Query Intensive Interface Information Extraction Protocol for Deep Web

Dilip Kumar Sharma
Reader- Dept. of Computer Science & Engineering
G L A Institute of Technology and Management,
Mathura, India
todilipsharma@rediffmail.com

A. K. Sharma
Professor & Head-Dept. of Computer Engineering
YMCA Institute of Engineering,
Faridabad, India
ashokkale2@rediffmail.com

Abstract—A new Query Intensive Interface Information Extraction Protocol (QIIIEP) for deep web retrieval process is proposed. Auto query word extraction and auto form unification procedure are newly proposed in order to comprehend various functions of the proposed protocol. Proposed protocol offers great advantages in deep web crawling without over burdening the requesting server. However, conventional deep web crawling procedures result in heavy communication processing loads and procedural complexity for applying either schema matching or improper ontology based query. This makes it difficult to crawl entire contents of deep web. In the proposed protocol, the trade-off between correct query response and communication loads is solved by generating knowledge base at QIIIEP server. Therefore, the proposed protocol can realize flexible and highly efficient data extraction mechanism after deploying QIIIEP server on deep web domain. It enables not only the one stop information retrieval process but also provides auto authentication mechanism for supplied domain.

Keywords-Deep web, HTTP, QIIIEP, query interface analysis, values allotment, response analysis and navigation, relevance ranking.

I. INTRODUCTION

A large part of the Web is "hidden" behind search forms and is indexed only by typing a set of keywords, or queries, to the forms. These pages are known as the Hidden Web or the Deep Web as search engines generally cannot index the deep web pages and do not show them in the results. Searching the deep web is difficult process because each source searched has a unique method of access. Hidden web crawlers must also provide input in the form of search queries. This raises the issue of how best to equip crawlers with the necessary input values for use in constructing search queries. To overcome these issues an open framework based protocol for deep web retrieval process is proposed for simultaneous searches. It supports the current trends in the field of deep web information retrieval process which consist of four steps i.e. Query interface analysis, values allotment, response analysis & navigation and relevance ranking. Proposed protocol will reduce complexity in these activities except relevance ranking of deep web data querying.

II. RELATED WORKS

Pages for search interfaces are commonly HTML forms which is filled and submitted by users and server respond accordingly. But every form is not search interface. Search form can be identified by using one of the simplest method i.e, heuristic rules [1],[2]. Other approaches to detect search interface are decision trees based classification models to detect search interface and random forest algorithm where classification is made by aggregating predictions of individual decision trees in the forest in which each classifier is realized from a subset of the feature space. The aggregated approach can fully exploit the useful features in search forms [3].

The QIIIEP (Query Intensive Interface Information Extraction Protocol) will eliminate this step by self guiding to the crawler about the search interface.

After detection of hidden web search interface, the next task is to identify accurate matching for finding semantic correspondences between elements of two schemas. Many automatic or semi-automatic matching systems meticulous in a simple 1:1 matching, such as Cupid method [4], OMA method[5], GLUE and LSD method[6][7] and Similarity Flooding method[8], for schema extraction are considered for the non-hierarchical structure of query interface, which neglects the grouping and hierarchical relationships of attributes. So the semantics of a query interface cannot be captured correctly. Based on the nonhierarchical model, literatures [9],[10] proposed a hierarchical model and schema extraction approach which can group the attributes and improve the performance of schema extraction of query interface. But they show the poor clustering capability of pre-clustering algorithm due to the simple grouping patterns and schema extraction algorithm and require human interaction and not suitable for dynamic large scale data sets. Other approaches are DCM [11] and MGS framework [12] which pursues a correlation mining approach by exploiting the co-occurrence patterns of attributes, and proposes a new correlation measure while other hypothesizes that every application field has a hidden generative model and can be viewed as instances generated from models with possible behaviors [13].
New schema extraction algorithm Extr[14] which is based on the pre-clustering of attributes P by using MPreCluster, Komal Kumar Bhatia et al [15] presented in his research literature that mapping can be done by using domain specific interface.

The QIIIEP (Query Intensive Interface Information Extraction Protocol) will reduce this complexity by using pre-information about the form and its elements from QIIIEP server. The knowledge base is generated by auto query word extractor or it is provided by site administrator.

Ontology is a formal specification of a shared conceptualization [16]. This step is required for analyzing area or specialization of web page so that in further steps appropriate data set will be efficiently placed in query part of the page. Deitel et al. [17] present an approach for learning ontology from RDF annotations of Web resources. Stojanovic [18] presents an approach for an automated migration of data-intensive web sites into the semantic web. The paper [19] presents an approach TANGO (Table Analysis for Generating Ontologies) to generating ontologies based on HTML table analysis. Zhiming Cui et al published his research [20] which makes Mini-Ontology from Query Interfaces by applying employs vision-based approach.

Ontology identification is not required in proposed protocol because QIIIEP server's pre knowledge about the form and its elements will be enough for choosing correct value for each element of form.

Integration of the databases with the query interfaces is further step in this process. The search form interface brings the attributes together and this step will analyze appropriate data values by their structural characteristics of the interface and the order of attributes in the area as possible as it can. For integrating interfaces, the core part is dynamic query translator, which can translate the users' query into different forms [21] [22]. Mapping is done by Fuzzy comprehensive evaluation methods [23] which map the attribute of the form to the data values.

Query Translation Technique is used to get query from different deep web sources i.e. to translate queries to sources without primary knowledge. Some methods can be concerned such as type-based search-driven translation framework by leveraging the "regularities" across the implicit data types of query constraints. In [24] they found that query constraints of different concepts often share similar patterns, and encoded more generic translation knowledge for each data type. Type-based predicate mapping method [25] proposed by Z.Zhang focusing on text type attribute with some constraint.

The QIIIEP (Query Intensive Interface Information Extraction Protocol) will reduce this complexity by using QIIIEP server's query words database which is generated by auto query word extractor or it is provided by site administrator.

III. PROPOSED WORK

The QIIIEP (Query Intensive Interface Information Extraction Protocol) is an application-level protocol for semantic otology based query word composition, identification and retrieval systems. It is based on request/response semantics. This specification defines the protocol referred to as QIIIEP 1.0. It will work on port 55555 on http server and generate response encoded by using XML.

The initial specification of this protocol can be discussed on the basis of figure1 given below.

1. In first step, crawler will request for any web server to fetch a page.
2. In second step, crawler will analyses the form to identify search interface. Search interface must include rel tag to describe the QIIIEP server address and form id.
3. In this step, crawler will send the request to QIIIEP server for getting the semantic query word list which is defined by the site administrator or QIIIEP auto query word extractor to correlate the form fields.
4. In this step, QIIIEP server will reply to the crawler about each entry of that form.
5. In this step, crawler will send the filled form by placing received query words to the HTTP server.
6. In this step, crawler will crawl the contents generated by that query word.

a. In this step, QIIIEP auto query word extraction module continuously watch the form interface to extract query word supplied by user as well as from the content generated by processed query.
b. Finally, it will store the query word into the QIIIEP database for further analysis.
c. It merges the form ids to the forms at the time of form generation.
d. Fetch the form id to query word table relationship from QIIIEP server.

A. HTTP Server
The http server is regular http server on which the web site is deployed.

B. Auto form id generation module
This module will help to implement QIIIEP protocol on current architecture of web site. It will parse every form of that web site and merge the form id with QIIIEP server query word list so that at the time of crawler request the crawler will properly identify the search interface for sending the request of keywords to the QIIIEP server.

C. Auto query word extraction module
This module will extract the query words supplied by users of that site so that QIIIEP server can extend the query word list by exploiting advantage of human curiosity to find relevant information on that domain.
D. Query word ranking module
This module is responsible for providing the best match query word assignment to the form filling process and reduce the over loading by less relevant querying to the domain.

E. User Authentication to domain mapper
This module of crawler is responsible for mapping the login credentials provided by users to the crawler with the information provider domain. The main benefit of using this mapper is to overcome the hindrance of information retrieval in between result link and information. The crawler will use the mapping information to allow the specific person to receive information contents directly from the domain by automatic login procedure and reduce a step of separate login for user.

F. Agent for authenticated crawling
This module work for those sites where the information hidden by the authentication form. It stores authentications credentials of every domain in its knowledge base situated at crawler, which is provided by the domain administrator. At the time of crawling it will automatically authenticate it self on the domain for crawling the hidden contents. These contents will be used only to indexing keywords and will not be available by search service as catch because of privacy issue.

G. Query word to URL relation manager
This module will store each and every query word associated with specific element of form by creating reference of domain path, so that at the time of link generation in response to search, query word can be mapped to provide the contents by sending query word in post request to the domain.

IV. IMPLEMENTATION ISSUES
This protocol is still in it’s initial specification stage but proposed framework’s mechanism is not only very simple to implement but also imposed lesser amendments in existing infrastructure because all the functionality is implemented over http server so there is no need to resolve any major issue before experimenting the propose framework. The crawler will send the request for keywords in xml encoded document which is having information of url and form-id QIIIEP sever will also respond this request in xml encoded document which is transferred by secure network. Further it will use guidelines inherited from OAI -PMH for achieving request- response XML document formulation. The QIIIEP server can be deployed on to the same server or it can be deployed on separate server.

V. COMPARISON WITH OTHER PROTOCOLS
Some of other supporting mechanisms are also proposed to deep web retrieval processes in which frameworks are designed for extraction of information. Search/Retrieval via URL (SRU) protocol is a standard XML-focused search protocol for Internet search queries that uses Contextual Query Language (CQL) for representing queries. The SRU uses the REST protocol and introduces sophisticated technique for querying databases by simply submitting URL-based queries. For example
URL?version=1.1&operation=retrieve&query=dilip&maxRecords=15

Proposed protocol is deployed on the querying server as well as on crawler so there is no need to make request from get method only .The crawler will find the contents by using post method of HTTP protocol.

The Open Archives Initiative (OAI) [26] Protocol for Metadata Harvesting (OAI-PMH) provides an interoperability framework based on the harvesting or retrieval of metadata from any number of widely distributed databases. Through the services of the OAI-PMH, the disparate databases are linked by a centralized index. The data provider agrees to have metadata harvested by the service provider. The metadata is then indexed by the harvesting service provider and linked via pointers to the actual data at the data provider address.

Proposed protocol is simple to implement and controlled by site administrator so the main advantage is that the contents provided to crawler is controlled by authorized entity. A separate repository for meta data is not required in this implementation.
VI. RESULTS
We implemented initial version of this protocol specification on www.qiiiep.org and analyzed the results. We used three different test domains to judge the precision of retrieved contents.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Query Forms</th>
<th>Correct identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Book</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Job</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

As shown above the graph is plotted in between received query words and successful content extraction at different domain, and concluding that contents extraction are close to query words received at a satisfactory level for all three domains.

TABLE II

<table>
<thead>
<tr>
<th>Domain</th>
<th>Form ids</th>
<th>Query words received</th>
<th>Successful contents extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
<td>37723d89ce0b0518</td>
<td>38</td>
<td>36</td>
</tr>
<tr>
<td>Book</td>
<td>9c563a9b717356ce02</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>Job</td>
<td>61fa2718c890a61b</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

ACKNOWLEDGMENT
All the companies/products/services names are used for identification purposes and may be the trademarks of their respective owners.

