IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693

Comparative Analysis of Hidden Web Crawlers

Ashok Kumar[1*], Manish Mahajan[2], Dheerendra Singh [3]

[1*]Research Scholar Phd Department of Computer Science & Engineering, IKG Punjab Technical University Kapurthala, Punjab, India
[2] Department of Computer Science & Engineering, CGC College of Engineering, Landra, Mohali, Punjab, India
3Department of Computer Science & Engineering, CCET, Chandigarh Punjab, India

[*]*Corresponding Author: er.ashokgoel@gmail.com, Tel.: +91.9717215215*

Available online at: www.ijcseonline.org

Accepted: 21/May/2018, Published: 21/May/20182018

***Abstract*—** Huge data on the internet is not available for the crawler of surface web to index. It can be accessed through search forms when required. This data cannot be accessed by using the hyperlinks present in a web page. Research on hidden web mainly focus on exploring ways to access databases that are usually present behind the search forms. The main effort was to put on how to fill the searched forms with meaningful values. This paper compares different type of hidden web crawler to mention the features and shortcomings.

Keywords*— *WWW, Hidden Web Crawler, Surface Web, Search forms etc

I. INTRODUCTION

To search any data on the internet majority of user depends of the search engines. No of search engines are present now a day like Google, Bing, Yahoo etc. which helps to search the desired data available on the internet. Majority of them provides access of surface web database, which can be accessed by crawling the hyperlinks and storing the metadata of pages for adding these in the search engine's index [1]. The search engine provides results based on the local copy. Lot of information present on the internet is in the form of hidden web. This information is not discovered by simply crawling the hyperlinks. Simple example is, searching flight data on a travelling web site though this data is available in public domain but it is available after submitting a searched form. Second category of hidden web is dynamic data which is available thorough web applications on real time based on the query like the railway ticket reservation system. The same query produces different data at different time. This signification portion of the publicly available information is not efficiently search by conventional search engine designed for general purpose searching.

The purpose of contribution in literature of hidden web crawler is compared based on the class that work on effectively retrieving and accessing hidden web data [2][3][4]. Motivation of the paper is to review and compare the strengths and weakness of the crawler's present.

II. SIZE AND CHARACTERISTICS OF HIDDEN WEB

The hidden web crawler provides entry to huge and continuously growing databased on the web. No of authors present the approximation about the size of hidden web. In 2001, an initial study [1] shows the size of hidden web is approximately 500 times the size of surface web. In 2004, to measure the size of hidden web databases a random IP sampling approach is used and the results of this approach showed that large part of this data in hidden web is structured [5]. In 2007 the analysis was done to know the percentage overlap between search engines that are mostly used such as google, MSN etc. and it was found that only 37% of the data is being indexed by the search engines [6]. This shows that hidden web is the fastest growing category of latest information present in the forms of documents and other media over the internet.

- Approximately 600 billion text documents are present
- Hidden web data is approx. 2500 times than that of surface web data
- More focused content that surface web sites
- Most of the publicly accessible information are not subject to subscription

III. HOW TO ACCESS HIDDEN WEB DATA

Hidden Web can be accessed by filling the form to the search forms which gives a list of links those are relevant pages on

the web. This list is further used to find the relevant web pages by crawling the links. To access the data on hidden web search forms are the only option. There are two basic approach for this:

- One module of search engine is surfacing which refers the activity that collects the most of the data in background and then update the index of search engine. The crawler needs to automatically fill the search forms and submit these forms so that it can download the resultant pages. These are further integrated with the index structure of available search engine. Though it is challenging task to compute the relevant form submission. As this is not an active task which crawler usually do offline. This technique is simple and easily apply.
- Virtual data integration is a technique which creates a virtual database structure and map each attribute of that domain with the fields of searchable forms. Then query of particular domain can with the resources by submitting the filled form to the database. API are used to fetch data from hidden web, the results are simply the responses of the API.

There is n no of domains present on the web and defining the limits of a particular area and then designing the database structure is a challenging task No of research has been done for integration of the system but the technical problem that are part of this integration approach motivate to accept the technique of surfacing as the path to success and discussed hereafter

IV. APPROACH TO SURFACE THE DATA ON HIDDEN WEB

The steps that are followed by human the same steps are followed by hidden web crawler to fetch the data like when searched forms are filled and then submitted to crawler it sends to the web downloads the indexed pages and then crawl the links for finding the actual web pages. Following generic algorithm for hidden web crawler have been proposed [4]

Algorithm

Step1: Repeat till resources are available.

