# AN EFFICIENT DATA EXTRACTION FROM THE HIDDEN WEB DATABASES

*J.Ramyabharathi [1], G.Anitha[2]*

**ABSTRACT**

The vast amount of information on the web is stored in backend databases called Hidden Web Databases. The hidden web databases hold the high quality information and have a wide coverage. Nowadays web databases are present everywhere. The data from these web databases cannot be accessed by Search engines and web crawlers directly. Through the query interfaces and filling up number of HTML forms for a specific domain, the hidden databases could be accessed. So many researches have been carried out in this area and it has a highest importance for the recovery and integration of data in the hidden web with a view to provide quality of information to the user. In this paper, an approach is proposed to identify the web page templates and the tag structures to extract the data from hidden web sources as the outcome of a user query.

***Keywords:*** Hidden web, Deep web, Global interface, Hidden web crawlers, Surface web, attribute, search engines.

## I. INTRODUCTION

The data in the hidden web are hidden behind search forms and those are stored in structured or unstructured databases [4]. It is different from the surface web in terms of quality and quantity. The content quality of the surface Web is 1,000 to 2,000 times lesser than that of the deep web. Comparing with the surface Web, the hidden web contains 550 billion individual documents and approximately 7,500 terabytes of data, which is reported about 167 terabytes [4]. Even though, the hidden web is the largest source for structured data, it is not publically indexed and retrieving and processing the content of the hidden web is a big task especially for the dynamically created web pages.

An inverted index as a data structure to index and retrieve the data in the web by the conventional search engines. However, evolving the Hidden Web is more complicated work by several aspects.

- The hidden web index structure deals among the structured data with the large quantity of data.
- The query interfaces enclose more number of features and needs, their relevant values to be submitted.

The following steps to be followed by the user in order to find information hidden in the search interface.

- User has to discover the hidden websites' URLs.
- Visit the websites' home pages
- Queries could be sent through HTML forms.

Assistant Professor, Dept. of CS, CA & IT, Karpagam Academy of Higher Education, Coimbatore.
E-mail : jramyamca18@gmail.com

Assistant Professor, Dept. of CS, CA & IT, Karpagam Academy of Higher Education, Coimbatore.
E-mail : florenceanitha7@gmail.com

" Retrieve the necessary relevant data from the resultant web pages.

" Evaluate and combine the outcome from various web sources.

Hence, there is some necessity for some innovative information services which assist web users to locate the required information. To reduce the effort of the user, this paper discovered an approach to deal with the problems of automatic interaction with hidden web data sources.

## 2. RELATED WORK

This section discusses the works which are strongly interrelated to the proposed system. Even though, there are many proposed hidden web crawling techniques, simple work has to be done on the indexing of information in the hidden web.

Anuradha, and A.K.Sharma [2] find the search interfaces for a specific domain automatically by finding the keyword (a word depicts the domain) in the URLs, the heading and the source code attribute. The web pages are categorized into different types in feature space model using domain ontology. Most of the query interfaces of the identical domain are then combined into the ISI (Integrated Search Interface). This Integrated Search Interface allows the web users with consistent access of from several sources of a particular domain.

Jian Qiu [3] also describes an index structure for getting the content from hidden web. But this research is based on only single attribute. Clustering of data is a best technique to compress the index. But this approach is not suitable for multi-attribute data.

In [5], Raghavan and Garcia-Molina depict a model for retrieving data from hidden web. This research mainly focuses to know about the hidden query interfaces, but not for generating the queries automatically. This approach requires human input so it is not an automatic one. This study only deals with the crawling problem.

Hidden Seek [6] presents the searching technique for single-attribute data. This approach utilizes an inverted index as a data structure to index and retrieve the data in the web by the conventional search engines. To obtaining the results by checking that the keyword is appearing in the URL of a webpage. Here the hidden seeks uses a keyword as a factor.

Siphon++ [7] proposed an approach for the automatic retrieval data hidden at the back of keyword based interfaces. It uses the query creation and selection techniques by identifying the features of the index. Keyword based interfaces are very easy to query than the multi-attribute forms. Because they do not need to elaborates knowledge for the structure of data. Thus this is simple to generate an efficient result automatically to crawl the interfaces.