REFERENCES


Search Engine: A Backbone for Information Extraction in ICT Scenario

Dilip Kumar Sharma
Shobhit University, Meerut, UP, India
todilipsharma@rediffmail.com

A. K. Sharma
YMCA University of Science and Technology, Faridabad, Haryana, India
ashokkale2@rediffmail.com

ABSTRACT

ICT plays a vital role in human development through information extraction from a variety of ways. ICT includes a wide area from computer network to telecommunication network. One of the important module of ICT is computer network which is the backbone of World Wide Web (www). Search engine helps in information extraction from WWW. A Search engine is a computer program that browses and extracts information from the WWW in systematic and automatic manner. This paper focuses on three main components of search engine: Extractor, a web crawler which starts with a URL, Analyzer, an indexer that processes words on the web page and store the resulting index in a database, and Interface Generator, a query handler that understands the need and preferences of the user. This paper not only concentrate on the information available on surface web that is available through general web pages but also on the hidden information that is behind the query interface called as deep web. This paper highlights the purpose of optimizing the search results as intended by the user from the network to serve billions of users worldwide with an immense pool of information available without having to remember its location or URL. Further this paper emphasizes on the Extraction of relevant information to generate the preferred content for the user so that the user gets the needed information at the very first result of his search query. The deep web information can play a significant role in research and development. This paper also discusses the aspect of deep web with analysis of few existing deep web search engines.

Keywords Search engine, web crawler, deep web, analyzer, extractor, interface generator, ICT, WWW.

I INTRODUCTION

Information and communication technology have tremendous potential for social impact, human development and improving the lives of people they serve. Through ICT peoples are able to communicate in better way and can access relevant information. It also helps in developing collaborative and research skills. People can gain confidence and avail opportunities on their potential. Information and communication technology provides appropriate hardware, software and networking services to the search engine. To find out relevant pages instantaneously from billions of web pages available on the internet is a complex task. So, information extraction in web scenario is must to provide the relevant search to the user at the very first instant. An effective search engine is the necessity of today’s information era. Search engine is a software program that searches for web sites that exist on the World Wide Web. Search engines search through its personal databases of information in order to provide the relevant information. A web crawler is an automated program that starts with a set of URLs called seeds and stores all the URL links associated with downloaded web page in a table called crawl frontier. The extractor sends all these information attached to the textual raw data to the analyzer. The analyzer then takes the entire HTML code of the downloaded web page and analyzes the code, keeping the relevant data and rejecting the rest. Some composing techniques are applied to link containing the similar types of information from the database to generate the relevant query results.

Information and communication technology can be related to information extraction in web context or in search engine in a variety of ways (Anderson, J., Weert, T.V. et al.,2002)( Kundu, A., and Sarangi, N.,2004). Traditional web crawling techniques have been used to search the contents of the web that is reachable through the hyperlinks but they ignore the deep web contents which are hidden because there is no link is available for referring these deep web contents. The web contents which are accessible through hyperlinks are termed as surface web while the hidden
contents hidden behind the html forms are termed as deep web. Deep web sources store their contents in searchable databases that produce results dynamically only in response to a direct request (Bergman, 2001). Figure 1 shows the benefits of information extraction using ICT in human development in context of search engine.

![Figure 1: Benefits of information extraction using ICT in human development in context of search engine](image)

II ANALYSIS OF APPLICATION AREA OF ICT

Some of the area in which ICT plays a significant role in their development is analyzed below.

**ICT in Education**

In 1999 an analysis was done to find out the use of computer in schools. In that analysis it was found that a large number of students were sound enough to use the computers without taking help from school. The analysis also reveals that male and female students have different area of interest regarding the use of computer. A complete frame work can be divided into five modules.

**Resource:** It corresponds to a range of sources to access information.

**Tutorial:** It helps to acquire new knowledge along with feedback.

**Exploration and Control:** It investigates and provides the situations.

**Support:** It facilitates in communicating and providing the information to users.

**Link:** It facilities the interactive information exchange between individuals and groups.

Analysis of ICT evolution reveals that four specific approaches should be applied to adoption and use of ICT in educational organization. These four approaches are evolvement, application, hybridization and transformation (History education and information communication technologies, 2010) (Anderson, J., Weert, T.V. et al., 2002).

The deep web provides for a wide range of educational resources which varies from a student searching for an ideal school based on key personal requirements to an administrator looking for fund-raising resources. The key resources include directories and locators, general education resources, statistics resources etc.

**ICT in Business**

ICT is also useful in business environment. It underpins the achievement of current business and it offers government with a proficient communications. At the same time, ICT adds value to the processes of learning in the organization and management of learning institutions. The Internet is a driving force for large development and innovation in both developed and developing countries. The following competencies are gaining importance with reference to ICT:

- Decision-making,
- Expert advice
- Control on dynamic situations,
- Collaborative working
- Seamless communicating

Technological developments lead to changes in work and changes in the organization of work, and required competencies are therefore changing (Kundu, A., and Sarangi, N., 2004).
**ICT in Human Resource Development**

ICT can be applied in rationalization and transformation of human resource development. ICT facilitates managers and employees to have direct accessibility to resources. Human resource development with ICT is termed as eHRDM. The public service commission (PSC) is an autonomous body for recruitment of human resource for government jobs. The Various steps in recruitment and selection process of PSC are to receive demands of human resource from government offices, verification of the given information, advertisement through media, receiving and sorting the applications, screening of applications, conduct pre-selection process, conduct interview process and final appointment.

Over the year’s, whole recruitment and selection process of PSC is in a typical paper based system. Recently, PSC has introduced the recruitment and selection database system which invites online job applications for the advertized jobs. The recruitment and selection (R&S) process developed by ICT personnel has objectives to find out the duplicity and redundancy in the R&S system, develop and maintain the system at par with the organization, which has successfully employed the ICT based R&S system, documentation of benchmark standard with recommendation, establishment of system implementation committee with technical personnel, implementation of the system, to conduct the workshops related to the system organization, formation of rollout process and establishment of monitoring and evaluation system.

Guidelines required for successful establishment of ICT based system for R&S are to establish LAN, MAN and WAN, procurement of leased line, increase the band width of system to facilitate the efficient data transmission. The requirements for system hardware and software for ICT based R&S system are to develop the system database server, backup server, windows based server and SQL based server, development of antivirus and firewall software, endorsement of physical access and control, connecting PCs website through World Wide Web, 24/7 browsing of advertisement by easy submission of online application and tracking of the job application (Wachira, F. N.,2010).

**ICT in Social Issues**

Planners, policy makers and researchers hold highly polarized and equivocal views on the diffusion of Information Communication and Technology (ICT). The role of search engine is significant in promoting objectives such as poverty alleviation, universal education, reduction in mortality and health hazards, sustainable development and bridging the socio-economic divides in the world. It leads us with many online social work search that are providing number of free services to the social work and related professions.

**ICT in Job information**

ICT can be used in job information extraction. It can therefore enhance human development. User can utilize a search engine for extracting job related data hidden behind the search forms and can identify the job according to their requirement. This makes the job searching easier and thereby increasing the number of online job seeker and net users. Therefore, it is beneficial for both common people and Internet service providers. It helps us to provide the different and desired web pages related desired year of job and attaining the job according to their demand.

**ICT in Shopping and auctioning**

ICT can also be useful in terms of web context in online shopping and online bidding. There are various shopping and auction sites such as amazon.com, ebay.com etc. that utilize the concept of ICT to support human development. User is required to enter the prerequisites on the website form and submit it online. The search engine then produces the list of items related to the user’s query. It helps us in determining the current market trends by leading us to give the idea about the price raised and price fall for the smooth flow of business.

**ICT in Database related information extraction**

Information from the database can also be extracted with the help of Web crawler that uses ICT. Tremendous amount of data remain hidden behind the database which can be explored using various programs which illustrates the use of ICT in database information extraction.

**ICT in E Commerce/Banking**

ICT plays a vital role in e commerce as well as e banking that has a better transparent system in which users can trade efficiently and can participate globally. Security is prime concern in this matter and requires further improvement. It helps to provide many E commerce development solutions to give the company many supports that are needed to run the day to day business (Kumar, M., and Sareen , M.,2009).

**ICT and Environment**

The impact of ICT on environment has made tremendous changes like paperless offices and global society for environmental protection. ICT companies are working more upon green technologies and promoting biodiversity
and preservation to reduce the impact of their own activities. It helps in optimizing many of the environmental health safety jobs and to use this employment by various workers.

In spite of the high-quality and authoritative information it provides, Deep web offers some excellent resources focusing on entertainment that are as useful as its serious counterparts. These entertainment resources such as movies, music, amusements, live performances, and other activities people do mostly for fun and pleasure.

<table>
<thead>
<tr>
<th>Content Types</th>
<th>Topic</th>
<th>General Search</th>
<th>Messaging/Chatting</th>
<th>Job Search</th>
<th>Calculators</th>
<th>White/Yellow Pages</th>
<th>Library</th>
<th>Portals</th>
<th>Classified</th>
<th>Shopping</th>
<th>Publications</th>
<th>Internal Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage</td>
<td>54%</td>
<td>15%</td>
<td>14%</td>
<td>2%</td>
<td>2%</td>
<td>2%</td>
<td>2%</td>
<td>5%</td>
<td>5%</td>
<td>11%</td>
<td>13%</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2:** Distribution of web sites on content types (Bergman, 2001)

**ICT in Health**

ICT has increased the access to information and has therefore increased the effectiveness of health care services by promoting the expansion of health and social services. It provides many links related to the health field showing numerous numbers of diseases and their related researches as well as their therapies concerned to it. The deep web avails a vast amount of authoritative information, offered by reputed health care organizations. Unlike the surface web resources that can even mislead the users, the deep web resources promises to provide the exceptionally high quality information on diseases, medical procedures, pharmaceutical drugs, nutrition, clinical trials, or other healthcare related issues.

**ICT in E Government**

The use of ICT in governance is increasing to deliver its services to the citizens at the location of their connivance in an efficient and transparent manner. Electronic governance is the application of ICT. Through the ICT, government can exchange information and services, communicate transactions between Government and citizen (G2C), Government and business (G2B). Therefore, being a service provider, Government should motivate their employees for delivering services through ICT (Sharma, D.K., Varshneya, G. et al., 2007). Governments today are putting more and more information on the every day. Most of the portals to government information are covered by the surface web materials. The outstanding government Deep Web resources provide for sites that offer “general” type of information, directly from government entities themselves. These resources are useful for searcher who relies solely on the general purpose search engine.

**ICT in Other fields**

ICT can be used in online messaging and chatting. In general, web portals, classifieds, publications etc. use ICT. All of these areas use the concept of Information extraction based on form values. The Deep Web delivers many resources that meet all the important criteria required by the people who conduct legal research on the web by providing the correct, authoritative, and easy to access material in a timely manner. Real-time information is probably the “purest” type of Deep Web data, and it’s not likely that general-purpose search engines will ever in their indices. Real-time information is almost always stored in databases that are constantly updated in real or near-real time. In some cases, such as stock quotes or airline flight arrival information, each update obliterates the previous data record. Even if a search engine could somehow crawl and index this information, it would be like isolating a single frame from a feature length movie. In other cases, real-time data is preserved, but the key point is that it is archived data in raw form, which a searcher cannot easily manipulate.