Step2. Select a term to send it to the web let's assume it is Xi

Step3. Now query the web site based on the query qi it fetches the resultant pages it is denoted with the R(Xi)

Step4. Now download the (R(qi))

Step5. Exit.

Crawling of hidden web consists of two main steps

- Search the Resources
- To download the contents.

Step 1 involves automatically find the relevant web site which contains a search form interface. Step 2 involves filling out the searchable form with the data and fetching the data from the desired website. Due to the large size of the hidden web databases the common way to crawl the hidden web is

- Breadth-Oriented technique: The size of hidden web is very big; this approach focusses on finding more and more data sources rather than crawling the content inside one specific data source. The major issue in this approach seems to locate the hidden web resources and analyzing the returned results for learning and understanding the interface required to automate the process of content extraction.
- Depth-Oriented technique: This approach is to fetch the content from designated hidden web i.e. the purpose is to fetch maximum data from the given data sources. The challenge in this approach is to actively issue a query for the search interface for a designated database in order to fetch the data from the database and the cost to fetch this data should be minimum. The crawler must be intelligent enough that it automatically generates promising queries so as to carry out efficient crawling. This problem is known as query selection.

The data in databases are categorized either as structured or unstructured. Unstructured database contains plain text document. Unstructured data can be searched using simple keyword-based search where user type a list of keywords to fill the search interface. Structured database provides multi attribute search interface.

V. HIDDEN WEB CRAWLER

Crawling technique have been started since the starting of world wide web but the focus to fetch the data from hidden web extensively started in 2001. Raghvan and Molina focused on a design for fetching data from electronic database [2]. After that no of depth oriented hidden web crawler for structured and unstructured data have been developed. Let's have a review all these

Crawler for structure data based on Depth-oriented technique

Raghvan in 2001 stated the problem and model for solving this has been proposed model as shown below. This shows the interaction between crawler and search forms [2].

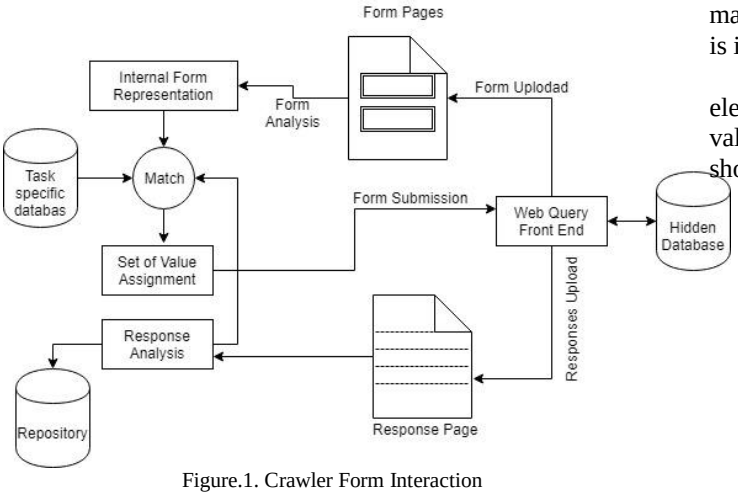

Figure.1. Crawler Form Interaction

This proposed model works as the input for hidden web crawler HiWE. The architecture of HiWE is shown in Figure 2.

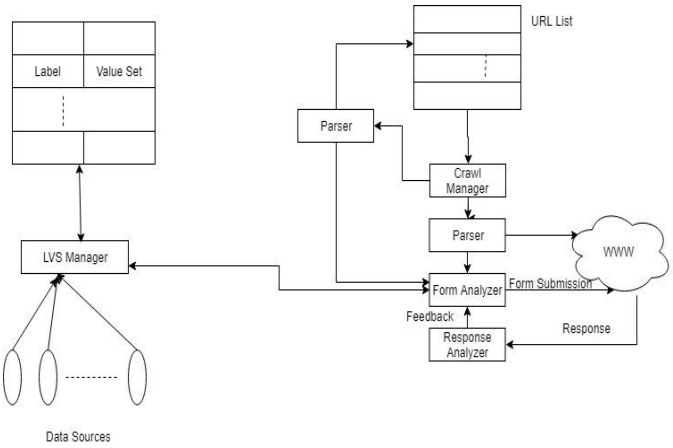

Figure. 2. HiWE Architecture

A method has been proposed to fill the search form either manually or collected it from the query interface. Form page denoted the page containing the search form and the page received after submission of form page is called as page received. A form contains the element either of text box, combo box, list box, text area, radio button or check boxes. Each element is identified with the label i.e. the text associated with every element on the web for which it tries to get the four closest texts to that element then one of them is chosen from a heuristic set formed by taking into candidate assignment for particular form this is generated from the value in the label value set(LVS). This table contains pairs of L &V, L represent the label and V represent the fuzzy set of all the possible values associated to the label. Although this architecture doesn't exhaust all of the possible assignment for a form. Multiple queries which are independent of each are input then cartesian product of all the input is done after this some specific URL are selected. This architecture has a major challenge in case no of domain for each form element is infinite.

