ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

A Survey on Smart Web Crawler: Domain-Specific Crawler for Locating Deep-Web Content

Deepali Dhaygude

M.E. Student, Dept. of Computer Engineering, SKNCOE, Savitribai Phule Pune University, Pune, Maharashtra, India

ABSTRACT: As the huge amount of information in the deep web grows, there has been increased significance in techniques and tools that help efficiently locate deep web interfaces. The nature of the deep web is dynamic. Because of this and huge amount of web resources obtaining high efficiency and broad coverage is challenging issue. This paper proposes a novel two stage approach, namely, Smart web Crawler for harvesting deep web interfaces. First stage provides broad coverage where Second stage proves high efficiency. In the First Stage, Smart Web Crawler discovers relevant sites for the given topic. In the Second stage, it uncovers searchable forms from the site. Smart Web Crawler achieves higher harvest rate than other crawler.

KEYWORDS: Two-Stage web Crawler, Smart Web Crawler, Adaptive learning, deep web content.

I. INTRODUCTION

Due to large volume of web resources and the dynamic nature of deep web, achieving wide coverage and high efficiency is challenging issue. Wide coverage and high efficiency need to be maintained. The deep or hidden web means to the content lies behind searchable web interfaces that cannot be indexed by search engines. The deep web makes up about 96 percentage of all content on the internet. The deep web is 500-550 times larger than surface web. There is a challenge to locate the deep web databases because they are not registered with any search engines and these are sparsely distributed and keep constantly changing. To address the above problem previous work proposed two crawlers: Generic Crawler and Focused Crawler. Generic Crawler is mainly developed for characterizing deep web and directory construction of deep web resources that do not limit search on a specific topic but attempts to fetch all searchable forms.

Focused crawler would like to fetch only web pages that are relevant to a particular topic and neglect downloading all others. Focused crawler predicts the probability that link to a specific page is fetching the page. The objective of Focused Crawler is to select links that lead to pages of interest while neglecting links that lead non-relevant topics. They have been shown to lead to better quality indexes and to improved crawling efficiency than exhaustive crawlers.

There are 2 types of focused crawlers which searches online databases on a specific topic. They are given as: Form-Focused Crawlers (FFC) and Adaptive Crawlers for Hidden web Entries(ACHE). Form-Focused Crawler is formed with link, page form classifiers for crawling of web forms on a specific topic. ACHE is the execution of the Form-Focused Crawler, with added components. It is used for filtering the form adaptive link learner. The link classifier in FFC and ACHE plays an important role. That is, it achieves higher crawling efficiency than best-first crawler.

II. RELATED WORK

As the huge amount of information in the deep web grows, there has been increased significance in techniques and tools that permit users and applications to leverage this information. Crawling process should be efficient and neglect visiting huge unproductive part of the web. In “Searching for Hidden-Web Databases” paper, Luciano Barbosa and Juliana Freire proposed a new crawling strategy to automatically locate hidden web databases. This proposed strategy focuses on a specific topic. That is, selecting the links within topic that are more likely to fetch pages that contain forms with proper stopping criteria. The proposed crawler is more efficient than other crawlers. This crawler is

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

able to achieve a broad search as well as to retrieve a large number of searchable forms. Focusing the crawl on a specific topic helps to improve effectiveness of the link classifier.[1]

Most of datasets exist on web. But the useful datasets are not crawlable. These datasets cannot be discovered using conventional search engine technology. These datasets can be discovered and accessed automatically by using distributed information retrieval techniques. In "Automated Discovery of Search Interfaces on the Web" paper, Jared Cope, Nick Craswell, David Hawking proposed one method for automatically generating features for HTML forms. Automatically feature generation can be possible with specific parameters, such as Name and Value and Name and Word, found in the HTML form markup. First parameter is Name. This parameter is used for an input control. Second parameter is Value. This parameter is used for an input control. Third parameter is Name. This parameter is used for a form. Fourth parameter is a distinct word. This parameter is from a form action. Distinct word is nothing but the string of characters that exists between slashes.Automated feature generation was carried out separately on both the ANU (Australian National University) training set and the random web training set [3]. C4.5 learning algorithm is used because it is well known and implemented in multiple places and amenable to the types of features generated [3]. This algorithm generates a classification rule. This rule is implemented in any language and easily understandable.

The web has been increased with deep web interfaces. The deep web data lies behind the web. In "Toward Large Scale Integration: Building a MetaQuerier over Databases on the Web" paper, Kevin Chen-Chuan Chang, Bin He, Zhen Zhang proposed a MetaQuerier System, which is having dynamic and ad-hoc nature. The MetaQuerier System is used for discovering and integrating databases on the web. This paper focused on two goals. First, it helps to systematically access the deep web and discover online databases. Second, it helps to query online databases.

III. PROPOSED ALGORITHM

A. *DESCRIPTION OF THE ALGORITHM:*

Algorithm 1 : Reverse searching-

This algorithm is introduced when the crawler initializes and when the size of site frontier decreases to threshold. In Smart Web Crawler System, Firstly the result page from search engine is parsed to extract links. Then these pages are downloaded and analysed to decide whether the links are relevant or not. This can be done using heuristic rules. They are given as follows:

- If page contains related searchable forms then it is relevant.
- If the number of seed sites or fetched deep web sites in the page is larger than user defined threshold then page is relevant.