Figure 2 shows the distribution of websites based on content type. In order to utilize the full potential of web, there is a need to concentrate on web content so that it can prove to be a great source of information i.e., information extraction should be done in the context of web which leads to utilization of Information Communication and Technology.

Deep web information are useful in education, business environment, human resources development, social issues, job information, shopping, in e-commerce/banking, database related information extraction, health, e-government, and other field such as messaging & chatting because the database can publish the result through direct query. Deep web sites post their result as dynamic web pages in real–time. These dynamic web pages have a distinct uniform resource locator address that permits them to be recovered again later. But in surface web pages, web pages are static and linked to other pages. Static pages do not have a unique URL address and therefore are not allowed to access information again later. Deep web sites also tend to improve the quality of search because it does not provides a long list of hits instead a right list. It means, it provides relevant information for each query. Through Deep web sites users can choose authoritative sites, but it is advisory to be careful about the selection of searchable sites. Users can make their own determination about the quality.
Search engine is a tool to gain information to the search of a specific collection; this may be a library, the internet, or have a personal collection. Search engine plays a vital role in extraction of information from World Wide Web. A search engine consists of web crawler whose function is to download web pages from the internet and store them into the database. The list of URLs is stored in the database queue, from where a scheduler selects URLs. These URLs are then downloaded by a multi-threaded downloader. Once downloaded, the text and meta data is stored in the database storage.

**Need of search engine**
The size of the World Wide Web is drastically increasing. Therefore, it is necessary to find out the required information in lesser time. Web crawlers are used to take all the links from the visited pages by the search engine and indexing is done in order to arrange them according to their preferences. Crawlers are used to carry out the maintenance by checking the links and HTML code. It is used to test web pages and links for valid syntax and structure. A Search engine should be able to search the information in World Wide Web in different formats from different sources. Search engine combines all the modules required for a particular application. Examples are online discovery, compliance of financial regulatory services, pharmaceutical research, counter measure for terrorisms sells prediction and customer support etc.

**Working of the search engine**
Figure 3 shows the working of a typical search engine. A crawler uses the HTTP network protocol in order to browse the internet which allows it to download or upload data from and to it. The crawler browses this URL and downloads the associated web page. It then looks for hyperlinks in the downloaded page. The URLs attached to these hyperlinks are then added to the queue. First of all, the crawler crawls over the http server through the search engine in which the crawl frontier, contains the links to be crawled. After the Links are extracted, they are parsed for further processing. After indexing, all the links are saved into the Content Database after applying ranking on them. Whenever the user enters the query in the search engine, it first checks the link from the Content Database and shows the corresponding results to the user. After that user does the event on the given link, the corresponding to which relevant pages are shown to the user from the web. A search engine consist of three major modules i.e. extractor, analyzer and indexer and interface generator (Brin, S. and Page, L.,1998). Algorithm of these modules is given below.

**Extractor**
The Extractor is a Page Fetcher which fetches the web pages from the internet. A link extractor takess out a web address from a server reply to play it back as well as the dynamic parameters of the web address (Craven T. C., 2003) (Kaplan, D., Iida, R., and Tokunaga, T.,2009). The algorithm of the extractor is as follows:
- Extract the URL from the crawl frontier table which have not been parsed till now.
- Send HTTP GET request for that particular URL to the server.
- Download the URL’s related page for further parsing procedure.
- Call the analyzer module to parse this URL.
- Repeat the steps from step1 until all the URL of crawl frontier are been parsed.
Analyzer and indexer

Analyzer is an indexer that processes words of the web page and stores the resulting index in a database. The algorithm of the analyzer is as follows:

- Receive the downloaded URL’s HTML code string.
- Check for internal links, and if present, convert them into external links.
- For each external hyperlink present in the HTML code, do:
  - Extract the URL attached to the hyperlink if not already present in the crawl frontier table else skip.
  - If this URL is unwanted (.gif, .jpg, .css, .xml, .doc, .pdf, .mp3 etc), then skip.
  - Else, insert this URL in the crawl frontier table.
- Extract other information attached to the downloaded web page from its HTML string, like title, meta description etc.
- If two of the strings are same then decide their precedence on the basis of HTTPs labelled.
- Save all this information in the data table in the database.
- For each search word in the array, do:
  (a) Search the number of occurrences of the word in the database table data.
  (b) Arrange the search results in decreasing order of number of ‘hits’.
- Stop

Interface generator

Interface generator is a query handler that understands the need and preferences of the user. The algorithm for the composer is as follows:

- The query that is to be searched is been entered.
- The query is been filtered by stemming process by removing white space, special characters, symbols etc from the user’s search query.
- If search query is empty, then return, else continue.
- Break the search query into individual search words and store them into an array.
- Club together the search results of individual search words and again arrange them.
- Display the results on the output screen for the user.
- Stop.

Utility of search engine

These algorithms describe the procedure about how the URL are been parsed and fetched from the web and how actually the crawler works by getting collected with all these URL from the web. They also describe the way the parsing should be done after fetching the HTML page related to that corresponding URL. Since the crawler is intended for getting the information depending upon the query fired, so these algorithms also describes that how query is processed and the related information to that of the query is displayed to the user. The main part of search engine is web crawler which is used for collecting and storing the information in database.

Types of Web Crawlers

The following are the general types of web crawlers.

Simple Crawler

Developed by Brian Pinkerton in 1994, is a, single-process information crawler which was initially made for his desktop. Later, it was extended on to the internet. It had a simple structure, and hence had limitations of being slow and having limited efficiency.

Parallel Crawler

Initially given by Junghoo Cho in 2002, this approach relied on parallelizing the crawling process, which was done by multi-threading. It had faster downloading and was less time consuming but required more bandwidth and computational power for parallel processing. It parallelizes the crawling process and uses the multitithreading that reduces the downloading time. For example, Google Bot employs parallel, single-threaded processes (Yadav, D., Sharma, A.K., and Gupta, J.P., 2008).

Focused Web Crawler

This concept was given by Manczer in 1998, refined by Soumen, Chakrabarti et al. (1999). They focused on specific topics and get the relevant result. They are also called as topical crawler. They do not carry out the unnecessary task by downloading all the information from the web instead it download only the relevant one associated with those of the predefined topics and ignores the rest. This is advantageous for time saving factor. They are limited in its field. Examples: Healthline, Codase etc (Chakrabarti, S., Berg, M., and Dom, B., 1999).

Distributed Web Crawler
Distributed web crawling is a distributed computing technique whereby Internet search engines employ many computers to index the Internet via web crawling. It uses distributed computing and reduces the overload on server. It also divides the works into different sub servers. Their main focus is on distributing the computational resources and the bandwidth to the different computers and the networks. Examples: YaCy: P2P web search engine with distributed crawling (Boldi, P., Codenott, B., Santini, M. and Vigna, S., 2004).

**Deep Web Crawler**

ICT provides various tools and one among them is deep web or web wrappers. But to have an access on these tools developer requires deep knowledge as web wrappers are site specific solutions that have dependencies with web structure and also, web a wrapper requires constant maintenance in order to support new changes on the web sites they are accessing. The deep web / hidden web refer to World Wide Web content that is not part of the surface Web, which is not indexed by surface web search engines. In Deep web, Pages are not indexed by search engine. It mainly requires registration and signing up. Example- Yahoo Subscriptions, LexiBot etc (Sharma, D.K., and Sharma,A.K., 2010)( Bergman, M. K., 2001).

**Issue of network interoperability in ICT**

ICT is built upon numerous computers and telecom networks. For efficient communication on a global basis, all networks should be compatible to each other or some type of interface should be provided between networks. The main reasons to cope up with inter platform and architecture compatibility that enables the seamless information extraction. So research should be done in the direction of network cooperation and international standards should be made that can facilitate the cooperation among the diverse platforms (Acevedo, M., 2009).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Advantage</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple crawler</td>
<td>Single process information, iteratively download the web pages and follow the breadth first traversal.</td>
<td>Simple structure, indexing process is straight forward.</td>
<td>Slow and limited efficiency.</td>
</tr>
<tr>
<td>Parallel crawler</td>
<td>Process is parallelized and indexing is done through identifying a keyword to make the search more relevant.</td>
<td>Faster downloading, less time consumption.</td>
<td>More bandwidth, more computational power.</td>
</tr>
<tr>
<td>Focused Web crawler</td>
<td>Downloaded web pages related to predefined topic or domain.</td>
<td>Relevant downloading, provide relevant results, reduces number of retrieving pages thereby regulating the visiting pages and analyses is more deep so as to define high quality pages.</td>
<td>Limited in field. Pre decided resource extraction.</td>
</tr>
<tr>
<td>Distributed Web crawler</td>
<td>Distributed computing technique and interact in peer to peer fashion.</td>
<td>Reduces overload on server, division of work in sub servers.</td>
<td>More computational power. Complex to manage.</td>
</tr>
<tr>
<td>Hidden/Deep crawler</td>
<td>Dynamic generation of web pages.</td>
<td>Access huge amount of online data.</td>
<td>More resource and processing needed</td>
</tr>
</tbody>
</table>

Table 1: Summary of different types of web crawlers
1. Surface web page is static and linked to other pages.

2. They are not narrower with deeper content.

3. Total quality content of the Surface web is less.
4. Surface web is not relevant for every information need and for others domain.
5. The surface web does not publish the result through direct query. As search engines look through links, they are unable to access certain type of web pages. These pages never enter the system and, therefore, are never indexed.

6. Surface web make static HTML pages that are less likely to be from professional content suppliers.
7. The Surface Web, crawled by popular search engines running today, contains only a fraction of the overall unstructured content available on-line today.
8. The surface web is the “general web” and is what one can find using general web search engines. It is also what one see in almost all subject directories.

1. Deep web page is dynamic content served up in real time from a database in response to a direct query.

2. Deep Web sites ought to be narrower with deeper content.
3. Total quality content of the Deep Web is thousand times much greater than Surface web.
4. Deep Web content is highly pertinent to every information need and other domain.
5. The Deep Web search is the content that resides in searchable databases, the results from which can only be discovered by a direct query. Without this direct query, one would not be able to reach the results.
6. Professional contents suppliers typically have the kind of database-based sites that are more secure in deep web.
7. The Deep Web contains all the “unknown or hidden”, unstructured content that the surface web failed to provide to its users.
8. The Deep web is the “hidden web” and is what one cannot find using the normal search engine for this they need the Deep web crawlers to get the information fetched.

