ISSN No. 0976-5697

Volume 8, No. 3, March – April 2017

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

A Survey on Crwaling Strategy

Swati G. Bhoi
M. E. Scholar, Dept. Of CSE
R. C. Patel Institute of Technology
Shirpur, Maharashtra, India

Prof. Ujwala M. Patil
Associate Professor, Dept. Of CSE
R. C. Patel Institute of Technology
Shirpur, Maharashtra, India

Abstract: The vast collections of web pages are available on the Internet. Web search engine strives to gather information as more relevant as possible to the user is a hard task. Nowadays maintaining high efficiency is a difficult issue because of the large amount of information available on the web as well as deep web has dynamic nature. As crawler plays important role in such cases. Here we surveyed how the smart crawler can provide accuracy and highest harvest rate than other crawlers due to its two-stage framework. The first stage is site locating, the search engine can search highly ranking pages to avoid visit of a large number of pages. The second stage is a site exploring; it uses a link tree to balance link prioritizing of relevant links for fast searching.

Keywords: Crawler, URLs, FFC, ACHE, Deep Web.

INTRODUCTION

In web crawling, the information on the World Wide Web is gathered and categorized by a crawler. The alternate name of the crawler is a robot as well as we can also call it as a spider. The crawler is the system which works as downloading the bulk of web pages. The web archiving used a web crawler which collects the exceeding set of web pages periodically and archived for posterity. It can also use in web data mining, where web pages are analyzed for statistical properties. The web monitoring provided services in which clients can submit standing queries or triggers and continuously crawls the web and notify clients of pages that match those queries [1].

Crawler contains three stages as spider, index and software. The spider meets the pages, mines the information and then follows the links which are present in other web pages within a site. The spider crawler's site over the same time interval. The catalog is the process in which second stage follows information which is found in the first stage, the index. The crawler finds some web pages from that each copy of the web page includes in an index and when the web page change then information updated in the database. The software contains millions of web pages, these are recorded in an index and it can be evacuate to find matches to search after that it ranks web pages as per their relevancy [1].

The Generic crawler and Focus crawler are two types of crawler. Web page included many links this all links are followed in Generic crawler hence evoke all searchable forms, but does not perform topic specific searching. Focused crawlers contain two types as Form-Focused Crawler (FFC) and Adaptive Crawler for Hidden-web Entries (ACHE). So each topic has an online database which is searched by FFC and ACHE [1].

Crawler also crawls the deep websites. The World Wide Web included hidden web. The large size of information present on the web, due to that high efficiency is challenging issue. The Deep web is rapidly growing day after day and to locate them efficiently there is need of effective techniques such as a Smart Crawler, contains two stage frameworks: site locating and site exploring [1]. For detecting appropriate sites, site locating used two techniques which are reverse searching and incremental two-level site prioritizing. Link ranker is spontaneously constructed by online feature selection which is used in the adaptive learning algorithm. In Smart Crawler search engine helps to find highly ranked pages and also avoid meeting a large number of pages to prioritize relevant sites. Link tree data structure used for balance link prioritizing and also removing the bias towards web pages in popular directories and relevant links are prioritized for fast in-site searching which is designed in the second site exploring stage.

LITERATURE REVIEW

Olston *et al.* described three steps for crawling the deep web: locating deep web content sources, selecting relevant sources and extracting underlying content [1]. Generic crawler does not focus on insistent topic, but try to fetch all searchable forms. In Database crawler using IP-based sampling, web server starting from the root pages and performing shallow crawling to crawl the pages. The problem with IP-based sampling is one IP address has many virtual hosts , but IP based sampling ignore this fact, but this problem solved in the database crawling because it uses random sampling in which Host graph characterize national deep web. It can meet to page which extracted from respective link and avoid. Linking to off-topic regions develop the focus crawler. A page classifier is component of best-first focused crawler, which uses to guide the search. In this crawler as per topic relevancy it can classify pages as well as provide priority to links. Finally a crawler must extract the content lying behind the form interfaces of the selected content sources. As the content of web pages are dynamic to obtain a more recent snapshot of its content Olston *et al.* used batch crawling and incremental crawling [1]. In batch crawling, the crawl order does not contain duplicate occurrences of any page, but the entire crawling process is periodically halted and restart as a way to obtain more recent snapshots of previously crawled pages. Whereas in incremental crawling, Web pages may appear multiple times in the crawl order and crawling is an uninterrupted process that conceptually never aborted. The most modern crawlers perform incremental crawling, which is more powerful than the batch crawling because it allows re-visitation of pages at different rates [1].