Liddle et al. in 2002 proposed a technique to find form element and form a HTTP GET request inputting s default values for every field [7]. This flow of this architecture is shown using flowchart as shown in Figure 3.

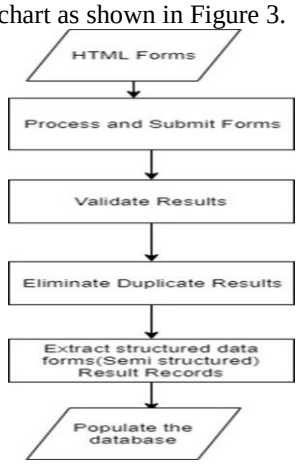

Figure. 3. Flowchart of liddels approach

This is not fully automated approach but it is simpler approach. It takes the data as and when required. With the help of this approach the significant percentage of data can be retrieved before submitting all the queries. In the first step of proposed technique a default query is issued to sample the searchable forms that is used to find whether this query is comprehensive and no of queries are issued to until threshold is reached.

Madhvan et al. discuss the approach used by Google for web form filling. An algorithm has been presented that select the input combination so as to effectively navigate to that search space by incorporating the generated URLs those seems appropriate for inclusion in web search index [8]. The first step is to form query i.e. the combinations of inputs. The next step develops an algorithm which can judge the appropriate input value for the various form fields.

Bhatia et al. proposed a hidden web crawler that domain specific which consider multiple input search forms [9]. This proposed architecture was divided into different parts. First part contains the downloading of searched form. Second part describes the domain specific interface mapper that identifies the relationship between the fields of different search interfaces. Then next step is merging these interfaces so to form the Unified search interface. The USI is filled automatically and submitted to the database. The downloaded pages are stored in page repository that maintain the documents retrieved and it stores the URL along with that. This is a fully automated crawler which submit the filled form and in response get the hidden web pages.

VI. BREADTH ORIENTED CRAWLER

Bergholz et.al proposed a framework which automatically finds the entry point to the hidden web [10]. Domain specific hidden web technique has been implemented that starts on

surface web using normal search engine to know about the hidden web in a particular domain. Crawling technique has been detected by the author which is used to find whether the search form is hidden web resource or not. Through experiments it is proved that hidden web databases are domain dependent. The proposed technique can work on both random mode of crawling and domain specific crawling.

Barbosa and Freire et. al proposed a Form Focused crawler to automatically identify the hidden web forms based on a particular topic [3]. The proposed architecture is shown in Figure 4. The architecture combines

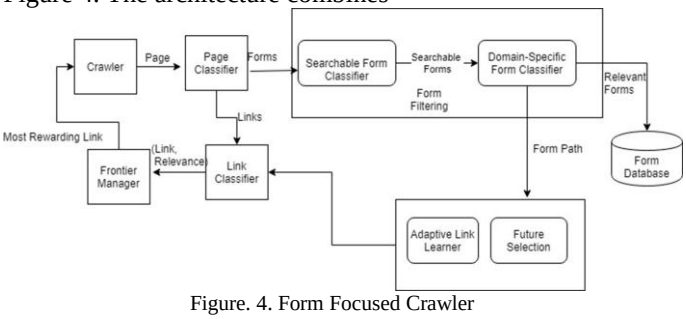

Figure. 4. Form Focused Crawler

page classifier and link classifier these were designed and trained in a manner so that these can crawl in particular domain. Firstly, the backward search is done which analyse and prioritize links those points to a searchable form in subsequent steps. The frontier manager is vital part of the proposed architecture which is used to choose the next designated link for the purpose of crawling. Form classifier is also used to remove the forms that are useless and remaining forms that can be further used is added in database if the existence is not there.