Table 2: Surface web versus Deep web

<table>
<thead>
<tr>
<th>Surface Web</th>
<th>Deep Web</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Surface web page is static and linked to other pages.</td>
<td>1. Deep web page is dynamic content served up in real time from a database in response to a direct query.</td>
</tr>
<tr>
<td>2. They are not narrower with deeper content.</td>
<td>2. Deep Web sites ought to be narrower with deeper content.</td>
</tr>
<tr>
<td>3. Total quality content of the Surface web is less.</td>
<td>3. Total quality content of the Deep Web is thousand times much greater than Surface web.</td>
</tr>
<tr>
<td>4. Surface web is not relevant for every information need and for others domain.</td>
<td>4. Deep Web content is highly pertinent to every information need and other domain.</td>
</tr>
<tr>
<td>5. The surface web does not publish the result through direct query. As search engines look through links, they are unable to access certain type of web pages. These pages never enter the system and, therefore, are never indexed.</td>
<td>5. The Deep Web search is the content that resides in searchable databases, the results from which can only be discovered by a direct query. Without this direct query, one would not be able to reach the results.</td>
</tr>
<tr>
<td>6. Surface web make static HTML pages that are less likely to be from professional content suppliers.</td>
<td>6. Professional contents suppliers typically have the kind of database-based sites that are more secure in deep web.</td>
</tr>
<tr>
<td>7. The Surface Web, crawled by popular search engines running today, contains only a fraction of the overall unstructured content available on-line today.</td>
<td>7. The Deep Web contains all the “unknown or hidden”, unstructured content that the surface web failed to provide to its users.</td>
</tr>
<tr>
<td>8. The surface web is the “general web” and is what one can find using general web search engines. It is also what one see in almost all subject directories.</td>
<td>8. The Deep web is the “hidden web” and is what one cannot find using the normal search engine for this they need the Deep web crawlers to get the information fetched.</td>
</tr>
</tbody>
</table>

From the table 2 (Bergman, 2001) it clear that Deep web searching would be useful for development work because it create dynamic content, response to a direct query and gave relevant information for market and all other domain. It more secure for professional contents, Government reports, strategy statements, research/sell reports, operational papers.

IV ANALYSIS OF SOME OF IMPORTANT DEEP WEB SEARCH ENGINES

Complete Planet (www.completeplanet.com)
Complete Planet is an invisible web portal with fast service, relevant results and an easy to use interface. Complete Planet searches over more than 7000 database and search engines. Complete planet’s advanced search is pretty standard. It provides the option to search by title, keyword, description, date etc. Every database is extremely alert in character. While surfing the web, user can click on the links that are provided by the search engine, to reach the individual high value databases. It is easy to use, simple and broadens the search.

IncyWincy (www.incywincy.com)
IncyWincy is an invisible Web search engine and it behaves as a meta-search engine by tapping into other search engines and filtering the results by searching the web, directory, forms, and images. It discovers search engines when spidering the web. It features a unique search engine relevancy algorithm. It provides user listings, premium keyword purchase, and custom website spidering. Information may change quickly and become unavailable or may become the part of the visible web.

Scirus (www.scirus.com)
It contains the latest search engine technology and searches over 410 million- specific web pages that enable the user to quickly pinpoint the scientific, scholarly, technical and medical data on the web. It has a wide range of special features to help to get the scientific information which are needed. It can find specific conference, abstracts and patents. It helps to refine, customize and save the searches. Scirus is a search engine mainly made for science subjects. It concentrates simply on pages containing technical content.

DeepDyve (www.deepdyve.com)
DeepDyve is the largest online rental service for scientific, technical and medical research with over 30 million articles from thousands of authoritative journals. It makes research easy and affordable. It can copy entire sentences, paragraphs and even complete articles against the specific query. It also finds related information for every article by
clicking the “More like this” button on search result page. Their search is not restricted to keywords or literals. It can search by simply pasting the whole of the article into the search bar. Some of the articles in the DeepDyve are “open access” and are marked as “free” for any user to read.

**Biznar (www.biznar.com)**

Biznar is a free, publicly available deep web search engine that uses advanced “federated search technology” to return high quality results against the search query in real time. It accelerates search by returning the most relevant results from over 60 authoritative business collections to one easily navigable page. It is very effective search engine created especially for those professional businesses that need to get access to specific information for their works. This system is a federated search which means that by using this tool one can look the information not from only one source but from many databases at the same time (Price, G., and Sherman, C.,2001) (Saikat Basu,2010).

**V COMPARATIVE ANALYSIS OF DIFFERENT SEARCH ENGINES**

After analyzing the different types of search engines, it is concluded that surface web search engine results in large number of surface web results, whereas deep web search engine extracts the hidden web data. Table 3 and table 4 shows query words versus results counts for surface web search engines and deep web search engines respectively.

<table>
<thead>
<tr>
<th>Query words</th>
<th>Surface Web Search Engine</th>
<th>Deep Web Search Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Google Search</td>
<td>Yahoo Search</td>
</tr>
<tr>
<td>Mobile computing</td>
<td>9,980,000</td>
<td>311,000,000</td>
</tr>
<tr>
<td>Electronic commerce</td>
<td>8,400,000</td>
<td>2,290,000</td>
</tr>
<tr>
<td>Digital signal processing</td>
<td>9,850,000</td>
<td>1,590,000</td>
</tr>
<tr>
<td>Compiler</td>
<td>15,200,000</td>
<td>965,000</td>
</tr>
<tr>
<td>Soft computing</td>
<td>3,110,000</td>
<td>1,510,000</td>
</tr>
<tr>
<td>Medical</td>
<td>465,000,000</td>
<td>57,100,000</td>
</tr>
<tr>
<td>Research</td>
<td>708,000,000</td>
<td>88,800,000</td>
</tr>
<tr>
<td>Numerology</td>
<td>38,20,000</td>
<td>7,190,000</td>
</tr>
</tbody>
</table>

Table 3. Query words versus Results counts for surface web search engines

Table 4. Query words versus results counts for deep web search engines

Figure 4 and figure 5 reflects the graph between the results obtained against query words for the different search engines.
VI CONCLUSION

This paper highlights the role of information extraction in human development. Information extraction is facilitated by ICT. Search engine plays a vital role in information extraction from World Wide Web. One of the important module of search engine is a web crawler. Web crawler can result in desired information extraction from www. A lot of important information is hidden behind the deep web. Normal web crawlers are not capable of effective crawling of the deep web. For effective crawling of the deep web, a specialized web crawler is required, which is known as deep web crawler. Finally, this paper also presents the analysis of some of the important deep web crawlers to find out their role in the information extraction from www. One of the two important areas of application of ICT i.e. education and human resource development are discussed in details. ICT can be applied in the education field very effectively. Educational organizations are the main contributors in the field ICT revolution. Use of ICT brings out the new teaching methods with the use of new gadgets. ICT becomes the common source of information in present knowledge age. ICT enables the commons users to access the relevant information at one click at their home. ICT improves the resource functioning of human resource development management practice by making the system cost effective by minimizing the cost by providing the resources to the people without a traditional paper work system. ICT makes information omnipresent without the limitations of time, place and availability. ICT minimizes the cost of the system by providing the information to various users from one source in
less time. Using ICT human’s resource development service can be improved by redefining the responsibilities of the employees through a improved strategic orientation of human resource development.

ACKNOWLEDGMENT
All the companies/products/services names are used for identification purposes and may be the trademarks of their respective owners.

REFERENCES


A Novel Architecture for Deep Web Crawler

Dilip Kumar Sharma
Shobhit University, Meerut, UP, India
todilipsharma@rediffmail.com

A. K. Sharma
YMCA University of Science and Technology, Faridabad, Haryana, India
ashokkale2@rediffmail.com

ABSTRACT

Traditional crawler picks up a URL, retrieves the corresponding page and extracts various links from that page and adds them to the queue while, deep web crawler after adding the links to the queues checks if any form is present. If so, then it processes the forms and retrieves the required information and checks the same for next link in the queue. Various techniques have been proposed for crawling the deep web information. The developments in this area have been fruitful yet the scope is still huge i.e. even after discovering a large part of the deep web, an indefinite amount of deep web information is remains to be discovered. In this paper, analysis and comparison of some of the important deep web information crawling techniques is done to find their relative limitations and advantages. For minimizing the limitations of exiting deep web crawlers, a novel architecture for deep web crawler is proposed based on QIIIEP specifications (Sharma and Sharma, 2009). The proposed novel architecture for deep web crawler is improved from the existing deep web crawler in the sense that it is cost effective and it has features of both privatized search as well as general search for the deep web data that is hidden behind the html forms.

Keywords Deep web, hidden web, invisible web, QIIIEP, authenticate crawling, crawler.

I INTRODUCTION

Traditional web crawling techniques have been used to search the contents of the web that is reachable through the hyperlinks but they ignore the deep web contents which are hidden because there is no link is available for referring these deep web contents. The web contents which are accessible through hyperlinks are termed as surface web while the hidden contents hidden behind the html forms are termed as deep web. Deep web sources store their contents in searchable databases that produce results dynamically only in response to a direct request (Bergman, 2001). The deep web is not completely hidden for crawling. Major traditional search engines can be able to search approximately one-third of the data (He, Patel, Zhang, and Chang, 2007) but in order to utilize the full potential of web, there is a need to concentrate on deep web contents since they can provide a large amount of useful information. Hence, there is a need to build efficient deep web crawlers which can efficiently search the deep web contents. The deep web pages cannot be searched efficiently through traditional web crawler and they can be extracted dynamically as a result of a specific search through a dedicated deep web crawler (Peisu, Ke, and Qinzhen, 2008) (Sharma and Sharma, 2010). This paper attempts to find the advantages and limitations of the current deep web crawlers in searching the deep web contents. For this purpose an exhaustive analysis of existing deep web crawler mechanism is done for searching the deep web contents. In particular, it concentrates on development of novel architecture for deep web crawler for extracting contents from the portion of the web that is hidden behind html search interface in large searchable databases with the following points.

• Analysis of different existing algorithms of deep web crawlers with their advantages and limitations in large scale crawling of deep web.
After profound analysis of existing deep web crawling process, a novel architecture of deep web crawling based on QIIIEP (query intensive interface information extraction protocol) specification is proposed. Figure 1 depicts the mechanism of QIIIEP (query intensive interface information extraction protocol) specification based deep web crawler.

The remainder of this paper is organized as follows: In section II related work is discussed, Section III summarizes the architectures of various deep web crawlers. In section IV, comparison of the various deep web crawlers architectures is done. The architecture of proposed deep web crawler is presented in section V. Experimental results are discussed in section VI and finally, a conclusion is presented in section VII.

II RELATED WORK

Deep web store their data behind the html forms. Traditional web crawler can efficiently crawl the surface web but they cannot efficiently crawl the deep web. For crawling the deep web contents various specialized deep web crawlers are proposed in the literature but they have limited capabilities in crawling the deep web. A large volume of deep web data is remains to be discovered due to the limitations of deep web crawler. In this section existing deep web crawlers are analyzed to find their advantages and limitations with particular reference to their capability to crawl the deep web contents efficiently.

Application/Task Specific Human Assisted Approach

Various crawlers are proposed in literature to crawl the deep web. One of the deep web crawler architecture is proposed by (Raghavan and Garcia-Molina, 2001) S. Raghavan and H. Garcia-Molina. In this paper, a task-specific, human-assisted approach is used for crawling the hidden web. Two basic challenges are associated with deep web search, i.e. the volume of the hidden web is very large and there is a need of such type of user friendly crawler which can handle search interfaces efficiently. In this paper a model of task specific human assisted web crawler is designed and realized in HiWE (hidden web exposure). The HiWE prototype was built at Stanford and it crawls the dynamic pages. HiWE is designed to automatically
process, analyze, and submit forms, using an internal model of forms and form submissions. HiWE uses a layout-based information extraction (LITE) technique to process and extract useful information. The advantages of HiWE architecture is that its application/task specific approach allows the crawler to concentrate on relevant pages only and automatic form filling can be done with the human assisted approach. Limitations of this architecture are that, it is not precise in responding to partially filled forms and it is not able to identify and respond to simple dependency between form elements.