Satyawan Dongare *et al.* developed the Meta search engine. The first Meta search engine was created during the year of 1991-1994. It provides the access to many search engines at a time by providing single query as input GUI. It is known as a Meta crawler. In a Meta search engine the searching result is collected by using multiple search engines. For the full text of web pages it is finding the unique key word

phrases, quotes, and Knowledge. Site locating and in-site exploring are two stages of Meta crawler. It is worked same as smart crawler. But the difference is that in Meta crawler Search engines allow users to enter keywords and then examine this keyword in its table followed by database [2]. The number of machines is used in Internet Archive Crawler to crawl hundred million URLs at a time and it uses several machines to crawl the web. Each crawler fetches pages from a per site queues simultaneously and to fetch these pages it uses synchronous I/O instructions. This crawling process first reads an overall list of seed URLs for its assigned sites from disk into a per site queue. In Google Crawler, 300 web servers provide information simultaneously and to do that it can use asynchronous I/O instructions. Then single Store Server process includes all the downloaded pages and after that it compressed the page and stores them on disk. Google Crawler was based on C++ and Python. The Mercator web crawler written in Java. The first version of Mercator was non-distributed and later the distributed version. All distributed version split up the URL space over the crawlers according to host name and avoid the potential bottleneck of a centralized URL server. Multi-threaded crawling processes, duplicate content and centrally controlled process are three major components of Web Fountain crawler which is responsible for assigning work. It was written in C++ and used MPI to facilitate the communication between the various processes. The In IRLbot Web crawler is a single process web crawler. It is able to scale to extremely large web collection without performance degradation. It can download the 6.4 billion web pages [2].

Allan Heydon *et al.* presented the Mercator crawler. This crawler is scalable and extensible, which can extract tens of millions of web documents. Due to the use of the bounded amount of memory, Mercator helps to achieve scalability and it can implement data structure, due to this reason the large set of data structure stored in hard disk and less set stored in memory for efficiency. Mercator arranged in a modular way with the prospect that other person added new features or functionality in that, then and then it's known as Mercator is extensible. There are two crawlers as Google crawler and Internet archive crawler. In Google crawler contain five components in which several crawler processes received URLs from a URL server. The indexer is another component of crawler which read back pages from disk. The pages are stored in a different disk file which extract from HTML pages. A URL resolver process stores the correct URLs to the disk file which is read by the URL server before that it reads the link file, derelativizes the URLs contained therein. In Internet Archive basically it needs between four to eight machines to design the entire system, but first it can use three to four crawler machines. Each crawler process added 64 sites to crawl, but at one time only one site assigned to one crawler. First page is downloaded after that link contained in downloaded page is extracted. If a link contains appropriate content then it stores in site queue otherwise, it adds to disk [3].

Jenny Edwards *et al.* have designed the Web fountain crawler periodically crawl the entire web. In this the repository contains data about data of each page up to one megabyte which is to be used for indexing, mining, etc. This crawler acts as incremental crawler because it crawl original page as well as update the copy of each page which is stored in the repository. Web fountain crawler is fully distributed and incremental. Distributed that means it can distribute responsibility for scheduling, fetching, parsing and storing among a homogeneous cluster of machines. The sites contain a cluster to which a group of URLs is assigned. It does not contain any global scheduler, and any global queues to be maintained. In Web fountain crawler all components communicate using message passing interface which has three components as Ants, duplicate detectors and controller. The Ants are machines assigned to crawls sites, duplicate detector identify duplicate and machine cluster has control point which control by controller and Ants store a dynamic list of sites [4].

Luciano Barbosa *et al.* developed new crawling strategy. This crawling strategy used to spontaneously locate hidden-Web databases. It avoids the need to crawl a large number of inappropriate pages also need to perform a broad searching. The strategic focus on a given topic which contains some links from this all links, it can select the specific link which follows within topic, these links lead to pages that contain forms. This approach is efficient and effective because in this number of pages retrieved as form is greater than other crawler. Author studied different crawling strategies. The form focus crawler is select link and avoiding link that does not contain a topic related information. Also described the best first focus crawler follows all links in a breadth first manner and assign priority to links also uses page classifier, exhaustive crawler to supervise the search. Database crawler uses for locating online database which neither focuses the search on the topic nor attempt to select the most promising links. It is used as a seed for crawler IP addresses of valid web servers, using a breadth first search it crawls up to fixed depth from the root page of these servers. Page classifier and link classifier are two main classifiers of form focus crawler. The useless forms are filtered out by third classifier which is formed classifier. Topic wise classification performs in Page classifier. If the form is not already present in Form Database then it's added to form database but before that the form classifier decides it is a searchable form or not. The link classifier trained links that lead to pages that contain searchable form [5].

