ENTRY POINT TO HIDDEN WEB: QUERY INTERFACE

Neelu Chaudhary
CSE, Manav Rachna College of Engineering
neelu.mrce@mrei.ac.in

Pooja Kapoor
CSE,Manav Rachna College of Engineering
pooja.kapoor26@gmail.com

ABSTRACT
The myriad of information is stored on the Web. Anyone hungry for information rushes towards the search engine to look for quality content. As finding content on the web is like playing hide and seek, sometimes end user is able to find it and sometimes it's hidden. The content is hidden because it is not indexed by the crawlers as this data is stored in the web databases and is accessible through HTML forms only. The content in web databases is of great value, but difficult to query and search. This paper discusses the role of query interfaces in hidden web and proposes an approach to uncover hidden web content through these query interfaces.

Keywords
Hidden Web, Crawler, Query Interface, Invisible web, HTML form

1. INTRODUCTION
The World Wide Web is growing with a huge pace and the information that is available to the user is in various forms like images, videos, websites, databases and many more. The rapid development of the World Wide Web has significantly changed the way in which information is accessed. So the web can be divided into Surface Web and Hidden Web [9]. Surface web consists of static pages or the interlinked information which can be indexed using crawlers that recursively follow these links. But much of the web is not accessible to crawlers as the content is not directly linked and is hidden behind search forms. This part of the web is referred to as Hidden web or Deep web. It has been found that Hidden Web is 400 to 550 times larger than surface web and also Hidden Web contains 7,500 terabytes of information compared to nineteen terabytes of information in the surface Web [9]. Unlike Surface Web, Hidden web consists of dynamic pages that are stored in the relational databases. It consists of content that the general-purpose search engines can't find due to access limitation or due to deliberate choice and is hidden simply because it is not indexed by a search engine but that doesn't mean that the content is always inaccessible, some pages are temporarily hidden and can be visible at a later date.

The reasons for the invisibility of content that can be accessed but not included in search engine index are as follows [1, 7]:

- Web pages that can be but are not, included in search engine indices due to restrictions on the depth of crawl, frequency of crawl, maximum number of viewable results, disconnected URLs. These web pages constitute a web called Opaque Web.
- Web pages that have deliberately been excluded by using various mechanisms such as robot exclusion principle, no index meta tag, password protected pages all these constitute a web called Private Web.
- Web pages those are visible only to those users who are willing to register to view them. This constitutes a Proprietary Web. Registration is fee based sometimes and web bots are unable to fulfill this requirement and therefore these web pages are not indexed.
- Web pages that search engines can't index due to technical reasons such as pages that use file formats like executables, flash files, and postscript are not crawled by traditional bots as they are not programmed to do so. Also, the information stored in relational databases is not indexed by the bots. These web pages make up a Truly invisible web.

2. CRAWLER
A crawler or spider or robot is a name given to a program that crawls the web for different purposes like a surface web crawler crawls to find hyperlinks in a web page and a hidden web crawler [4] crawls to find form in a web page and if it contains then perform semantic mapping and fill the form to retrieve the required content. The main goal to design such a program is to uncover the web as much as possible and index the text so obtained. This is essential as a user is able to see only those contents that are indexed by a search engine.
The algorithms for the surface web and hidden web crawler are as follows [5]:

2.1 Algorithm for Surface Web crawler

Input: Seed URLs

1. Pick one URL from a set of URLs
2. Download the page
3. Extract links and add them to Queue
4. Store the text in database
5. Go to step1

The traditional crawler starts with a set of seed URLs (can be famous sites or as per requirement). It then picks one of the seed URLs, download the corresponding web page and extract links if there are any. Store the links in the queue and text in the database. The scheduler picks one URL from the queue based on priority and the whole procedure is repeated. Crawler will continue crawling till the depth provided by the search engines or until all resources are used.

2.2 Algorithm for Hidden Web Crawler

Input: Seed URLs

1. Pick one URL from a set of URLs

2. Download the page

3. Extract links and add them to Queue

4. Check whether the downloaded page contains a form

4.1 If yes , analyze the form, assign values and submit the form to extract hidden content.

4.2 else goto step 1

A vast amount of information on the web is stored on the relational databases which is not indexed by the traditional search engines .Therefore to extract the hidden web content, hidden web crawler is required, which not only extract the links from the web pages but also analyze the web page to determine if it contains a search interface and if it has HTML form, fills it automatically and extract the content which is hidden behind the HTML forms.

3. QUERY INTERFACE

Query interface is simply a search form which is filled by a user to get the relevant information. To extract hidden web content is a challenging task as the pages are dynamically created through query interfaces. Therefore query interface has a major role in retrieving hidden web content from the web databases. Since query interfaces provide access to Web databases they have to visually convey information about the data in the databases so that the user can easily infer the necessary data he/she has to provide in order to obtain the desired information. An important aspect of user interfaces, is a sort of"spatial locality" property among the fields [2]. That is, related fields are usually placed close to each other in the interface. For example, in the automobile domain, the make and the model of a car are grouped together very often. Moreover, several related groups can form a super-group . As an example, the day, month and year can form the super-group date. Query interface consist of fields with predefined values, fields without predefined values, label to define the meaning of a field, an internal name which is used by programmers.

