ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Special Issue 3, March 2018

Quick Review on SmartCrawlers

Shubhangi Gajanan Malas

Assistant Professor, Department of Computer Science and Engineering, Sanmati Engineering College, Washim, Maharashtra, India

ABSTRACT: As we all know that deep web is growing very quickly day by day; hence it is very tedious job to get the accurate contents from deep web. Deep web is nothing but the hidden webs-whose contents are hidden behind query forms and also with the interfaces which may have connection with database and that database may contain high quality information. Conventional search engines were not able to access hidden part of web as well as not able to index the hidden part of web. Getting information from this hidden web was hard job, hence crawler called SmartCrawler is designed. The given crawler is having two stages. At first stage site locating is done using search engine. And at second stage in-site investigation is done.

KEYWORDS: Keyword searching, site-locating, site-collecting, deep web, reverse searching.

I. INTRODUCTION

All over the world the internet is a vast collection of billions of web pages which contains large number of information or data arranged on number of servers. It is really challenging task to locate the deep web databases, because they are not recorded with any search engines, are generally sparsely distributed, and keeps continuously changing. The basic Crawler fetches web pages and harvest or filtered out the relevant pages. The web crawler is used to create a copy of all visited pages for later processing by search engines. Crawler crawls around the web-pages, gathers and categorizes information on the World Wide Web. The crawler contains of three parts: First is the spider, also called as crawler. The spider visits the pages, fetches the information and then follows the links in other pages within a site. The spider returns to crawled site over regular interval of time. The information found in the first stage will be given to the second stage. It is also well-known as catalog. The index is like a database, containing every copy of web-page that crawler finds. If a web-page changes then the copy is updated with new information in the database. Third part is software. This is a program that shifts millions of web pages recorded in the index to find matches to search and level them in order of what it believes as most relevant. There are three key processes in delivering search results to you are: Crawling, Indexing and Serving.

Searching on the Internet today can be compared to dragging a net across the surface of the ocean. While a great deal may be caught in the net, there is still a wealth of information that is deep, and therefore, missed. The reason is simple: Most of the Web's information is buried far down on dynamically generated sites, and standard search engines never find it. Traditional search engines create their indices by spidering or crawling surface Web pages. To be discovered, the page must be static and linked to other pages. Traditional search engines cannot "see" or retrieve content in the deep Web those pages do not exist until they are created dynamically as the result of a specific search. The deep Web is qualitatively different from the surface Web. Deep Web sources store their content in searchable databases that only produce results dynamically in response to a direct request. But a direct query is a "one at a time" laborious way to search.

The given SmartCrawler targets at hidden web interfaces using two stage frameworks. This given crawler not only classifies sites at first stage but also categories searchable forms at second stage. Instead of simply classifying links as relevant or not, it first ranks sites and then prioritizes links within a site with another ranker. Current web search engines include indices only a portion of the Web. The significant part of the Web is unknown to search engines. It means that search results returned by web searchers enumerate only relevant pages from the indexed part of the Web while most web users treat such results as a complete set of links to all relevant pages on the Web. Paper is organized as follows. Section II describes Existing System. The flow diagram represents the architecture of

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Special Issue 3, March 2018

SmartCrawler with its two stages. Proposed System is given in Section III. Section IV presents Experimental Results. Finally, Section V presents conclusion.

II. EXISTING SYSTEM

To leverage the large volume information buried in deep web, previous work has proposed a number of techniques and tools, including deep web understanding and integration hidden web crawlers and deep web samplers. For all these approaches, the ability to crawl deep web is a key challenge.

Olston and Najork systematically present that crawling deep web has three steps: locating deep web content sources, selecting relevant sources and extracting underlying content. Following their statement, we discuss the two steps closely related to our work as below. The IP based sampling ignores the fact that one IP address may have several virtual hosts, thus missing many websites. To overcome the drawback of IP based sampling in the Database

FFC with an adaptive link learner and automatic feature selection. Source Rank 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. Different from the crawling techniques and tools mentioned above, Smart Crawler is a domain-specific crawler for locating relevant deep web content sources. Smart Crawler targets at deep web interfaces and employs a two-stage design, which not only classifies sites in the first stage to filter out irrelevant websites, but also categorizes searchable forms in the second stage. Instead of simply classifying links as relevant or not, Smart Crawler first ranks sites and then prioritizes links within a site with another ranker.