Luciano Barbosa *et al.* presented two crawlers a Adaptive crawler and Form Focus Crawler. Form Focus Crawler has two main drawbacks hence author developed Adaptive crawler. Form focus crawler focuses only on the topic and its work more efficient. The two drawbacks of FFC are the set of forms extract by the FFC is highly confusing or heterogeneous and training and updating link classifier is another headache which required more time. This problem address by adaptive crawler which has four components: the expected reward increases by behavior generating element(BGE) in which it can selects current state as an action, the problem generator (PG) that is responsible for suggesting actions that will lead to new experiences, as per action it gives the online learning element feedback on the success (or failure) and the element is online learning element used to update some policy of BGE which takes the critic's feedback into account. Author developed Adaptive Crawler for Hidden Web Entries (ACHE) contain learning elements is an adaptive link learner, so the features automatically mine from a successful path by using feature selection component. The accuracy of the form filtering process is more hence adaptive link learner is more effective than FFC [6].

Soumen Chakrabarti *et al.* described the focus crawler which prioritizes pages as per topic. General crawler follows all links present on a page and copy the whole content of the web page but focused crawler perform a topical search means it concentrate on fixed topic and it will be quickly identify changes to pages. The classifier in focus crawler supervises to identify the relevance from examples embedded in a topic domain, and a distiller which identifies topical benefit points on the Web. Classifier, the distiller and crawler are three main components of focus crawler. The web page contains a large amount of content and many web pages, each page included

many links so classifier makes as per topic, refer to pages crawled to decide on link expansion and to identify the visit priorities distiller appoints a measure of centrality of crawling pages. Author also introduced that when users searched specific topic three different crawls were done: unfocused crawl, soft focused and hard focused crawl. Alta Vista followed traditional content distillation and by using this it can assemble URLs by keyword query and to remove inappropriate pages some screening done by hand. In unfocused case, crawling speed will slow when new URLs fetches in pseudo random order. Same as unfocused crawl focus crawls also start with highly relevant links, but the relevance goes rapidly to zero because focus crawler lost its relevancy within the next bulk of pages retrieved. As compared to unfocused author stated that hard focused crawl is highly efficient because it acquiring appropriate pages over thousands of pages in a short time. Similar observations hold for the soft focused crawler it is difficult to compare between hard and soft focusing, they both do very well [7].

Mangesh Manke *et al.* reviewed that data divide in multiple servers using Hyper Text Markup Language. The information is extracted when there is obstruction due to the size of the collection is large. Generally user performs searching on search engine and if he found relevant information he gain information from a search engine, web crawler is one of the building blocks of search engine and for data management and scrutiny a web crawler around the Internet collecting and storing it in the database. Nowadays, the many users searching information on the Internet, these users limit their searches to the online, thus the specialization in the contents of websites will limit this text to look engines. A look engine contained spiders which is a special code robots, the many words found on websites and it make a list of the words and search information on the many ample sites that exist. Once a spider is building its lists, the application is termed the net crawling. So as to make and maintain a helpful list of words, a look engine's spiders ought to crosscheck plenty of pages. Author developed the Crawdy which has the same behavior as Smart crawler means contain two stage frameworks as site locating and in site exploring. The Crawdy works effective than SCDI (site-based crawler for deep Internet interfaces) and ACHE. ACHE is also known as an adaptive crawler for gathering hidden-web entries with offline-online learning to coach link classifiers. SCDI shares constant stopping criteria with Crawdy, totally different from Crawdy, SCDI follow the out-of-site links of relevant websites by site classifier while not using a progressive site prioritizing strategy. The Crawdy is our projected crawler for gathering deep net interfaces, almost like ACHE, Crawdy uses associate degree offline-online learning strategy, with the distinction that Crawdy leverages learning results for web site ranking and link ranking [8].

Mustafa Emmre Dincturk *et al.* developed new technology Rich Internet Application (RIAs) which is also known as AJAX. The previous web application techniques are not sufficient for RIAs. Traditional methods are breadth first search and depth first search, the breadth-first explores the neighbor nodes first, before moving to the next level neighbors' , whereas the depth-first crawling strategy explores the most recently discovered state first. The model based crawling is the new methodology to work more RIAs efficient. This model based crawling contain hypercube strategy in which chain decomposition is performed by the set of chain that means hypercube which overlay every element. Author developed some Meta models based on model based crawling. In Menu model, each event classified in one of three categories. Menu event is events that always bring the application to same state regardless of state from which these events were executed and self-loop event that always bring the application back to the state from which these events were executed. In Probability model for each event a probability of finding a new state is calculated. It can be seen as a fineness of the menu model with a dynamic range of categories in which events are assigned and where the category in which an event is assigned changes over time [9].