Algorithm 2 Incremental Site Prioritizing

One can make crawling process resumable and achieve deep coverage on websites. This can be achieved by using incremental site prioritizing. The main idea of incremental site prioritizing is to capture the learned patterns of deep web sites and form paths for incremental crawling. Firstly, the prior knowledge is used for initializing Site Ranker and Link Ranker. Then unvisited sites are assigned to the Site Frontier and unvisited sites are prioritized by Site Ranker and visited sites are inserted to fetched site list.

Smart Web Crawler accurately classify out-of-site links. Site Frontier uses two queues to save unvisited sites. These two queues are High Priority Queue and Low Priority Queue. Site Ranker is assigned relevant scores for prioritizing sites. The Low Priority Queue is used to provide more candidates sites.

IV. ALGORITHM

Algorithm 1 : Reverse searching

Input: seed sites and harvested deep websites

Output: relevant sites

1 while # of candidate sites less than a threshold **do**
2 // pick a deep website
3 *site* = getDeepWebSite(siteDatabase, seedSites)
4 resultPage= reverseSearch(*site*)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

5 *links* = extractLinks(*resultP age*)
6 foreachlinks in links **do**
7 *page* = downloadPage(*link*)
8 *relevant* = classify (*page*)
9if *relevant* **then**
10 *relevantSites*= extractUnvisitedSite(*page*)
11 Output relevantSites
12 end
13 end
14 end

Algorithm 2: Algorithm 2 Incremental Site Prioritizing
Incremental Site Prioritizing.

Input:siteFrontier

Output: searchable forms and out-of-site links

1 *HQueue*=SiteFrontier.CreateQueue(HighPriority)

2 *LQueue*=SiteFrontier.CreateQueue(LowPriority)

3 while *siteFrontier is not empty* **do**

4 if *HQueue is empty* **then**

5 HQueue.addAll(LQueue)

6 LQueue.clear()

7 end

8 *site* = HQueue.poll()

9 *relevant* = classifySite(site)

10 if *relevant* **then**

11 performInSiteExploring(site)

12 Output *forms* and OutOfSiteLinks

13 siteRanker.rank(OutOfSiteLinks)

14 if *forms are not empty* **then**

15 HQueue.add (OutOfSiteLinks)

16 end

V. CONCLUSION AND FUTURE WORK

This paper proposed domain specific crawler for locating deep web content, namely, Smart Web Crawler. It achieves both high efficiency and broad coverage. Smart Web Crawler is a focused crawler, which consists of two stages such as Site locating and In-site Exploring. Smart Web Crawler minimizes the number of visited URLs and simultaneously maximizes the number of deep websites.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

REFERENCES

1. Feng Zhao, Chang Nie, Heqing, Hai Jin, "SmartCrawler: A Two-stage Crawlers for Efficiently Harvesting Deep-Web Interfaces", IEEE TransactionsOn Services Computing, vol. 9, no. 4, July/August. 2016.
2. Lucianco Barbosa and Juliana Freire, "Searching for Hidden-Web Databases", In WebDB, pages 1-6, 2005.
3. Lucianco 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.
4. Lucianco Barbosa and Juliana Freire, "Combining classifiers to identify online databases", In Proceedings of the 16th international conference on World Wide Web, pages 431-440. ACM, 2007.
5. Kevin Chen-ChuanChang, Bin He, and Zhen Zhang, "Toward large scale integration: Building a metaquerier over databases on the web", In CIDR, pages 44–55, 2005.
6. Luciano Barbosa and Juliana Freire, "Combining classifiers to identify online databases", In Proceedings of the 16th international conference on World Wide Web, pages 431–440. ACM, 2007.
7. Jared Cope, Nick Craswell, and David Hawking, "Automated discovery of search interfaces on the web", In Proceedings of the 14th Australasian database conference-Volume 17, pages 181–189. Australian Computer Society, Inc.,2003.
8. SoumenChakrabarti, Martin Van den Berg, and Byron Dom, "Focused crawling: a new approach to topic-specic web resource discovery", Computer Networks,31(11):1623–1640, 1999.
9. Peter Lyman and Hal R. Varian,"How much information? 2003", Technical report,UC Berkeley, 2003.
10. Roger E. Bohn and James E. Short. "How much information? 2009 report on american consumers", Technical report ,University of California ,San Diego,2009.
11. Michael K. Bergman. White paper: "The deep web: Surfacing hidden value", Journal of electronic publishing, 7(1), 2001.
12. Yeye He, Dong Xin, VenkateshGanti, SriramRajaraman, 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.

BIOGRAPHY

Deepali Rajendra Dhaygude is a M.E. Student in the Computer Engineering Department, Smt. Kashibai Navale College of Engineering, SavitribaiPhule Pune University, Pune, Maharashtra, India. She received Bachelor of Computer Engineering (BE) degree in 2014. Her research interests are Data Mining, Information Retrieval, Web Mining, Knowledge and Data Engineering etc.