosif Viorel Onut. *A model-based approach for crawling rich internet applications. ACM Transactions on the Web, 8(3):Article 19, 1–39, 2014.*

[23] Kevin Chen-Chuan Chang, Bin He, Chengkai Li, Mitesh Patel, and Zhen Zhang. *Structured*

databases on the web: Observations and implications. ACM SIGMOD Record, 33(3):61–70, 2004.

[24] Wensheng Wu, Clement Yu, AnHai Doan, and Weiyi Meng. *An interactive clustering-based approach to integrating source query interfaces on the deep web. In Proceedings of the 2004 ACM SIGMOD international conference on Management of data, pages 95–106. ACM, 2004.*

[25] Eduard C. Dragut, Thomas Kabisch, Clement Yu, and Ulf Leser. *A hierarchical approach to model web query interfaces for web source integration. Proc. VLDB Endow., 2(1):325–336, August 2009.*

[26] Thomas Kabisch, Eduard C. Dragut, Clement Yu, and Ulf Leser. *Deep web integration with visqi. Proceedings of the VLDB Endowment, 3(1-2):1613–1616, 2010.*

[27] Eduard C. Dragut, Weiyi Meng, and Clement Yu. *Deep Web Query Interface Understanding and Integration. Synthesis Lectures on Data Management. Morgan & Claypool Publishers,2012.*

[28] Andr´e Bergholz and Boris Childlovskii. *Crawling for domainspecific hidden web resources. In Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on, pages 125–133. IEEE, 2003.*

[29] Sriram Raghavan and Hector Garcia-Molina. *Crawling the hidden web. In Proceedings of the 27th International Conference on Very Large Data Bases, pages 129–138, 2000.*

[30] Cheng Sheng, Nan Zhang, Yufei Tao, and Xin Jin. *Optimal algorithms for crawling a hidden database in the web. Proceedings of the VLDB Endowment, 5(11):1112–1123, 2012.*

[31] Panagiotis G Ipeirotis and Luis Gravano. *Distributed search over the hidden web: Hierarchical database sampling and selection. In Proceedings of the 28th international conference on Very Large Data Bases, pages 394–405. VLDB Endowment, 2002.*

[32] Nilesh Dalvi, Ravi Kumar, Ashwin Machanavajjhala, and Vibhor Rastogi. *Sampling hidden objects using nearest-neighbor oracles. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1325– 1333. ACM, 2011.*

[33] Jayant Madhavan, Shawn R. Jeffery, Shirley Cohen, Xin Dong, David Ko, Cong Yu, and Alon Halevy. *Web-scale data integration: You can only afford to pay as you go. In Proceedings of CIDR, pages 342–350, 2007.*

[34] Brightplanet's *searchable database dirctory. http://www.completeplanet.com/, 2001.*

[35] Mohamamdreza Khelghati, Djoerd Hiemstra, and Maurice Van Keulen. *Deep web entity monitoring. In Proceedings of the 22nd international conference on World Wide Web companion, pages 377–382. International World Wide Web Conferences Steering Committee, 2013.*

[36] Soumen Chakrabarti, Kunal Punera, and Mallela Subramanyam. *Accelerated focused crawling through online relevance feedback. In Proceedings of the 11th international conference on World Wide Web, pages 148–159, 2002*