In 2007, Barbosa et. al worked on the limitation of the FFC by proposing a new architecture Adaptive Crawler for Hidden-Web Entries (ACHE) in which crawler adapt and then learns to improve the behaviour from previous data. Suppose a set of web pages are given that is an entry point to the database, ACHE automatically search other forms efficiently in the same domain. The architecture of ACHE is shown in Figure. 5.

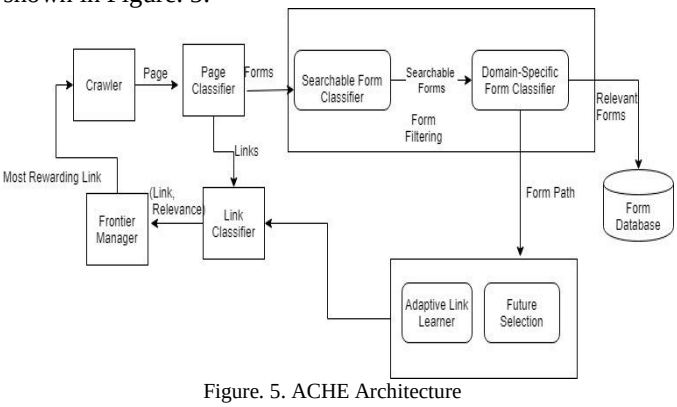

Figure. 5. ACHE Architecture

ACHE added two more classifier in FFC i.e. Searchable form classifier and domain specific form classifier. Searchable form classifier is used to classify whether the retrieved form is searchable or not. While the domain specific form classifier check whether the searchable form is part of particular area or not which we are trying to search. Adaptive link learner that dynamically adapts and learn automatically extracted from successful paths from feature selection and then update it to the link classifier.

In 2010 Bhatia proposed an architecture AKSHAR as shown in Figure. 6. It is based on multi attribute

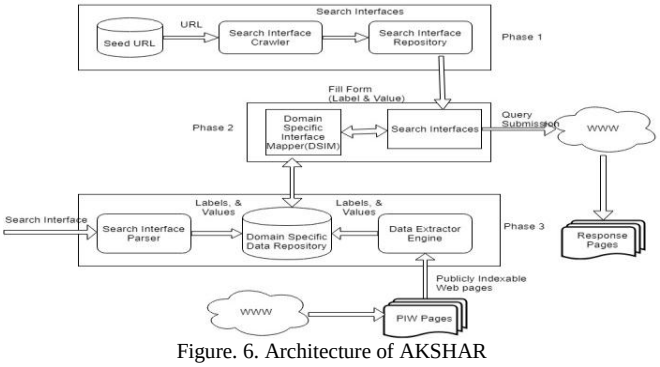

Figure. 6. Architecture of AKSHAR

or structured database. It uses Domain specific interface mapper (DSIM) to create unified query interface for a domain. It uses three types of matching 1. Semantic matching, 2. Fuzzy matching and 3. Domain specific thesaurus for finding the similarity between different interfaces of same domain. It calculates the re-visit frequency based on probability of change of web pages.

In 2017 Ranjan et. al proposed an architecture least cost vertical search engine based on domain specific hidden web crawler as shown in Figure 7.