Yeye He *et al.* presented entity oriented site over document oriented textual content. Deep-web site, Wikipedia, twitter, etc. follows document oriented textual content, but it opposed by some sites like online shopping sites. So these sites follow entity oriented textual content which contains two main properties. First is the entity-oriented site brings peerless opportunities and second is entity pages are to be crawled from a large number of entity sites. The thousands of sites as input, the realistic objective is to only obtain a representative content coverage of each site, instead of an exhaustive one. Some crawling techniques involved in entity oriented sites. The homepages available on sites are crawled by URL template generation. Some web forms founded on these homepages, parsing perform on that and it can generate URL template pages and URL templates. This URL template generates final URLs which will be collected in a central URL repository [10].

Trupti V. Udapure *et al.* presented the three components of crawler which are Frontier, Page Downloader and Web repositories. Seed URLs are the set of URLs, the working of Crawler start with seed URL. The list of unvisited URLs store in frontier, then page related to retrieve URL is downloaded by Page Downloader after that web repositories store web pages in the database which are received from crawler [11]. Trupti *et al.* studied various crawling strategies. The Focus web crawler has alternate name is Topic Crawler. This topic crawler performs topic related searching that means this crawler is focused only on topic and it is economically feasible in terms of hardware and network resources. Incremental crawler overcomes limitations of traditional crawler. When a new web page is available traditional crawler replace this existing document with newly downloaded documents but incremental crawler visits the existing pages again and again and periodically refresh them due to that there is no need of replacing the pages at each time hence incremental crawler provide valuable data to the user. In Distributed crawler, the synchronization of nodes and communication between them are managed by a central server and it is robust against system crashes. In parallel crawler, multiple crawler run in parallel time that means contain multiple parallel process can run on a network of workstation and it downloaded pages within a reasonable amount of time. But in this various crawling strategies most beneficial strategy is a Focus crawler because it provides topically relevant information within small amount of time and it is designed for advanced web users focuses on particular topic [11].

Searching for hidden web resources is the major problem so it leads to develop a Smart Crawler which contains a two-stage framework. The candidate frontier extracts the links from these pages, and to prioritize them it can ranking link using Link Ranker. Site URLs, is the Site's database contains a set of the entire site URL. Link Ranker extracts the appropriate forms which are related to extracting URLs, hence Link Ranker responsible to improve performance of the Adaptive Link Learner.

SYSTEM ARCHITECTURE

Smart Crawler is designed with a two-stage architecture, site locating and in-site exploring as shown in figure 1. In the

first stage site locating search the most relevant site for a given topic, and in the second stage in-site exploring provide priorities to sites. The seed sites are the collection of several sites store in the site database. Site locating stage is started from the seed sites. The URLs of seed sites are selected to search other pages and other domains. Reverse searching performs when the number of unvisited URLs in the database is less than a threshold value. Reverse searching searches the highly rank pages and also supply these pages to site database. The site frontier will extracts webpage URLs from the site database. The site frontier take un-visited sites and site ranker provide ranks to it, whereas there are various visited sites which are stored to fetched site list. There is various unvisited sites are present so site ranker assign scores to such unvisited sites as per their relevancy. An Adaptive Site Learner is responsible for improvement of the Site Ranker. It will adaptively learn about features of deep-web sites (web sites containing one or more searchable form) found. As per homepage content Site Classifier categorizes URLs as per relevancy for a given topic so it can maintain its accuracy. After the most relevant site is found in the first stage, the second stage performs efficiently in-site exploration for excavating searchable forms.

