International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-2, Issue-5 E-ISSN: 2347-2693

Deep Web Data Scraper: Search Engine

Sneh Nain[1*], Bhumika Lall[2]

[1*] *Computer Science Department, MDU University, India*
[2] *Faculty of Computer Science Department, MDU University, India*

www.ijcseonline.org

Received: 28/04/2014 Revised: 10 /05/2014 Accepted: 22/05/ 2014 Published: 31 /05/ 2014

***Abstract*—** World Wide Web is growing every day and people generally depend on search engine to explore the web. Searching on the web today can be compared to dragging a net across the surface of the ocean. Traditional search engine extracts data from the small portion of the web whereas the large portions of the web are hidden behind search forms, in searchable structured and unstructured database. Deep web contains the high quality content and large coverage area. A lot of research has been carried out in this area to make the hidden data float on the surface of web. In this paper, we discussed the problem faced by users in scraping the information from the deep web and also discussed the solution of these problems by using our new approach.

***Keywords*—**Surface Web; Deep Web; Search Engine; Deep Web Search Engine; Crawler; Indexer; Human Powered Directory

I. INTRODUCTION

Initially World Wide Web was small in size and easily gets browsed by using the hyperlink. Now days, the amount of information on the WWW is growing rapidly as well as number of new users inexperienced in the art of web researcher. This unabated growth of the web had resulted in a situation in which more information is available to large amount of people than ever in past human history. This unprecedented growth produced inevitable problem of information overload [1]. To remove this problem, users generally rely on search engine. Search engine are designed to help people in searching the information stored on other sites.

Traditionally search engine extracts data from the small portion of the web that is index able by following the hyperlinks which is called Surface Web. But large portions of the web are hidden behind search forms in structured and unstructured form which is known as Deep Web. It is also called by Invisible Web, Hidden Web and Deep Net.

Deep Web is World Wide Web content that is not part of Surface Web, which is indexed by standard search engines. It has high quality contents. World Wide Web contains 99% of deep web and 1% of surface web. It contains two type of component:-

- Surface Web
- Deep Web

Deep Web plays a significant role in extracting data. If users are missing the Deep Web, it means missing plenty of important and amazing resources of the web [7]. Their sources store their content in searchable database that only produce results dynamically in response to a direct request. In this paper, we discussed the problem faced by person in scraping content from the deep web and how search engine help in removing these problems.

Rest of the paper is organized as follows:- in Section 2 we will look at the different types of search engine, in Section 3 we define how deep web search engines work, Section 4 explain various deep web search engine, Section 5 show the comparisons of deep web search engine and Section 6 contain proposed work. Finally Section 7 concludes including some recommendations, contributions and Section 8 provides some suggestion for future work.

II. DIFFERENT TYPES OF SEARCH ENGINES

Web search engine is a software system that is designed to search for information on the World Wide Web. Searching should be done in two ways:-

- Human powered Directories
- Crawler Based Search Engine

A. Human Powered Directories

These directories are depends on human editors to create their listing. Webmasters built a directory which contains a short description about their websites or an editor write one for the sites they review, and these manually edited descriptions will form the search base. So changes made to individual web page will have no effect on how these pages get listed in the search result. Human Powered Directories are similar to a table of content that allow users to quickly turn to interesting section.

B. Crawler Based Search Engine

Crawler Based Search Engine create their listings automatically by using a piece of information to crawl or spider the web and then index what it finds to build the search base. If changes occur in web pages then it can be

Corresponding Author: *Sneh Nain*

dynamically caught by crawler based search engine and will show how these web pages get listed in the search results.

III. How Deep Web Search Engine Works

Deep Web includes the large part of total web approximately 99% which is much larger than surface web. Through previous section we have the knowledge about different types of search engines but these search engines only maintain directory and automatic listing of surface web.
Deep Web Search Engine operates in the following order:-

- Web Crawling
- Indexing
- Searching

A. *Web Crawling*

Crawler visits a web page, read the web page, extracts the links and then follows these links to other pages within the sites. It will return to the sites on a regular basis to look for changes on the web page. It is also known as Web Spider and a Web Scutter. Large quantities of web pages are laid in the deep web. These pages are accessible only by submitting queries to a database and regular crawlers are unable to find these pages if there are no links that point to them. Some of the hidden web crawlers are:-

- Deepbot
- HiWE (Hidden Web Exposer)
- Incremental Web Crawler

B. *Indexer*

Indexers receive the web pages from web crawler and index them. It will contain a copy of every web page that the Scutter finds. If changes occurred on web pages, then indexer will update with new information.

C. *Search Engine Software*

Search Engine is a software system that is designed to search for information on the World Wide Web. So this software system accepts the user-entered query, intercept it and investigate the millions of pages recorded by the indexer and ranks them in order of what it believes is most relevant and presents them in a customizable manner to the user. Among the these methods Web Crawler and Indexer are work on back end continuously whereas Search Engine Software is work on front end and it works only when user want to scrap some information or data.