L. Gravano [8] presents a problem that the several pages that cannot be indexed by search engines and these pages are not visible to other users those who are surfing even though the pages contain the required information. Dynamic web pages i.e. the resultant web pages of and executed query. Subsequently, these pages do not have a static URL and that is not possible for search engines to find, while the queries submitted by the users cannot be replicate by the crawlers. In general, search engines follow links to the index page and then crawl from the index page to other pages on a web site. So it is difficult for the search engine crawlers for visiting a hidden web page.

GIOSS system [8] provides support for the discovery of data sources for the text documents. This system creates the servers for boolean text databases. These servers provide good ideas for searching relevant databases for a given query based on the compact and collected statistics.

In the Niagara system [9], a superset of data sources related to a particular query is found by applying an index structure. But these systems are not supporting the structured queries like queries with multi attributes.

## 3. PROPOSED WORK :

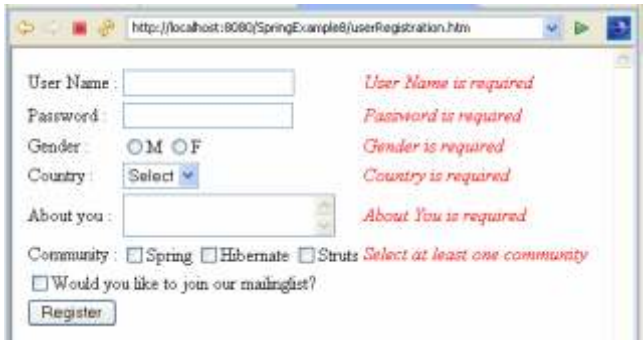
The primary objective of this study is to retrieve the data from several hidden web databases. The data is in the integrated form with no replication records and these data will be stored in large repository.

Search Query Interface is examined as an entry to the websites. The user needs to submit the proper queries to these interfaces for finding and getting the required information from the databases. The produced SQL queries are used to retrieve data from hidden web sources and the desired results could be sending back to the user. There are four phases in our proposed approach. First, Attributes are selected for submission by analyzing different query interfaces. Secondly, the selected queries are then submitted to interfaces. In third phase, the data is retrieved by finding the templates and structure of the html tags. Finally in the last phase, the retrieved data is integrated into one warehouse and all the replicated records are eliminated. There can be several methods are available for submitting queries.

### 3.1 Different types of Query Methods :

**Blank query :**

None of the field is chosen at the time of submitting the query to the query interface is known as blank query. This retrieves the whole database at just once. In this case, we can leave all the fields blank and press the submit button in the form. But this kind of input is not accepted by most of the sites. Many of the sites contain restrictions like "The field is required" or "you must fill all the fields". Some fields are mandatory that must be filled in this case. The following figure shows the type of restriction.


**Query with all combinations :**

This kind of query method used for selecting the values of all the fields. The query (make= " honda", model= "wift", city = " mumbai") which is chosen and then submit this query for the retrieval of information in case of car domain. This type of input provides very accurate

result. But there can be millions of such combinations and needs to be done all combinations prior to submission. Because the study is dealing with multiple attributes. Query interfaces and each attribute may have a huge amount of values. Therefore, this is a tedious process.

**Query with mandatory field :**

Selection of mandatory field is the third query selection method. It is examined that most of the sites have one mandatory field that must be filled and submitted. So it provides the entire database as a result and that database is used for further searches. To uphold the consistency, identical field is chosen from all the local interfaces. The chosen field must be mandatory field in the majority websites.

**3.2 Data Extraction and Integration Architecture :**

The Integrated Search Interface allows the web users with consistent access of the information from several sources of a particular domain. Because some websites have may restricts some types of inputs, like it is not possible to perform blank form submission. So the crawler fills the values of mandatory fields and submits to retrieve the results. The mandatory field is chosen and the values for all options are provided to the global interface. Then submitting these values local interfaces and finally the results would be extracted. The result of each local site must be transferred into the local database. In order to make a database global, the large depository would be created to obtain the data from the local databases.