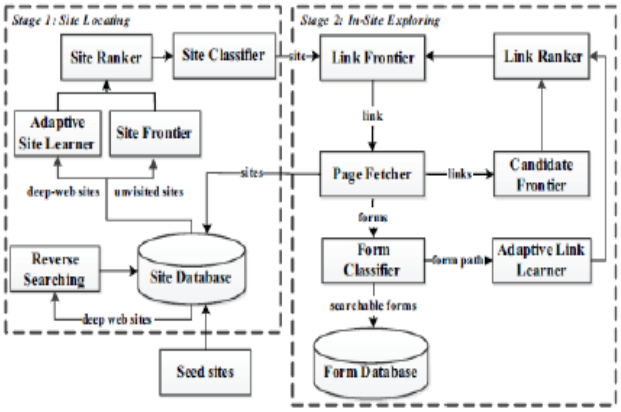

Figure 1. The two stage architecture of Smart Crawler

Each web page contains various links which leads to pages are stored in Link Frontier ,so using these links corresponding pages are fetched. To searches all searchable forms Form Classifier categories form. Additionally, the candidate frontier storess all links on these pages is fetch. The link ranker prioritizes links in Candidate Frontier. When new site is coming then this site URL is inserted into the Site Database. The Link Ranker is adaptively improved by an Adaptive Link Learner, which learns from the URL path leading to relevant forms. In site locating searches appropriate site for given topic included with site collecting, site ranking, and site classification. Site Collecting finding out-of-site links from visiting web pages may not be enough for the Site Frontier. So the size of Site Frontier may decrease to zero for some sparse domains. So to solve this problem, there are two crawling strategies, reverse searching and incremental two-level site prioritizing, to find more sites. Reverse searching is performed and it can work as per reverse algorithm. The algorithm starts working when the crawler starts crawling and also if the site frontier decreases below the predefine threshold. In Incremental site prioritizing included two queues namely high priority queue and low priority queue. The low priority queue is used to provide more candidate sites. Once the high priority queue is empty, sites in the low priority queue are pushed into it progressively.

After that, the site ranker assigns a score for each unvisited site that corresponds to its relevance to the already discovered deep web sites. After ranking site classifier categorizes the site as topic, relevant or irrelevant for a focused crawl. In site exploring the relevant links are prioritized for fast in-site searching. It contains the link ranker and Form classifier. In Link ranker prioritizes links so that Smart Crawler can quickly discover searchable forms. A high relevance score is given with a link that is most similar to links that directly point to pages with searchable forms. The Form classifier Classifying forms, aims to keep from focusing crawling, which filters out non-searchable and irrelevant forms. The Smart Crawler is a two-stage crawler, which achieves higher harvest rates than other crawlers and it can try to provide more accuracy than other crawler. But when the number of sites is increasing then the process becomes slow that means the crawling time required more.

CONCLUSIONS

In this paper, we surveyed the different crawling techniques. One of them is smart crawler which provides effectiveness, accuracy and higher harvest rate from other crawler. It has two stage framework involving site locating and in site exploring. Smart crawler is a best crawler because in site locating it can rank collecting sites and focusing on crawling topic where as in site exploring fast in site searching perform by excavating most relevant link. And in the survey also review that the smart crawler can provide effectiveness, accuracy and higher harvest rate than other crawler.

REFERENCES

[1] C. Olston and M. Najork, “Web crawling,’’ in Foundations and Trends in Information Retrieval, vol. 4, pp. 175- 246, 2010.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] S. Dongare and K. Gawali, “Smart crawler: A two stage crawler for efficiency harvesting deep web interface,’’ in International Journal of Innovative Research in Computer and Communication Engineering, vol. 4, pp. 663-666, 2016.K. Elissa, “Title of paper if known,” unpublished.

[3] A. Heydon and M. Najork, “Mercator: A scalable, extensible web crawler,’’ in World Wide Web Conference, vol. 01, pp. 219-229, 1999.

[4] J.Edwards and K. S. McCurley, “An adaptive model for optimizing performance of an incremental web crawler,’’ in Proceedings of the Tenth Conference on World Wide Web, vol. 3, pp. 106-113, 2001.

[5] L. Barbosa and J. Freire, “Searching for hidden-web databases,’’ in International Work-shop on the Web and Databases, vol. 4, pp. 01-06, 2005.

[6] J. Freire and L. Barbosa, “An adaptive crawler for locating hidden-web entry points,’’ in International World Wide Web Conference, vol. 6, pp. 441-450, 2007.

[7] S. Chakrabarti and M. V. den Berg, “Focused crawling: a new approach to topic-specific web resource discovery,’’ in Computer Networks, vol. 7, pp. 1623-1640, 1999.

[8] M. Manke and K. Singh, “Crawdy: Integrated crawling system for deep web crawling,’’ in International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, pp. 389-393, 2015.

[9] M. E. Dincturk and G. V. Jourdan, “A model-based approach for crawling rich Internet applications,’’ in ACM Transactions on the Web, vol. 7, pp. 1-39, 2014.

[10] Y. He and D. Xin, “scrawling deep web entity pages”, In ACM, vol. 7, pp. 611-620, 2013.

[11] T. V. Udapure and R. D. Kale, “A study of web crawler and its different types,’’ in IOSR Journal of Computer Engineering, vol. 16, pp. 01-05, 2014.