IV. Deep Web Search Engine

Different deep web search engines have been created by the researchers to put the deep web on the surface. In this section, we are going to discussed various deep web search engine and its advantages and disadvantages.

A. *Hidden Seek*

Ntoulas developed a deep web search engine known as Hidden Seek [7, 11]. In this, he defined how effectively collect the data from the deep web and enables the users to search for information within the collected data. Hidden Seek is a deep web search engine that employs linguistic analysis to improve search relevance. Main component of Hidden Seek are:-

- Crawling the Hidden Web
- The Link Database
- Linguistic Processing
- Inverted Indexes
- Page Summaries
- Answering a Query
- User Interface

1) Crawling the Hidden Web

This component handles the task of retrieving pages from the web and storing them locally for processing. It identifies the links in the web page and follows these links in order to download other web pages. Crawler uses the sampling based policy to maintain a fresh subset of the web pages with a least overhead. After scraping all the links from the web page and send them to the link database. It also hands the page off to the linguistic analysis component.

2) The Link Database

Link Database assigns a globally unique ID to every link identified by the crawler. It also stores various static properties about the links. This complete information is used for maintaining the rank the web page and rescheduling the crawls.

3) Linguistic Processing

In this component, Search relevancy is improved by Natural Language Processing techniques like RAPIER, SRV, and WHISK etc. It means linguistic analysis should be performed in every page that the crawler downloads.

4) Inverted Indexes

It retrieves certain documents that are relevant to particular keyword and stores the ID of all the web pages for every term. It also stores certain other information like number of occurrence of the term in the document, a list of positional information, formatting information, functional attributes etc.

5) Page Summaries

It obtained the input data from linguistic processing module and store the HTML stripped version web pages in a compressed format. This module provides the complete summary of the document that displays the context around the query words.

6) Answering a Query

Web pages are obtained from the deep web with the help of standard keyword and phrase searching. It means we have to obtain the document which must contain all the keyword that the user specified. It is performed by doing the AND operation between the keyword and obtained the result list. Arrange the list so that the most relevant documents are

displayed first. Ranking is based on certain parameters like frequency of keyword in the document, keyword appear in the URL or title of the page.

7) User Interface

User Interface is a component through which user can enter its query. Users are free to type any textual query in the search box and search the deep web site that Hidden Seek has currently indexed.

B. Hidden Web Search Engine

Anuradha developed a deep web search engine to put the deep web data on the surface. This search engine is named as Hidden Web Search Engine [12]. It remove the disadvantage of Hidden Seek Search Engine, this search engine automatically fill the web query interface, extract the record according to the query and store these results in the repository for fast and efficient searching latter. Main component of the Hidden Web Search Engines are:-

- Query Interface Processing and Form Submission
- Domain Ontology
- Interface Unification
- Filling Attribute-Value Database
- Form Submission
- Hidden Web Data Extraction, Integration and Searching
- Table Area Detection and Extraction
- Dynamic Rule Generation
- Data Extraction
- Repository Formation

1) Query Interface Processing and Form Submission

This component provide an interface to user through which user can enter its web query. It automatically detects the domain specific search interface by looking for domain ontology in the source code of the web page.

2) Domain Ontology

Ontology usually provides a knowledge base which is required for the classification of search results. It organized the search results into hierarchical structure form. This structure helps the user in navigating, seeking and in finding the information more quickly, they are looking for.

3) Interface Unification

Interface unification is created by using the information stored in domain ontology. When interface get selected its attributes are sent for preprocessing and attribute matching. Matching is of two types either single matching or group matching.

4) Filling Attribute-Value Database

After performing the above steps, attribute value repository be maintained from which the values for particular attribute would be taken and search query form of the different site will be filled and submitted.

5) Form Submission

Unified Interface is filled by scraping data for each attributes and their field values. This interface is used to fill local interface, after matching the query with the local interface then the result page are extracted. Extracted results pages should be stored in the database for further processing.

6) Hidden Web Data Extraction, Integration and Searching

WWW contains structured data as well as unstructured data. But a large area of the hidden web is in structured form. It means in relational table form. So result pages contain the data in structured relational tables. The proposed hidden web search engine find the information from various deep web sources and after some processing, present it as a free web search service.

7) Table Area Detection and Extraction

Web page contains large amount of information, among these information some are required and some not. This component detects the table area, extract the relevant area and discard the other ones. This selects the required table area and sends it for record area extraction.

8) Dynamic Rule Generation

To find the relevant record area detection, firstly this component analyzes the behavior. Different web sites have different presentation styles so relevant area extraction can be done in two ways:-

- Similar Behavior
- Different Behavior

If few details of a product are clubbed in one cell then the information is displayed by product company takes human user into the consideration and not the extraction process.
If every detail is stored in different cells then no human assistance is required and also very easy to scrap the data.

9) Data Extraction

By analyzing the behavior, relevant area will be obtained. Now extract all child nodes from the area. This area contains rows and columns that should be extracted. After extracting rows and columns, each websites have the separate table and data is filled according to the data packed inside its result page.