**Focused Crawling With Automatic Form Filling Based Approach**

A focused crawler architecture is proposed by Luciano Barbosa and Juliana Freire. This paper (Barbosa and Freire, 2005) suggests a strategy which deals with the problem of performing a wide search while avoiding the crawling of irrelevant pages. The best way is to use a focused crawler which only crawl pages relevant to a particular topic. It uses three classifiers to focus its search: Page classifier is which classifies pages, belonging to topics, Form classifier is used to filter out useless forms and Link classifiers is used to identify links that are likely to lead to pages that contain searchable form interfaces in one or more steps. Advantages of this architecture are that only topic specific forms are gathered and unproductive searches are avoided because of application of stopping criteria. A limitation of this architecture is that quality of forms is not ensured since thousands of forms are retrieved at a time. Furthermore scope of improvement in this type of crawler is that focused crawlers can be used for making domain specific crawlers like hidden web database directory.

**Automatic Query Generation for Single Attribute Database Based Approach**

A novel technique for downloading textual hidden web contents (Ntoulas, Zerfos and Cho, 2005) is proposed by Alexandros Ntoulas, Petros Zerfos and Junghoo Cho. There are two basic challenges in implementing a hidden web crawler firstly the crawler should understand and model a query interface secondly the crawler should generate meaningful queries to issue to the query interface. To address the above mentioned challenges, this paper suggests how a crawler can automatically generate queries so that it can discover and download the hidden web pages. It mainly focuses on textual databases that support single-attribute keyword queries. An advantage of this technique is that query is generated automatically without any human intervention therefore crawling is efficient and limitation of this technique is that it focuses only on single attribute databases. Further an extension can be made to this work by including multi-attribute databases.

**Task Specific and Domain Definition Based Approach**

A technique for deep web crawling based on task-specific approach is proposed by Alvarez et al. (Alvarez, Raposo, Cacheda and Pan, 2006). In this approach, a set of domain definition is provided to the crawler. Every domain definition defines a particular data collecting task. The deep web crawler recognizes the relevant query forms by using domain definition. The functioning of this model is based on a shared list of routes (URLs). The overall crawling process consists of several sub crawling processes which may run on different machines. Each crawling process selects a route from the route list, analyzes it and downloads the relevant documents. The advantage of this algorithm is that results are very effective against the various real words data collecting tasks. Further scope in this regard is to make this algorithm to be capable of automatically generate new queries from the results of the previous ones.
Focused Crawling Based Approach

A focused crawler named as DeepBot for accessing hidden web content is proposed by (Alvarez, Raposo, Pan, Cacheda, Bellas and Carneiro, 2007) Alvarez et al. In this work an algorithm is developed for developing the DeepBot, which is based on focused crawling for extracting the deep web contents. Challenges behind the crawling of the deep web contents can be broadly classified into two parts i.e. Crawling the Hidden Web at server-side and crawling the Hidden Web at client-side. Advantages of this DeepBot crawler are that a form may be used in this crawling but it may have some field that do not correspond to any attribute of the domain, Accuracy of DeepBot crawler is high when more than one associated text are present in the field. The context related to the whole form is also considered in the crawling mechanism and DeepBot is fully compatible with java-script sources. A disadvantage of such type of crawling is that all the attributes of the form should be filled completely and precisely. Problem in crawling arises when one uses sources with session mechanisms. As a future work, modification can be done in the DeepBot crawler so that it can be able to generate new quires in automatic fashion.

Sampling Based Approach

An approach to deep web crawling by sampling based technique is proposed (Lu, Wang, Liang, Chen, and Liu, 2008) by Lu et al. One of the major challenge while crawling the deep web is the selection of the queries so that most of the data can be retrieved at a low cost. A general method is proposed, which maximize the coverage of the data source, while minimizing the communication cost. The strategy behind this technique is to minimize the number of queries issued, by maximizing the possibility of the unique returns of each query. An advantage of this technique is that it is a low cost technique that can be used in practical applications and limitation of this algorithm is that the efficiency of the crawler reduces when the sample and pool size is very large. As a future work, modification can be done to lower the overlapping rate so that the maximum webpages can be downloaded.

Domain Specific Search with Relevancy Based Approach

An architectural framework of a crawler for locating deep web repositories using learning multi-agent systems is proposed (Akilandeswari, and Gopalan, 2008) by Akilandeswari et al. This paper uses multi-agent web mining system to discover pages from the hidden web. Multi-agents system is used when there are troubles in storing large amount of data in the database indices. The proposed system has variety of information agents interacting with each other to learn about their environment so that they can retrieve desired information effectively. Advantages of this framework are that crawling through this technique is efficient because the searching is concentrated on a specific domain. The crawling technique of this framework extract the relevant web forms by using the learning multi-agents and it learns effectively which reduces form retrieval time. Limitation of this framework is that it is not easy to maintain multiple agents. In future, this framework can be extended so that genetic algorithms can be implemented in the crawler to perform a broad search to improve the harvest rate.

Domain-Specific Deep Web Sources Discovery Based Approach

A technique of domain-specific deep web sources discovery is proposed (Wang, Zuo, Peng and He, 2008) by Wang et al. It is difficult to find the right sources and then querying over them online in huge collection of useful databases. Hence, this paper presents a new method by importing focused crawling technology to
automatically accomplish deep web sources discovery. Firstly, websites are located for domain-specific
data sources based on focused crawling. Secondly, it is judged where the website exists in deep web query
interface. Lastly, judgment is done to find whether the deep web query interface is relevant to a given topic.
Implementation of focused crawling technology facilitates the identification of deep web query interface
located in a specific domain and capturing of relevant pages associated with the topic. This method has
dramatically reduces the quantity of pages for the crawler to crawl the deep web. Advantage of this
technique is that fewer numbers of pages need to be crawled since it applies focused crawling along with
the relevancy search of the obtained results about the topic. Limitation of this technique is that it does not
take into account the semantics i.e. a particular query result could have several meanings. As a future work
this technique can be extended to include semantics while querying.

Input Values for Text Search Inputs Based Approach

A technique for surfacing deep web contents is proposed by Madhavan et al. (Madhavan, Ko, Kot,
Ganapathy, Rasmussen and Halevy, 2008). Surfacing the deep web is a very complex task because html
forms can be associated with different languages and different domains. Further large quantities of forms
are associated with text inputs which require the submission of valid input values. For this purpose authors
have proposed a technique for choosing input values for text search inputs by which keywords can be
accepted. Advantage of this technique is that it can efficiently navigate for searching against various
possible input combinations. The limitations as well as future work in this regard can be to modify the
technique to deal with forms associated with java script and to analyze the dependency between the values
in various inputs of a form in more depth.

Label Value Set (LVS) Table Based Approach

A framework of Deep Web Crawler is proposed by Peisu et al. (Peisu, Ke, and Qinzhen, 2008). The
proposed framework processes the actual mechanics of crawling of deep web. This paper deals with the
problem of crawling a subset of the currently unscrawled dynamic web contents. It concentrates on
extracting contents from the portion of the web that is hidden behind search forms in large searchable
databases. This proposed framework presents a model of form with form submission facility. One of
important characteristics of this proposed crawler is that it uses Label Value Set (LVS) table. Access and
additions to the LVS table is done by LVS manager. The LVS manager also works as an interface for
different application specific data sources. Advantage of this deep web crawler is that it uses the additional
modules like LVS table which help the crawler to design the model of frame. If the crawler uses the LVS
table than the number of successful form submissions increase. Limitations of this framework are that
crawler is unable to extract label (E) and the value of domain (D) in LVS is repeated.

Minimum Executing Pattern (MEP) & Adaptive Query Based Approach

A technique for crawling deep web contents through query forms is proposed by Liu et al. (Liu, Wu, Jiang,
Zheng, Liu, 2009). The proposed technique of crawling the deep web content is based on Minimum
executing pattern (MEP). The query in this technique is processed by deep web adaptive query method.
Query interface is expended from single text box to MEP set by using deep web adaptive query method. A
MEP, associated with keyword vector is selected for producing optimum local query. Advantages of this
technique are that it can handle a different web forms very effectively and its efficiency is very high
compared to non prior knowledge methods. Further it also minimizes the problem of “data islands” to some
extent. The limitation of this technique is that it does not produce good results in case of the deep web sites which have limited size of result set. Further boolean logic operators such as AND, OR, NOT cannot be used in queries which can be a part of the future work.

**Iterative Self Organizing Data Analysis (ISODATA) Based Approach**

A novel automatic technique for classifying deep web sources is proposed by Zhao (Zhao, 2010). The classification of deep web sources is a very critical step for integrating the large scale deep web data. This proposed technique is based on iterative self organizing data analysis (ISODATA) technique. This technique is based on hierarchical clustering method. Advantage of this technique is that it allows the user to browse the relevant and valuable information. Method for extraction of characteristics in web pages can be further improved to browse the valuable information.

**Continuously Update or Refresh the Hidden Web Repository Based Approach**

A framework for incremental hidden web crawler is proposed (Madaan, Dixit, Sharma and Bhatia, 2010) by Madaan et al. In a world of rapidly changing information, it is a highly required to maintain and extract the up-to-date information. For this, it is required to verify whether a web page has been changed or not. The time period between two successive revisits needs to be adjusted based on probability of updation of the web page. In this paper, architecture is proposed that introduces a technique to continuously update or refresh the hidden web repository. Advantages of this incremental hidden web crawler is that information fetched is updated even if web pages change and limitation of this incremental hidden web crawler is that efficient indexing technique is required to maintain the web pages in the repository. In future, a modified architecture of a search engine based on incremental hidden web crawler using some indexing technique can be designed to index the web pages that stored in the repository.

**Reinforcement Learning Based Approach**

An efficient deep web crawling technique using reinforcement learning is proposed by Lu Jiang et al. (Jiang, Wu, Feng, Liu and Zheng, 2010). In the reinforcement learning technique, the deep web crawler works as agent and deep web database plays a role of environment. The deep web crawler identifies a query to be submitted into a deep web database, depending upon Q-value. The advantage of this technique is that deep web crawler itself decides about a crawling technique to be used by using its own experience. Further, it also permits the use of different characteristics of query keywords. Further scope in this technique is that to develop an open source platform can be developed for deep web crawling.

After going through analysis of deep web crawlers, it is concluded that each deep web crawler has certain limitations with reference to their capabilities of efficient crawling of the deep web contents. Some of the challenges for efficient deep web crawling are that a crawler should not overload the web servers. A deep web crawler must be robust against hazardous situations. It must be fault tolerant so that its performance degrades gracefully. It should be highly configurable. The download rate of a crawler must be adequate so as to process the harvested data. A crawler must be flexible to enable quick adoption to new publishing technologies and formats used on the web as they become available. The crawler must include the management tools that enable the quick detection of its failure. The deep web crawler should have a focused way of crawling the information from the deep web. They should automatically download the pages from the deep web so that search engine can index them.
III SUMMARY OF VARIOUS DEEP WEB CRAWLER ARCHITECTURES

By going through the literature analysis of some of the deep web crawlers, It is concluded that every crawler have some relative advantages and limitations. A tabular summary is given below in table 1, which summarizes the techniques, advantages and limitations of some of important deep web crawlers.