Disadvantages:

- A recent study shows that the harvest rate of deep web is low — only 647,000 distinct web forms were found by sampling 25 million pages from the Google index..
- Existing hidden web directories usually have low coverage for relevant online databases, which limits their ability in satisfying data access needs.

III. PROPOSED SYSTEM

To get large volume of most relevant data number of techniques has been proposed. For all approaches the ability to crawl deep web is a key challenge. To address this problem, previous work has presented two types of crawlers, these are Generic crawlers and the Focused crawlers. A generic crawler which fetches all searchable forms and cannot focus on a particular topic. Focused crawlers are like Form-Focused Crawler (FFC) and Adaptive Crawler for hidden web Entries (ACHE) can automatically look for online databases on a individual topic. Form-Focused is designed with link, page, and build classifiers for focused crawling of web forms. It is expanded by ACHE with more components for form filtering and adaptive link learner. In the survey paper of How much information from 2003 of UC Berkeley[2] how much information is there in Berkeley is given. Some special conclusions of this paper are: It is not only statistically feasible to quantify the amount of information handled by society. There is still question that how to define the most fundamental measures for data and information. Some studies quantify information in binary digits, bits, word equivalents or number of hours consumption. Information quantity is not equal to information quality, but the second requires the first i.e., quantity must have quality. In the survey report of American consumers [3], also the report of 2009 on how much information is given. In this report it is given that total consumption of information in 2008 was 3.6 zettabytes (3.6*10^21 bytes) and 10845 trillion words. In the next reference paper of how much information is there in information society[4] and conclusion of this paper is that actually there is surprising lack of information about the amount of information in information society. In Crawling deep web entity pages [5] problem like query generation, empty page filtering and URL deduplication are handled.

A. Architecture of Smart-Crawler

The given figure shows the architecture of Smart-Crawler with its two stages. At very first stage Site locating is done. And in next stage In-site Exploring is done. In Site locating stage the canter page is find out sing search engine. This stage avoids visiting large number of pages as user enters for the page of his requirement. In next stage, i.e., in in-site locating stage a quick review is taken for getting the most relevant sites.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Special Issue 3, March 2018

Two stage Smart-Crawler works efficiently with its two stages. At site locating stage seed site that is input site is given to Smart-Crawler. When the number of unvisited URLs in database is less than threshold, then Smart-Crawler starts "Reverse searching" and saves these pages to database. Site frontier fetches web page URL from site database. Unvisited sites are given to site frontier and are prioritized by site ranker. And also unvisited sites are added to fetched site list. Site Ranker assigns score to each unvisited site. Site Ranker is improved during crawling by using Adaptive Site Learner. Site classifier classifies the URLs into relevant or irrelevant for given topic according to homepage content. After getting the most relevant site from first stage, in-site exploration is done. Links are stored in link frontier and corresponding pages are fetched. Then the forms are classified by Form Classifier to get searchable forms. Links are extracted into Candidate Frontier to prioritize links in Candidate Frontier. When the Crawler discovers a new site, then that URL is inserted into site database. Link ranker is improved by Adaptive Link Learner, which learns from URL path leading to relevant forms.

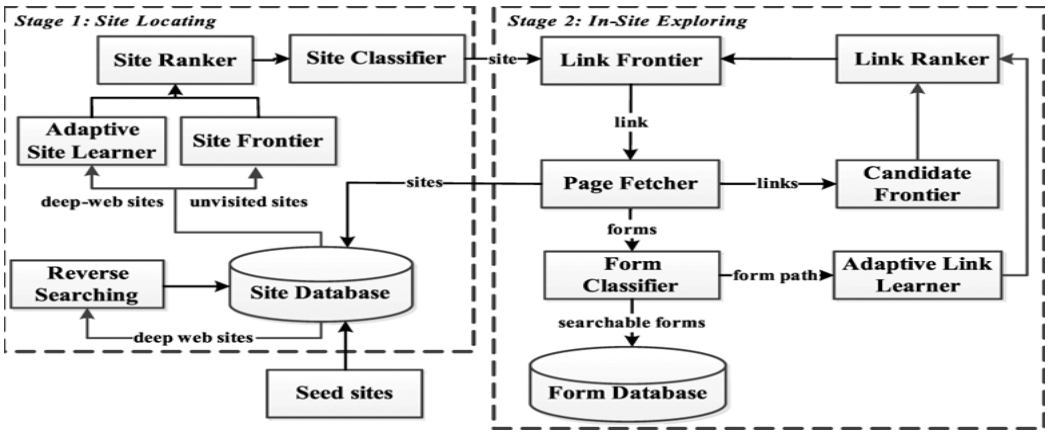

Figure: Architecture of Smart-Crawler with its two stages

❖ **Site Locating**

The site locating stage finds relevant sites for a given topic, consisting of site collecting, site ranking, and site classification.