3.1 Language to Represent Query Interface: HTML

HTML is used to represent query interfaces, which is a hypertext markup language. HTML elements(fields) like textboxes, radio button, check boxes, dropdown menu are the building blocks of a web page which are shown in fig.1. HTML source code consists of various tags which are written in angular brackets (< >).The HTML source code corresponding to query interface in fig.3 is shown in fig .2. <FORM> tag is used to represent a search form and it consists of various attributes like action which contains the URL address of the form that will get open after submitting the form, method which tells the HTTP method (get or post) that is used to submit the form.

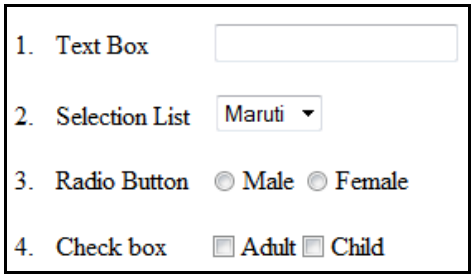

Fig. 1 HTML elements

3.2. Models to Represent Query Interfaces

There are two models that are used to represent query interfaces [2]. They are as follows:

3.2.1 Flat representation: This model represents the query interface as a list of fields where each field has a label, its internal name, and set of values.

3.2.2 Hierarchical representation: This model represents the query interface as a tree like structure. Hierarchical representation can be of two types:

3.2.2.1 Two level representation: Models a query interface as an ordered list of triplets of the form{attribute name, operator, operand}

3.2.2.2 Arbitrary depth hierarchies: Where a query interface is represented as an ordered tree of elements. The root of the tree is name of the website, internal nodes designate group of fields and leaves represent fields.

```
<html><body>
<form action="result.html">
<b>SEARCH CARS</b><br><br>
<table><tr>
<td>Make
<td><select><option>maruti</option></select>
</tr>
<tr><td>Model
<td><select><option>maruti800</option></select>
</tr>
<tr>
<td><input type= submit value=SUBMIT>
</tr>
</table></form></body></head></html>
```

Fig. 2 HTML source code

Fig.3 Query Interface

4. PROPOSED WORK

The size of hidden web is growing with a faster pace therefore, filling up different forms of same domain like airlines, books, automobiles etc; to get the same information is not scalable so it is required to integrate these forms into a unified form. Our proposed work will not only save the user time but also extract quality content that is hidden within the web databases. It requires two modules, first, interface extraction and second, data extraction. Interface Extraction Module performs three tasks:

- It obtains attributes from each of the query interfaces.
- Computes semantic mapping between the attributes of different interfaces.
- Creates a unified query interface

Data Extraction Module extracts the required information from the result pages by analyzing the source code of the web page. The HTML source code contains various tags and by analyzing the structure of the tags, the required information is retrieved.

The process of integration will be as follows:

1. Interface extraction module extracts interfaces from different sources of same domain and a unified interface is designed.
2. Query is send through the unified query interface (UQI) to all the sources from which interface is designed.
3. Result pages are extracted from the databases of all the selected sources.
4. Data is extracted from the result pages.
5. Duplicate results are eliminated.
6. Data is stored in the database of UQI.
7. Finally, hidden web content is retrieved.

5. CONCLUSION

Search engines use different types of crawlers and don't always index the same pages or Websites and no single engine indexes the entire Web. What is invisible to one search engine might be indexed by another.
Search forms have a major role in uncovering quality content that is stored in the web databases. Therefore there is a need to understand these forms in depth.

6. REFERENCES

[1] Mária Princza, Katalin E. Rutkovszky , Content Discovery of Invisible Web, 6th International Conference on Applied Informatics Eger, Hungary, January 27–31, 2004.

[2] Zhen Zhang, Bin He, Kevin Chen Chuan Chang, Understanding Web Query Interfaces: Best Effort Parsing, SIGMOD 2004 June 1318, 2004, Paris, France.

[3] Lalitha K. Sami, Rabia Iffat, Understanding the Deep Web, Library Philosophy and Practice 2010 ISSN 1522-0222

[4] Sriram Raghavan Hector Garcia-Molina, Crawling the Hidden Web, Proceedings of the 27th VLDB Conference, Roma, Italy, 2001

[5] Christopher Olston and Marc Najork, Web Crawling, Foundations and TrendsR_ in Information Retrieval Vol. 4, No. 3 (2010) 175–246_c 2010 C. Olston and M. Najork DOI: 10.1561/1500000017

[6] Ritu Khare Yuan An Il-Yeol Song, Understanding Deep Web Search Interfaces: A Survey, SIGMOD Record, March 2010 (Vol. 39, No. 1)

[7] L. Gordon-Murnane, The Invisible Web: what Search Engines can't Find and Why Libraries Digital Dateline Series, Nov. 2003

[8] S. Lawrence and C. L. Giles, Searching the World Wide Web. Science, 280(5360):98, 1998.

[9] Bergman, M., K., The deep web: Surfacing hidden value, White Paper. 2001