<table>
<thead>
<tr>
<th>Authors, Year</th>
<th>Technique</th>
<th>Advantages</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raghavan et al., 2001</td>
<td>Extraction is application/task specific.</td>
<td>Extraction of irrelevant pages is minimized.</td>
<td>Crawling is not precise due to possibility of missing of some pages.</td>
</tr>
<tr>
<td>Barbosa et al., 2005</td>
<td>Focused crawling with automatic form filling.</td>
<td>Crawling is highly relevant, which saves time and resources.</td>
<td>Quality of forms is not ensured and form verification is a complex task.</td>
</tr>
<tr>
<td>Ntoulas et al., 2005</td>
<td>Based on automatic query generation form.</td>
<td>Efficient crawling due to crawler generated query.</td>
<td>Does not involve frequently used multi-attribute database.</td>
</tr>
<tr>
<td>Alvarez et al., 2006</td>
<td>Task specific approach. A set of domain definition is provided to the crawler. Every domain definition defines a particular data collecting task. The deep web crawler recognizes the relevant query forms by using domain definition.</td>
<td>Results are very effective against the various real world data collecting task.</td>
<td>Algorithm can be modified to be able to automatically generate new queries from the results of the previous ones.</td>
</tr>
<tr>
<td>Alvarez et al., 2007</td>
<td>Focused crawling for extracting deep web contents</td>
<td>Accuracy is high with the field having more than one associated text. The context of the whole form is used. Fully compatible with java-script sources</td>
<td>The form should be filled precisely and completely. Difficulty with sources having session mechanism.</td>
</tr>
<tr>
<td>Lu et al., 2007</td>
<td>Sampling data from the database.</td>
<td>Low cost, Efficient in practical applications.</td>
<td>Efficiency is less in case of large sample and pool size.</td>
</tr>
<tr>
<td>Akilandeswari et al., 2008</td>
<td>Use of multi-agent system on a large database.</td>
<td>Time efficient, fault tolerant and easy handling due to multi-agents.</td>
<td>Cost may be high due to maintenance of multi-agents.</td>
</tr>
<tr>
<td>Wang et al., 2008</td>
<td>Focused crawling and results are located in a specific domain.</td>
<td>Crawl fewer numbers of pages due domain specific technique.</td>
<td>Sometimes semantics may be wrong due to crawling of useless pages.</td>
</tr>
<tr>
<td>Madhavan et al., 2008</td>
<td>Input values for text search inputs are selected. Identification of inputs for a particular type of values.</td>
<td>It can efficiently navigate for searching against various possible input combinations.</td>
<td>Technique can be modified to deal with forms associated with java script and the dependency between the values in various inputs of a form can be analyzed in more depth.</td>
</tr>
<tr>
<td>Peisu et al., 2008.</td>
<td>Proposes a model of form. Form submission with four additional modules with LVS table.</td>
<td>Successful form submissions increase with the use of LVS table.</td>
<td>If crawler is unable extract label (E) then the value of domain (D) in LVS is repeated.</td>
</tr>
<tr>
<td>Liu et al., 2009.</td>
<td>Based on the concept of minimum executable pattern (MEP).</td>
<td>Effective handing of different web forms. Higher efficiency against non prior knowledge method. Reduces the problem of “data islands”.</td>
<td>Results are not good with websites having limited size of result set. Boolean logic operators (AND,OR,NOT) cannot be used.</td>
</tr>
<tr>
<td>Zhao, 2010.</td>
<td>Based on iterative self organizing data analysis (ISODATA) technique.</td>
<td>Allows the user to browse the relevant and valuable information.</td>
<td>Extraction method of characteristics in web pages can be further improved to browse the valuable information.</td>
</tr>
<tr>
<td>Madaan et al., 2010</td>
<td>Regularly updates web repository.</td>
<td>Fetched web pages are regularly updated.</td>
<td>Indexing of web page is required.</td>
</tr>
<tr>
<td>Jiang et al., 2010</td>
<td>Based on reinforcement learning technique. Deep web crawler works as agent and deep web database plays a role of environment. Identify a query to be submitted into a deep web database, using Q-value.</td>
<td>Deep web crawler itself decides about a crawling technique to be used by using its own experience. Permits the use of different characteristics of query keywords.</td>
<td>Can be developed as an open source platform for deep web crawling.</td>
</tr>
</tbody>
</table>
IV COMPARISON OF THE VARIOUS DEEP WEB CRAWLER ARCHITECTURES

Based on the literature analysis, a comparison of some of various deep web crawlers architectures is shown in table 2 and in table 3. Comparison is done on the basis of some parameters such as technique used, need of user support, reflection of change in web page, automatic query selection; accuracy of data fetched, database sampling and focused crawling.

Table 2: Comparison of the various deep web crawler architectures

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Crawling technique</th>
<th>Crawling the hidden web</th>
<th>Searching for hidden web databases</th>
<th>Downloading Textual Hidden Web Contents</th>
<th>Domain specific deep web sources discovery</th>
<th>Approach to deep web crawling by sampling</th>
<th>Framework for incremental hidden web crawler</th>
<th>Locating deep web repositories using multi-agent systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technique used</td>
<td>Application/task specific human assisted approach.</td>
<td>Focused crawling with automatic form filling.</td>
<td>Automatic query generation for single attribute database.</td>
<td>Domain specific relevancy based search.</td>
<td>Sampling of data from the database.</td>
<td>Frequent updation of web repository.</td>
<td>Multi-agent system is helpful in locating deep web contents.</td>
<td></td>
</tr>
<tr>
<td>Need of the user’s support</td>
<td>Human interface is needed in form filling.</td>
<td>No human interface is needed in form filling.</td>
<td>User monitors the filling process.</td>
<td>It doesn’t require user’s help.</td>
<td>Not mentioned.</td>
<td>It doesn’t require user’s help.</td>
<td>Users monitor the process.</td>
<td></td>
</tr>
<tr>
<td>Reflection of web page changes</td>
<td>It doesn’t reflect such a change</td>
<td>No concept involved for dealing with it.</td>
<td>It doesn’t reflect such a change.</td>
<td>Such changes are not incorporated</td>
<td>It doesn’t reflect such a change.</td>
<td>It keeps refreshing the repository for such changes.</td>
<td>It doesn’t reflect such a change.</td>
<td></td>
</tr>
<tr>
<td>Automatic query selection</td>
<td>Such feature has not been incorporated</td>
<td>Nothing is mentioned about such a feature.</td>
<td>Query selection is automatic.</td>
<td>Nothing is mentioned about such a feature.</td>
<td>Automated query selection is done.</td>
<td>Nothing is mentioned about such feature.</td>
<td>Nothing is mentioned about such feature.</td>
<td></td>
</tr>
<tr>
<td>Accuracy of data fetched</td>
<td>Data fetched can be wrong if the web pages change.</td>
<td>Data fetched can be wrong if the web pages change.</td>
<td>Data fetched can be wrong if the webpages change.</td>
<td>Data fetched can be wrong if the webpages change.</td>
<td>Data fetched can be wrong if the webpages change.</td>
<td>Data fetched can be wrong if the webpages change.</td>
<td>Only correct data is obtained since repository is refreshed at regular time interval.</td>
<td></td>
</tr>
<tr>
<td>Database sampling</td>
<td>Such feature is not incorporated.</td>
<td>Such feature is not incorporated.</td>
<td>Such feature is not incorporated.</td>
<td>Large database is sampled into smaller units.</td>
<td>Such feature is not incorporated.</td>
<td>Such feature is not incorporated.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Focused crawling</td>
<td>Focused crawling is not involved although it is task specific.</td>
<td>Focused crawling is the basis of this work.</td>
<td>Nothing is mentioned about focused crawling.</td>
<td>Focused crawling is done.</td>
<td>Nothing is mentioned about such concept.</td>
<td>Nothing is mentioned about it.</td>
<td>Nothing is mentioned about it.</td>
<td></td>
</tr>
<tr>
<td>Crawling technique</td>
<td>Parameters</td>
<td>Technique Used</td>
<td>Need of user’s support</td>
<td>Reflection of web page change</td>
<td>Automatic query selection</td>
<td>Accuracy of data fetched</td>
<td>Database sampling</td>
<td>Focused crawling</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>-------------------------------</td>
<td>------------------------------</td>
<td>--------------------------</td>
<td>------------------------</td>
<td>-------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>A Framework of Deep Web Crawler</td>
<td></td>
<td>Proposes a model of form with form submission facility with four additional modules with LVS table.</td>
<td>Human interaction is not needed for modeling the form.</td>
<td>No effect of web page changes in the result.</td>
<td>Yes</td>
<td>Good</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Google's Deep-Web Crawl</td>
<td></td>
<td>Input values for text search inputs are selected and identification of input for a particular type of values.</td>
<td>Human interface is needed in form filling.</td>
<td>There is effect of these changes in the result because it is based on the content of the pages.</td>
<td>No</td>
<td>High</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Efficient Deep Web Crawling Using Reinforcement Learning</td>
<td></td>
<td>Based on reinforcement learning technique. Deep web crawler works as agent and deep web database plays a role of environment. Identify a query to be submitted using Q-value.</td>
<td>Human interaction is required</td>
<td>There is effect of these changes in the result but changes may generate an error.</td>
<td>Yes</td>
<td>Average</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Crawling Deep Web Contents through query forms</td>
<td></td>
<td>Minimum executable pattern (MEP) and based adaptive query technique.</td>
<td>Need of user support.</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Study of Deep Web Sources Classification Technology</td>
<td></td>
<td>Based on iterative self organizing data analysis (ISODATA) technique.</td>
<td>No need of user support.</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>A Task-Specific Approach for Crawling the Deep Web</td>
<td></td>
<td>Task specific approach. A set of domain definition is provided to the crawler. Every domain definition defines a particular data collecting task. The deep web crawler recognizes the relevant query forms by using domain definition.</td>
<td>No need of user support.</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

## V PROPOSED WORK

By going through literature analysis with their relative comparison with reference to efficient deep web crawling, it is observed that each deep web crawler has limitations in efficient crawling of the deep web. To
fulfill this need, a novel architecture for efficient deep web crawling is proposed with particular reference to QIIIEP specification. Proposed architecture incorporates all the features of existing deep web crawlers and tries to minimize the limitations of existing deep web crawlers. Figure 2 shows the architecture of proposed novel deep web crawler mechanism.

Figure 2 Architecture of proposed deep web crawler mechanism
Description of Modules of Proposed Architecture for Deep Web Crawler Mechanism

Module wise description of proposed novel architecture of deep web crawler is given below.

**Agent for Authenticated Crawling Module**

This module works when the information in a site is hidden behind the authentication form. It stores authentication credentials of every domain in its knowledge base situated at crawler, which is provided by the individual user. At the time of crawling, it automatically authenticates itself on the domain for crawling the hidden web contents. The crawler extracts and store keywords from contents and makes it available to privatized search service to maintain the privacy issue.

**Page Fetcher Module**

Page fetcher fetches the pages from the http server and sends them to the page analyzer to check whether it is required appropriate page or not based on the topic of search and the kind of form, the page contains.

**Page Analyzer/Parser Module**

The page analyzer/parser is used to parse the contents of the web page. Texts and links are extracted. The extraction of links and text is done on the basis of the concept of page classifier. Form classifier link is used to filter the pages topic wise. It also filters out useless forms and identifies links that are likely to lead to pages that contain searchable form interfaces.

**Form id Generator**

This module helps to implement QIIIEP (query intensive interface information extraction protocol) on current architecture of website. It parses every form of that web site and merge the form ID with QIIIEP server query word list so that at the time of crawler request the crawler, it identifies the search interface properly for sending the request of keywords to the QIIIEP server.