In HTML, <table> </table> tag is used to define tables. A table can have any number of rows ( table row is represented by the <tr> tag), and each row is separated into several data cells (table data is represented by the <td> tag), that contains the cell data. The backend databases store data in table format. By using these tags the data could be extracted from the backend databases in the table format

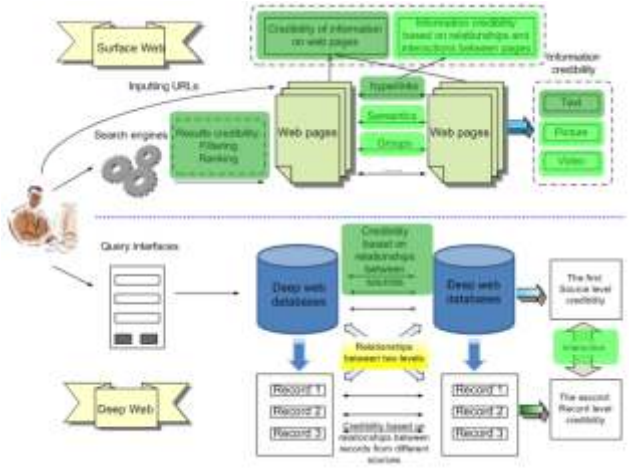



Fig2: Data Extraction and Integration Architecture

**3.2.1 Removing Duplicate Records :**

Submit the same query to the relevant query interfaces for retrieving the desired data from the local databases. On the other hand most of the websites may contain same results or records. Hence, the data warehouse can be created in a manner that the replicated records are eliminated while merging. SQL query is executed and the records are inserted into data warehouse to eliminate records.

**SQL query :**

insert into table1 select * from table2 union select * from table3;

**3.2.2 Formation of Query :**

The structure is prepared for the creation of query. When, the user enters the keyword for searching the search engine replies with the global search interface

for a particular domain which contains attributes. The user would fill the values for attributes.

If the user partially filled the values, then it will be filled with all the acceptable values. Query generator automatically constructs the SQL query using the attribute-value pairs in global interface. The query is triggered on data repository as follows.

Select * from table3

Where a1='y1' and a2='y2' and a3='y3';

Where a1, a2, a3 are the attributes and y1, y2, y3 are the values filled in the form.

## 4. CONCLUSION

This paper proposed a new information retrieval system that assists the users to locate the required information in an integrated form. The difficulty in automatic interaction with hidden web sources is discovered to reduce the user effort. This system consists of Interface analysis, query selection, query submission, result extraction. Repository of information is made with the extracted information from different websites. Not only the duplicate record elimination and the repository but also created the user to research the required information. In future, while the user has to fill the form for finding and then the query is submitted to the repository to retrieve the required information. Thus the effort of the user is reduced by just filling a single form.

## 5. REFERENCES

[1] Anuradha, A. K. Sharma, Komal Kumar Bhatia: "Optimized Merging of Query Interfaces for Domain-specific Hidden Web "Proc. Third International Conference on Advanced Computing and communication Technologies (ICACCT 2008) Volume 2, No. 2, pp. 196-199

[2] Anuradha, A.K.Sharma, "A Novel Approach for Automatic Detection and Unification of Web Search Query Interfaces using Domain Ontology" selected in International Journal of Information Technology and knowledge management(IJITKM), August 2009.

[3] Jian Qiu, Feng Shao, Misha Zatsman, Jayavel Shanmugasundaram, Index Structures for Querying the Deep Web, Workshop on the Web and Databases (WebDB), 2003, 79-86

[4] BrightPlanet Corp. "The deep web: surfacing hidden value."

[5] S. Raghavan and H. Garcia-Molina. Crawling the Hidden Web. In Proceedings of VLDB, pages 129-138, 2001.

[6] Ntoulas, A., Zerfos, P., Cho, J. Downloading Textual Hidden Web Content Through Keyword Queries. In Proceedings of the 5th ACM/IEEE Joint Conference on Digital Libraries (JCDL05). 2005.

[7] L. Barbosa and J. Freire. Siphoning hidden-web data through keyword-based interfaces. In SBBD, 2004.

[8] L. Gravano, H. Garcia-Molina, A. Tomasic, "GIOSS: Text-Source Discovery over Internet", TODS 24(2), 1999.

[9] J. Naughton et al, "The Niagara Internet Query System", IEEE Data Eng. Bulletin, 24(2), 2001.

[10] Brin, Sergey and Page Lawrence. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, April 1998.