➢ **Site collecting**

The traditional crawler follows all newly found links. In contrast, our SmartCrawler strives minimize the number of visited URLs, and at the same time maximizes the number of deep websites. Two crawling strategies are proposed. This includes reverse searching and incremental site prioritizing.

- ***Reverse searching***

  This strategy is used to find the canter pages of unvisited sites using search engines like Bing. To find weather the page is relevant or not following rules are used:

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

- ***Incremental site prioritizing***

  To make crawling process resumable and achieve broad coverage on websites, an incremental site prioritizing strategy is proposed. The idea is to record learned patterns of deep web sites and form paths for incremental crawling. First, the prior knowledge is used for initializing Site Ranker and Link Ranker. Then, unvisited sites are assigned to Site Frontier and are prioritized by Site Ranker, and visited sites are added to fetched site list.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Special Issue 3, March 2018

❖ **In-Site Exploring**

Once we find that the site is relevant, in-site exploring is done to get the searchable forms. The main goal is to quickly harvest or filter searchable forms and web directories of site. Some stoping criteria can be applied to avoid unproductive crawling. These criteria includes:

1. The maximum depth of crawling is reached.
2. The maximum crawling pages in each depth are reached.
3. A predefined number of forms found for each depth is reached.
4. If the crawler has visited a predefined number of pages without searchable forms in one depth.
5. A fifth criteria is the global criteria, which restricts the total pages of unproductive crawling.

IV. IMPLEMENTATION MODULES

Basically there are five modules. These are as follows-

1) Deep Web Search

- A user submits a web site link to deep web search.
- In contrast, our Smart Crawler strives to minimize the number of visited URLs, and at the same time maximizes the number of deep websites. To achieve these goals, using the links in downloaded webpages is not enough.
- When the crawler bootstraps.
- When the size of site frontier decreases to a pre-defined threshold.

2) Search key in Site Location

- In contrast, our Smart Crawler strives to minimize the number of visited URLs, and at the same time maximizes the number of deep websites.
- The result page from the search engine is first parsed to extract links. Then these pages are downloaded and analysed to decide whether the links are relevant or not using the following heuristic rules:
  - If the page contains related searchable forms, it is relevant.
  - If the number of seed sites are larger than a user defined threshold, the page is relevant.

3) Search key in External Site Links

- Once a site is regarded as topic relevant, in-site exploring is performed to find searchable forms.
- The goals are to quickly harvest searchable forms and to cover web directories of the site as much as possible.
- To achieve these goals, 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.

4) Ranking the Website

- In Smart Crawler, Site Ranker assigns a score for each unvisited site that corresponds to its relevance to the already discovered deep web sites.
- Link Ranker prioritizes links so that Smart Crawler can quickly discover searchable forms.
- A high relevance score is given to a link that is most similar to links that directly point to pages with searchable forms.

There is one more module called crawling multimedia files

5) Crawling multimedia files

- As website contains links to other pages likewise it also contains media files. Hence I have added this module.
- First we enter the web site URL and search all media files in the web sites
- Our crawler will find all the image files and show all description to the user.
- Also it stores all the image files in the specified location.

A. Types of SmartCrawlers

There are many web crawlers available. Some of them are: Lycos Infoseek, Exite, AltaVista and HotBot. These crawlers are used to index millions of pages. Some are explained below

a) Internet Archieve Crawler

In 1997, Mike Burner introduces the Internet Archive Crawler. The first paper was given which focused on the challenges caused by the scale of web. It uses multiple machine to crawl the web and it crawls about 100 million