**Form Submitter Module**

After filling the form, the form submitter sends again the request to the HTTP server for the further retrieval. This module simply sends the filled form to the http server for further information.

**Query Word to URL Relation Manager Module**

This module generates meaningful queries to be issued to the query interface. It stores each and every query word associated with specific element of form by creating reference of domain path, so that at the time of link generation, query word can be mapped to provide the contents by sending query word in post request to the domain.

**Crawl Frontier**
Crawl Frontiers contains all the links which are yet to be fetched from the HTTP server or the links obtained after URL filter. It takes a seed URL for starting the procedure and processes that page and retrieve all the form and add and rearrange them to the list of URL and rearrange them. The list of those URL is called crawl frontier.

**Link Extractor Module**

Link Extractor extracts the links or the hyper links from the text file for the further retrieval from the HTTP server. The extraction of links is done as per the link identified by the page analyzer/parser that is likely to lead to pages that contain searchable form interfaces in one or more steps. This dramatically reduces the quantity of pages for the crawler to crawl in deep web. Fewer numbers of pages are needed to be crawled since it applies focused crawling along with searching the relevancy of obtained result to the topic and hence result in limited extraction of relevant links.

**QIIIEP (query intensive interface information extraction protocol) Server**

The QIIIEP (query intensive interface information extraction protocol) (Sharma and Sharma, 2009) is an application-level protocol for semantic otology based query word composition, identification and retrieval systems. It is based on request/response semantics. This specification defines the protocol referred to as QIIIEP 1.0. The QIIIEP server work on this protocol. The QIIIEP (Query Intensive Interface Information Extraction Protocol) reduces complexity by using pre-information about the form and its elements from QIIIEP server. The knowledge base is either generated by auto query word extractor or it is provided by site administrator.

**Link Ranker Module**

This module is responsible for providing the best match query word assignment to the form filling process and reduce the over loading due to less relevant queries to the domain. It is required to rank the link accordingly so the more information is gathered from each link. This is based on link ranking algorithm.

**Link Indexer Module**

This module plays an important role in the indexing of the generated keywords to the content database. Indexer collects, parses, and stores data to facilitate fast and accurate information retrieval. It maps the keywords to URL for the fast access and retrieval.

**Content Database Module**

Content Data Base stores all the generated links or keywords in the Content Data Base. When user put any query into the user interface, the index is matched with the corresponding links and information is displayed to the user for further processing.

**Searching Agent Module**
It provides the searching interface through which user places the query. This involves the searching of keywords and other information stored in the content database which is actually stored in it after the whole process of authenticated crawling.

**Link Composer Module**

This module takes the reference from the Query word to URL Manager for the form submission.

**Interface Generator Module**

Interface generator is used to give the view of the contents stored in the content database after the search is completed. For example, the interface generator shows the list of relevant links indexed and ranked by link ranker module and link indexer module respectively.

**Link Event Analyzer**

Link event analyzer analyzes the link which is activated by the user so that it could forward the request to display the page on the requested URL.

**User Authentication to Domain Mapping Module**

This module of crawler is responsible for mapping the login credentials provided by users to the crawler with the information provider domain. The main benefit of using this mapper is to overcome the hindrance of information retrieval between result link and information. The crawler uses the mapping information to allow the specific person to receive information contents directly from the domain by automatic login procedure and eliminate the step of separate login for user.

**Working Steps of proposed Architecture for deep web crawler mechanism**

1. Crawler request to the web server to fetch a page.
2. a). After the page is fetched it is being analyzed and parsed for the relevant contents (links and text).
   b). The page is sent to the query word to URL relation manager.
   c). If the authentication of administrator credentials is required then the links are sent to the agent for authenticated crawling.
3. After being analyzed by the page analyzer/parser, links are selected and filtered out.
4. Filtered URLs sent to crawl frontier, which again chooses a link and sends it to the page fetcher.
5. Now, crawler analyzes the form to identify the search interface. The form must include the form id.
6. Then the form id is used to send the request to query word to the QIIIEP (query intensive interface information extraction protocol) server, where the extraction takes place to correlate the form fields.
7. Now, the server replies to the crawler about each entry to that form.
8. Crawler sends the filled form by placing the received query words to the HTTP server.
9. Crawler crawl the contents generated by that query word.
10. Finally fetched pages are ranked, indexed and stored in the search engines database

User interacts with user interface through the following steps:
1. User enters the query about search.
2. The query words validation takes place and link composer fetches the link from the database according to the query word.
3. Now links are searched from the content database.
4. The content is then provided to the user with the help of interface generator.
5. The link i.e. chosen by the user divert the user for user authentication through domain mapper so that the user can retrieve those authenticated contents without explicit login.
6. Query word, submitted to URL relation manager use the post and get request to generate the specific page.
7. The link opens the website in a browser.

**Algorithm for the simple web crawler is given below:**

**Input** initial URL = seed.

1. Maintain a queue Q={u}.
2. While(Q!=NULL)
3. Pop a element from Q.(using FIFO)
4. Process that URL and fetch all the relevant URL’s.
5. Assign an unique index for each page visited.
6. For each relevant URL’s fetched (URL1,URL2,URL3…..)
7. If(URL1 is not indexed && URL1 does not belong to Q )
8. Add URL1 to Q.
9. end.

**Algorithm for deep web data extraction is given below:**

1. Extract web page by using initial seed URL or crawl frontier.
2. Analyze for contents and form.
3. Extract query URL from page and store in crawl frontier after applying filter.
4. If page content form.
5. Request query word for all the elements in form from QIIIEP server.
6. Submit and extract the deep web contents.
7. Manage query word to URL relation.
8. Rank and index the page and store in content database.
9. Go to step 1

**Algorithm for search and result link generator is given below:**

1: User input in the form of query.
2: If this is a valid query then goto step3
3: Search the content database as follows
   3(a): Efficient query is generated
   3(b): Check the indexed words and the corresponding links
3(c): IF query words match then
    Select the appropriate links & goto step 4.
    else:
        goto interface generator & display keywords did not match & stop.
4: If found then Interface generator will display the results.
5: Link event analyzer will take user authentication to domain mapping if site want login.
6: If user’s credential found authenticated on website and open specific page by using query word to URL relation manager.
    else:
        Display login form of that web site.
7: If web site does not want login.
8: Just open the deep web content by using query word to URL relation manager.

// Code for parser, downloader and link repository module
void Crawler_book_1() throws IOException, MalformedURLException
{
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection con=DriverManager.getConnection("jdbc:odbc:web");
        Statement stmt=con.createStatement();
        ResultSet rs=stmt.executeQuery("select id from book");
        while(rs.next())
        {
            count=Integer.parseInt(rs.getString(1));
        }
        try{
            urlc =new URL(seedurl);
            pageInput = new InputStreamReader(urlc.openStream());
            source = new BufferedReader(pageInput);
        }
        catch(NullPointerException e){}
        while ((sourceLine = source.readLine()) != null)
        {
            content += sourceLine ;
        }
        tag = Pattern.compile("href="(.*?)",Pattern.DOTALL);
        mtag = tag.matcher(content);
        while (mtag.find())
        {
            content = mtag.group(1);
            if(content.startsWith("http:"))
            {
                if(!(content.endsWith(".css")||content.endsWith(".xml"))
                {
                    // System.out.println(""+content);
                    dsp=""+content;
                    ta.append(dsp+"
                    if(!ar.contains(content))
                    {
                        ar.add(content);
                    }
ai = ar.iterator();
while (ai.hasNext())
{
    content1 = null;
    runner.sleep(100);
    System.out.println(ai.hasNext());
    o = ai.next();
    urls = new URL(""+o);
    pageInput1 = new InputStreamReader(urls.openStream());
    ta.append(dsp+"\n");repaint();
    source1 = new BufferedReader(pageInput1);
    dsp = "link";
    ta.append(dsp+"\n");repaint();
    while ((sourceLine = source1.readLine()) != null)
    {
        !(content1.contains("under construction")||content1.contains("cannot be displayed ")||content1.contains("not available")))
            content1 += sourceLine ;
    }
    tag1 = Pattern.compile("<form([^<]*?)</form>", Pattern.DOTALL);
    mtag1 = tag1.matcher(content1);
    dsp = "The Forms Link Are:....."+mtag1.find();
    ta.append(dsp+"\n");repaint();
    while ((mtag1.find()))
    {
        content1 = mtag1.group(0);
        dsp = ""+content1;
        ta.append(dsp+"\n");repaint();
        tag2 = Pattern.compile("book", Pattern.DOTALL);
        mtag2 = tag2.matcher(content1);
        tag3 = Pattern.compile("author", Pattern.DOTALL);
        mtag3 = tag3.matcher(content1);
        tag4 = Pattern.compile("title", Pattern.DOTALL);
        mtag4 = tag4.matcher(content1);
        tag5 = Pattern.compile("isbn", Pattern.DOTALL);
        mtag5 = tag5.matcher(content1);
        if(mtag2.find()||mtag3.find()||mtag4.find()||mtag5.find())
        {
            dsp = "Both r found";
            ta.append(dsp+"\n");repaint();
            tag6 = Pattern.compile("action=\"([\w\W]+)\"", Pattern.DOTALL);
            mtag6 = tag6.matcher(content1);
            System.out.println("tag6"+mtag6.find());
            if(!(mtag6.group(1).contains("http")))))
            {
                //System.out.println("www found");
                dsp = "www found";
                ta.append(dsp+"\n");repaint();
                content1 = mtag6.replaceAll("action="+o+mtag6.group(1));
            }
        }
        try
Features of the theoretically justified proposed architecture:

Features of the theoretically justified proposed architecture for deep web crawler is given below:

1. This proposed architecture crawl the deep web if the administrator of that site follows the framework of QIIIEP based on focused crawling.
2. It definitely removes the complex task of query interface identification and values allotment as the huge majority of deep web query interfaces on the web are html forms.
3. The program tries to classify deep web query interface according to different users.
4. Dynamic discovery of deep web sources are done according to the user’s query.
5. Provide input i.e. auto of filling the form of search queries.
6. Auto form ID generation & auto query extraction modules are used by QIIIEP server to extend its knowledge.
7. Privatized search plus general domain search are the two features which are based on the overall protocol.
8. Authenticated crawling is provided for privatized search.
9. Forms are classified with various levels depending on the administrator.
10. Different domains will have their content links and indexes in different databases.
11. Implicit authentication takes place when registered user clicks on a link at domain site.
12. The final content page opened after link is clicked which was crawled by post or get request.
13. Only the meta data of private area is stored so there is no privacy issue for crawling authenticated private content.
14. There is a huge list of query word generated through cross-site user query submission module but the ranking algorithm choose most appropriate query word for specific query interface to reduce the bandwidth wastage.

Limitations of existing deep web crawlers that are improved through proposed architecture:

1. Crawling is precise and the chances of missing of pages are less. As we are using multithreaded downloading techniques so multiple threads run simultaneously which eliminate the chances of page miss. URL filter module makes the crawling very precise, eliminating the duplicate links.
2. It involves multi-attribute database accessing.
3. Form verification can be done very easily with the help of our form manager which extracts different domains attribute accordingly.
4. The form filling can be done very precisely and the user needs not to fill the complete form.
5. Extraction method that is improved by query word extraction module.
6. Useless pages if crawled cannot affect the semantics.
7. Time and cost is less due to the use of multithreaded downloading technique and automatic form filling by the QIIIEP server.
8. Forms with searchable interfaces are provided directly to the users.
9. Web page forwarding technique that is used while forwarding by submitting the form has no chances of error at all. If the action part is not start with ="http: //" then we have attached the
relative link part to the web site, so if any relative link comes then it is attached with the web site and is forwarded to related page.