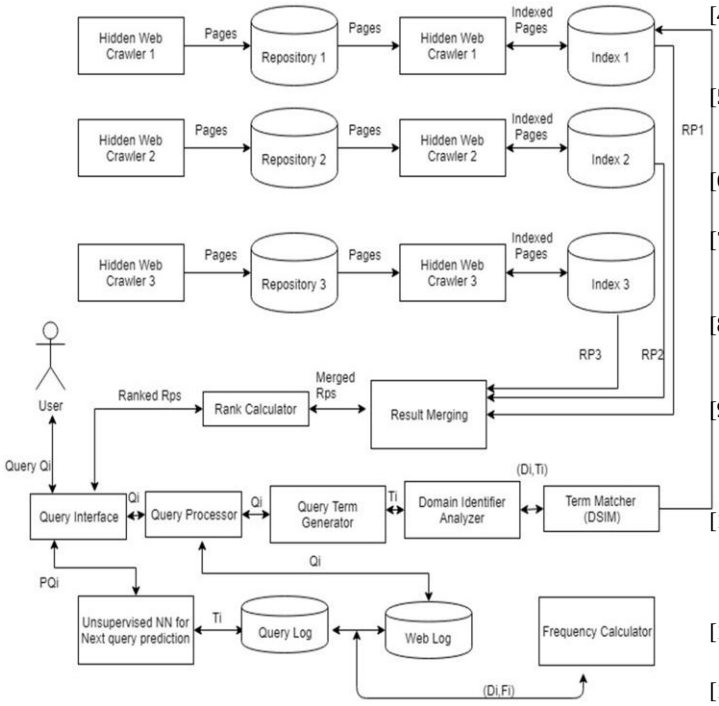

Figure. 7. Architecture of Least Cost Vertical Search Engine

The proposed architecture is a parallel crawler which has components like rank calculator to arrange the data in a particular order according to relevancy of the user query. Query log contains the record of queries used by various user at different time and it stores the database on the basis of the keyword of various domains and help in searching or firing the query whereas weblog maintains the action performed by the user Whenever user accesses the domain specific search engine, the web log updates the information and frequently stores it. The occurrence of each query in query log is calculated using frequency calculator.

VII. CONCLUSION:

Hidden web crawlers enable indexing, analysing and fetching the information from hidden web resources. The resultant data can further be used to categorize and classify. This paper discusses the various crawler that has been developed to fetch the relevant information from hidden web. Each of the proposed crawler have their own strengths and limitation but more research is required in this field to fetch the data effectively.

REFERENCES

[1] Michael Bergman, "The deep web: surfacing hidden value". In the journal of Electronic publishing 7(1) (2001).

[2] S. Raghavan, H. Garcia-Molina. Crawling the Hidden Web. In: the proceeding of 27th International conference on very large databases VLDB'01, Morgan Kaufmann publishers Inc. San Francisco, CA, p.p. 129-138.

[3] L Barbosa, J. Freire: Siphoning hidden-web data through keyword-based interfaces. In: SBBD, 2004, Brasilia, Brazil, pp.309-321.

[4] A. Ntoulas, P. Zerfos, J.Cho. Downloading Textual Hidden Web Content through keyword queries. In: 5th ACM/IEEE joint conference on Digital Libraries (Denver, USA, Jun 2005) JCDL05, pp. 100-109.

[5] K.C.Chang, B.He, M.Patel, Z.Zhang : Structured database on the web: Observation and implications: SIGMOD Record, 33(3), 2004.

[6] B.He, M.Patel, Z.Zhang, K.C. Chang: Accessing the Deep Web: A survey. Communications of the ACM, 50(5):95-101, 2007

[7] S.W. Liddle, D.W. Embley, D.T. Scott, S.H. Yau. Extracting data Behind web forms. In: 28th VLDB conference2002, HongKong, China

[8] J. Madhvan, D.Ko, L.Kot, V.Ganapathy, A Rasmussen, A Halevy: google's deep web crawl, In Proceeding of very large databases VLDB endowment, pp. 1241-1252, Aug 2008.

[9] Komal Kumar Bhatia, A.K.Sharma, Rosy Madaan: AKSHR: A novel framework for a domain specific hidden web crawler. In the proceedings of the first international conference on Parallel, Distributed and Grid Computing, 2010.

[10] A. Bergholz, B. Chidlovskii. Crawling for domain specific hidden web resources. Fourth international conference on web information system engineering (WISE'03) pp. 125-133. IEEE press, 2003.

[11] L. Barbosa, J. Freire. An adaptive crawler for locating hidden-web entry points. In proceeding of WWW, 2007, pp. 441-450.

[12] Sudhakar Ranjan, Komal Kumar Bhatia: "Design of Least Cost (LC) Vertical search based on Domain specific hidden web crawler" International Journal of Information Retrieval Research Volume7, Issue2, pp:19-33, doi:10.4018/IJIRR.2017040102, 2017

Authors Profile

Mr.Ashok Kumar is currently pursuing Phd from IKG Punjab Technical University. He has done his Mtech from YMCA Institute of Engineering and Technology, Faridabad. He has completed his MCA and BCA from Kurukshetra University, Kurukshetra. Currently he is working with Ventrue7 Technology Pvt. Ltd as a Lead QA automation

Dr. Manish Mahajan is currently working as Professor and HOD Department of Computer Science and Enginerring. Chandigarh Group of Colleges, College of Engineering. He has done his Phd from Punjab Technical University. Presently 8 students are doing their Phd under the supervision of Dr. Manish Mahajan.has 5 years of teaching experience and 4 years of Research Experience.

Dr. Dheerendra Singh is working as Associate Professor with Chandigarh College of Engineering and Technology in CSE department. He has approximately 15 years of experience. He has successfully supervised 3 Phd. Currently 5 more students are enrolled under Dr. Dheerendra Singh