10) Repository Formation

These tables at the end are merged. It means data should be collected and semantically labeled so that they can be organized into main repository and later own it is used for searching.

V. COMPARISON OF DIFFERENT DEEP W EB SEARCH ENGINE

Table 1 Comparison of Deep Web Search Engine

Deep Web Search Engine	Advantages	Disadvantages
Hidden Seek	Scrap pages from deep web and float them on surface web Maintain Repository fresh Detect spam website Different query selection used	Million of queries are defined to scrap deep data Work on single attribute database Time consuming Crawling should be done frequently Large repository required for mass storage of data
Hidden Web Search Engine	Automatically scrap pages from deep web and float them on surface web Works on multi valued attributes database Repository maintain database from different websites Provide web service of information sharing	Million of queries are defined to scrap deep web Queries executed again to maintain repository fresh Large repository required for mass storage of data Duplication of data record

VI. PROPOSED WORK

In section 5, we discussed about the advantage and disadvantage of deep web search engine. These disadvantages should be removed by doing following work. Web page contains the structured and unstructured text. Previous deep search engine focused only on structured text. In order to cover the unstructured text, we can do:-

- Wrapping by partitioning the web page into blocks based on heuristic
- Detect the relevant content block from the non-content block
- Compare it with the stored block to determine its behavior whether it is similar or not
- After checking its relevancy, design a Dom tree of each page
- Combined all the Dom tree that show the overall presentation of website which help in removing the unused or unessential matter
- When relevant content data get modeled and scrapped, applied page ranking mechanism
- Extracted data is shown in efficient manner

By using this technique, we can easily scrap text from structure and unstructured matter.

VII. CONCLUSION

Through this paper, we find that large amount of valuable information on the web is hidden behind search surface is known as deep web. Deep Web is a subset of total web. Certain researches should be done to float the information on the surface of web and help the user in searching. In this paper, we examine that how we can collect the data from deep web and enable the users to search for information within the collected data. It also shows how the deep web search engine works and what challenges are faced by search engine.

VIII. SCOPE FOR FURTHER RESEARCH

In future, this work can be extended by finding the more appropriate method, how efficiently we can store our data in repository and fast get accessed. So that overall efficiency of searching can be improved.

REFERENCES

[1] Bergman,Michael K., "White Paper: The Deep Web: Surfacing Hidden Value" Journal of Electronic Publishing Vol.7,Issue-1,**2001**.
[2] Ling Liu, James Caverlee, "Deep Web Data Extraction"
[3] Emilio Ferrrara, Giacomo F., Robert B., "Web Data Extraction, Applications and Techniques: A Survey" ACM Transaction on Computational Logic, Vol.5, June **2010**, pp.**1-20.**
[4] Brin, Lawrence Page "The anatomy of large-scale hypertexual Web Search Engine", Computer Networks and ISDN Systems, Vol.30, **1998**, pp.**107-111**.
[5] Laender, Silva, Juliana S., " A Brief Survey of Web Data Extraction Tools".
[6] Sriram R., Hector, "Crawling the Hidden Web" in the proceeding of the 27th VLDB Conference, Roma, Italy,**2001**.
[7] Babita, Anuradha, Ashish, "Hidden Web Data Extraction Tools" International Journal of Computer Applications, Vol.82,**2013.**
[8] Deep Web Website: //en.wikipedia.org/wiki/Deep_Web
[9] WikipediaWebsite: //en.wikipedia.org/wiki/Web_crawler
[10] Wikipedia Website: //en.wikipedia.org/wiki/Search_Engine
[11] Ntoulas, Zerfos, Junghoo Cho, "Downloading Hidden Web Content"
[12] Anuradha, A. K. Sharma, "Design of Hidden Web Search Engine" International Journal of Computer Application, Vol.30, **2011**.

[13] Chez Hong-ping, Fang Wei, Yang Zhou, "Automatic Data Records Extraction from List Page in Deep Web Sources" Vol.6, **2009**, pp.**370-373.**
[14] Chris Sherman, GARY Price, "Hidden web: Uncovering Information Sources Search Engines Can't See" CyberAge Book, **2001.**
[15] Manuel, Juan R., Fidel, Alberto Pan, "A Task specific Approach for Crawling the Deep Web" **2006.**
[16] Califf, M. E., and Mooney, R. J., "Relational Learning of Pattern-Match Rules for Information Extraction" In Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence (Orlando, Florida, **1999**), pp.**328-334**.
[17] Crescenzi, V., and Mecca, G., "Grammars Have Exceptions", Information Systems 23, 8, (**1998**), pp.**539-565.**

AUTHORS PROFILE

Sneh Nain, Did B.Tech in 2011 and now pursuing M.Tech in Computer Science Engineering. Her research interest includes Deep Web Information Retrival, Digital Watermarking, Cryptography and Networking.

Lecturer Bhumika Lall, received her B.Tech and M.Tech and now she is working as a lecturer in computer science department.