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Special Issue 3, March 2018

URLs[7]. Each crawler process, read a list of input URLs for its assigned sites from a disk into per-site queue, and then it uses asynchronous I/O instructions to fetch pages from the sequences in parallel. It also deals with the problem of continuously changing DNS records, so it keeps the historical archive of hostname to IP mapping

b) Google Crawler

Later in 1998, the original Google crawling system has five crawling components which was running in various process and download the pages [8]. Each crawler process uses asynchronous I/O instructions to fetch the data from 300 web servers in parallel. Then all the crawlers transmit downloaded pages to a single Store Server process which compresses the page and store them on disk. Google Crawler is designed using C++ and Python. This crawler is combined with the indexing process.

c) Mercator Web Crawler

Heydon and Najorkhave given a web crawler which is called as Mercator Web Crawler. It is highly scalable and easily extensible[8]. It is written in Java. The first version given was non-distributed and later the distributed version was made available. Tese distributed versions ssplit the URL space over the crawlers according to host name and avoid the bottleneck of a URL server.

d) ***WebFountain crawler***

In 2001, another distributed and modular web crawler is given by IBM. This crawler has three major components, Multi-threaded crawling processes, duplicate content and central controlled process. These processes are responsible for assigning work. This crawler is written in C++ and uses MPI for communication between the various process. It was deployed on a cluster of 48 crawling machine.

e) ***IRLbot Web crawler***

It is recent crawler given by Yan et al.. It is designed for single process web crawler[6]. This crawler is able to crawl large collection of web pages without decreasing its performance[9].

f) ***ACHE Crawler***

ACHE is an adaptive crawler for harvesting hidden-web entries with offline-online learning to train link classifiers. We adapt the similar stopping criteria as SmartCrawler, i.e., the maximum visiting pages and a predefined number of forms for each site.

g) ***SCDI Crawler***

Experimental system similar to SmartCrawler, called SCDI Crawler is designed, which shares the same stopping criteria with SmartCrawler. Different from SmartCrawler, SCDI follows the outof-site links of relevant sites by site classifier without employing incremental site prioritizing strategy. It also does not employ reverse searching for collecting sites and use the adaptive link prioritizing strategy for sites and link.

V. CONCLUSION

In this paper, we propose an effective harvesting framework for deep-web interfaces, namely Smart Crawler. We have shown that our approach achieves both wide coverage for deep web interfaces and maintains highly efficient crawling. Smart Crawler is a focused crawler consisting of two stages: efficient site locating and balanced in-site exploring. Smart Crawler performs site-based locating by reversely searching the known deep web sites for center pages, which can effectively find many data sources for sparse domains. By ranking collected sites and by focusing the crawling on a topic, Smart Crawler achieves more accurate results. The in-site exploring stage uses adaptive link-ranking to search within a site; and we design a link tree for eliminating bias toward certain directories of a website for wider coverage of web directories. Our experimental results on a representative set of domains show the effectiveness of the proposed two-stage crawler, which achieves higher harvest rates than other crawlers. In future work, we plan to combine pre-query and post-query approaches for classifying deep-web forms to further improve the accuracy of the form classifier

REFERENCES

[1] SmartCrawler: A Two Stage Crawler For Efficiently harvesting Deep Web interfaces,pp year 2015
[2] Peter Lyman and Hal R. Varian. How much information? 2003. Technical report, UC Berkeley, 2003
[3] Roger E. Bohn and James E. Short. How much information? 2009 report on American consumers. Technical report, University of California, San Diego, 2009.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Special Issue 3, March 2018

[4] Martin Hilbert. How much information is there in the ”information society”? Significance, 9(4):8–12, 2012.
[5] 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.
[6] Olston and M. Najork , “Web Crawling”, Foundations and Trends in Information Retrieval, vol. 4, No. 3 ,pp. 175–246, 2010
[7] M. Burner, “Crawling towards Eternity: Building an Archive of the World Wide Web,” Web Techniques Magazine, vol. 2, pp. 37-40, 1997.
[8] Allan Heydon and Marc Najork. Mercator: A scalable, extensible web crawler. World Wide Web Conference, 2(4):219–229, April 1999.
[9] Jenny Edwards, Kevin S. McCurley, and John A. Tomlin. An adaptive model for optimizing performance of an incremental web crawler. In Proceedings of the Tenth Conference on World Wide Web, pages 106–113, Hong Kong, May 2001. Elsevier Science.
[10] The UIUC web integration repository. http://metaquerier.cs.
[11] uiuc.edu/repository/, 2003.
[12] Open directory project. http://www.dmoz.org/, 2013.