VI EXPERIMENTAL RESULTS

The experiment has conducted on a machine having Intel Core 2 Duo T5870 @ 2.0 GHz with 2 GB of RAM. This machine was running with Windows 7 OS. Tests are performed using WAMP server equipped with php v. 5.2.6 and mysql v. 5.0.51b., Microsoft Visual Basic 2008 .net 3.5 and Net beans IDE. All of the tests were performed on Firefox 3.5.4., All tests were performed multiple times to minimize measurement errors. Results of the various modules are shown in table 4.

<table>
<thead>
<tr>
<th>Modules</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Page Fetcher</td>
<td>The downloaded html content</td>
</tr>
<tr>
<td>2. Page Analyzer</td>
<td>Selected pages which contain forms.</td>
</tr>
<tr>
<td>3. Form Id Manager</td>
<td>A list of Search Interfaces (forms).</td>
</tr>
<tr>
<td>4. QIIIEP SERVER</td>
<td>Provide query word to the forms wherever possible &amp; give result to response analyzer.</td>
</tr>
<tr>
<td>5. Form Submitter</td>
<td>Send the filled form to the Http server.</td>
</tr>
<tr>
<td>6. Link Extractor</td>
<td>A list of extracted links those are relevant.</td>
</tr>
<tr>
<td>7. Link Ranker Module</td>
<td>A list of ranked links.</td>
</tr>
<tr>
<td>8. Link Composer Module</td>
<td>Composed list of links by using query word to URL relation manager.</td>
</tr>
<tr>
<td>9. Interface Generator</td>
<td>Shows the list of relevant links ranked by rank module &amp; link indexer.</td>
</tr>
<tr>
<td>10. Link Filter</td>
<td>Fresh links are shown.</td>
</tr>
</tbody>
</table>

Certain performance metric are defined and calculated to judge the overall performance of the deep web crawler. Objectives of the performance metric are given below.

1. Analytical data based on the number of sites visited which reflects the reach of the crawler.
2. Assessment of the different forms which show the ratio of relevant forms processed.
3. Calculation of the performance ratio to check the overall efficiency of the deep web crawler.

Performance Metrics are calculated as follows:
The various performance criteria are taken to measure the overall efficiency of proposed deep web crawler architecture and its results are as follows:

1. Deep web crawler is implemented based on QIIIEP specification and the corresponding the results are analyzed. Three different test domains are used to judge the precision of retrieved contents.
Table 5 shows the form identification statistics and table 6 shows the query words and content extraction statistic.

**Table 5: Form Identification Statistics**

<table>
<thead>
<tr>
<th>Domain</th>
<th>Query Forms</th>
<th>Context Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Book</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Job</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

**Table 6: Query words and content extraction statistics**

<table>
<thead>
<tr>
<th>Domain</th>
<th>Form Id</th>
<th>Query Words Received</th>
<th>Successful Content Extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
<td>55c64ad2fDd6a6ef4388b33c54123890</td>
<td>34</td>
<td>30</td>
</tr>
<tr>
<td>Book</td>
<td>eOcf1fd062a28403d966261Ob421eOc</td>
<td>56</td>
<td>42</td>
</tr>
<tr>
<td>Job</td>
<td>67c76add7110dbe02Ob401a4672565f</td>
<td>131</td>
<td>96</td>
</tr>
</tbody>
</table>

The graph shown in figure 3 is plotted between received query words and successful content extraction at different domain. It can be concluded from the graph that contents extraction are close to query words received at a satisfactory level for all three domains. The successful content extraction means relevant page have more than five query words and minimum length of content is four hundred words.

![Figure 3 Comparison of success with number of query words at different domain](image)

**Figure 3** Comparison of success with number of query words at different domain

2. Table 7 shows the default values for some of the parameters that are used for experiments.
As indicated in Table 7, the crawler encountered 41 forms during crawling of the 12 sites, of which 2 were ignored, because form id manager do not found any associated id corresponding to the sites.

Figure 4 shows the number of links generated for a particular query word search and the overall count of pages indexed with given query words.

![Graph showing number of links generated for a particular query words search](image)

Figure 4: Graph showing number of links generated for a particular query words search

The graph shows the number of pages in result for the specific query word. It depends upon the total number of crawled pages. For overall 1508 stored pages the results have good ratio.

3. **Harvest Ratio:**

To evaluate the performance of the proposed architecture, the “Harvest Ratio” is adopted as the performance metrics, which is the ratio of relevant web pages downloaded among all the downloaded web pages. Here relevant downloaded pages are those which have at least five times repeated query word in overall content. It is defined as follows:

\[
HR = \frac{RWPD}{AWPD} \times 100,
\]
Where: HR= Harvest Ratio, RWPD: Relevant web pages downloaded & AWPD: All web pages downloaded

Table 8 depicts the domains and its relevant web pages downloaded & all web pages downloaded statistic.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Job</th>
<th>Book</th>
<th>Auto</th>
</tr>
</thead>
<tbody>
<tr>
<td>RWPD</td>
<td>629</td>
<td>377</td>
<td>93</td>
</tr>
<tr>
<td>AWPD</td>
<td>795</td>
<td>531</td>
<td>182</td>
</tr>
</tbody>
</table>

The harvest ratio for different domain is shown in table 9.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Harvest Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job</td>
<td>79.11 %</td>
</tr>
<tr>
<td>Book</td>
<td>77.99 %</td>
</tr>
<tr>
<td>Auto</td>
<td>51.09 %</td>
</tr>
</tbody>
</table>

The result shows very good harvest ratio when considering a focused domain approach which reflects that proposed architecture is significantly improved compared to existing approach.
Summary of the results of proposed architecture:

The features of proposed architecture are improved from the features of the existing crawlers as proposed architecture deals with the overall strategy of hidden query interfaces including the features of both privatized search and general search for the deep web data that is hidden html forms. Every solution is taken into consideration to make the crawler as efficient as possible including all the features of the existing web crawlers. Proposed architecture tries to minimize the overall cost and time that are relevant to the deep web searching. However at some points, time and space complexity are compromised with performance but overall results are as per expectation. Proposed architecture performs better compared to other existing crawlers as reflected from performance metric. For example, the performance of the proposed architecture is better and satisfactory with reference to number of links crawled. Cost of the deep web search also reduces through this crawler due to searching with domain specific formula. Initial seed set of links is such that the number of relevant search interfaces and forms are quite effective. Pre determination of the domain context provide effective results with more than 40% effective links with forms extraction on every loop while generating links for the databases. At last performance metric are calculated after the implementing and integrating all the modules and necessary modifications are done to improve the model.

VII CONCLUSION

Deep web information has a very large volume compared to surface web and the quantity of deep web content depends upon underlying domain and crawling mechanism. Extraction of deep web information can be highly fruitful for a general user or a specific user. Traditional web crawlers have limitations in crawling the deep web information so some of the web crawlers are specially designed for crawling the deep web information yet a very large amount of deep web information is yet to be explored due to inefficient crawling of the deep web. In literature survey, analysis of some of the important deep web crawlers is done to find their advantages and limitations. A comparative analysis of deep web crawlers is also done on the basis of various parameters and it is concluded that a new architecture for deep web crawler is required for efficient searching of the deep web information by minimizing the limitations of the existing deep web crawlers as well as incorporating the strengths of the existing deep web crawlers. The
architecture should be compatible to crawl existing deep web with nominal modification with ongoing infrastructure based on QIIIEP specification. A novel architecture for deep web crawler is proposed which possesses all the features of existing deep web crawlers but tries to minimize limitations of existing deep web crawlers. Experiments results reflect that it is efficient both for privatized as well as general search for the deep web information, which is hidden behind the html forms. Proposed architecture is cost and time effective as search process depends on query interface crawling the contents with ranking of most appropriate keyword against the context of domain.

REFERENCES


About the Authors

Dilip Kumar Sharma is B.Sc, B.E.(CSE), M.Tech.(IT), M.Tech. (CSE) and pursuing Ph.D in Computer Engineering. He is life member of CSI, IETE, ISTE, ISCA, SSI and member of CSTA, USA. He has attended 21 short term courses/workshops/seminars organized by various esteemed originations. He has published 23 research papers in International Journals /Conferences of repute and participated in 18 International/National conferences. Presently he is working as Reader in Department of Computer Science, IET at GLA University, Mathura, U.P. since March 2003 and he is also CSI Student branch Coordinator. His research interests are deep web information retrieval, Digital Watermarking and Software Engineering. He has guided various projects and seminars undertaken by the students of undergraduate/postgraduate.

Prof. A. K. Sharma received his M.Tech. (CST) with Hons. from University of Roorkee (Presently I.I.T. Roorkee) and Ph.D (Fuzzy Expert Systems) from JMI, New Delhi and he obtained his second Ph.D. in Information Technology from IIITM, Gwalior in 2004. Presently he is working as the Dean, Faculty of Engineering and Technology & Chairman, Dept of Computer Engineering at YMCA University of Science and Technology, Faridabad. His research interest includes Fuzzy Systems, OOPS, Knowledge Representation and Internet Technologies. He has guided 9 Ph.D thesis and 8 more are in progress with about 175 research publications in International and National journals and conferences. The author of 7 books, is actively engaged in research related to Fuzzy logic, Knowledge based systems, MANETS, Design of crawlers. Besides being member of many BOS and Academic councils, he has been Visiting Professor at JMI, IIIT&M, and I.I.T. Roorkee.
Design of a Framework for Privatized Information Retrieval in Deep Web Scenario

Dilip Kumar Sharma
todilipsharma@rediffmail.com
GLA. Institute of Technology and Management, Mathura, India

A. K. Sharma
YMCA University of Science and Technology, Faridabad, Haryana, India

ABSTRACT

In today's scenario web is working as a backbone for sharing and storing of information. Generally all the privatized information are delivered or distributed on different platforms such as social networking sites, email provider's and blogging sites. All these service providers do not have adequate internal searching facility so it is tough for the individual for searching its privatized contents on these different sites. This paper will not only persuade the requirement of a third party privatized searching services but also proposed a framework for solving this problem. This framework is proposed for QIIIEP [3] server extension. The functioning of proposed framework is divided into two parts. The mechanism of the first part is based on “authenticated crawling” and the mechanism of the second part is based on “user authentication to domain mapping”. The architectural details are simple to implement and can be implemented with minor modifications in ongoing existing infrastructure.

KEYWORDS Deep web, QIIIEP, authenticated crawling, privatized information retrieval.

1. INTRODUCTION

The world wide web is growing very rapidly. It is a challenge to explore all the useful information in such a scenario. Although general search engines like Google and AltaVista crawl and index a large amount of information but they ignore valuable data in text databases that are “hidden” behind search interfaces and whose contents are not directly available for crawling through hyperlinks. A large part of the Web is "hidden" behind search forms and can be indexed only by typing a set of keywords or queries to the forms. These pages are known as the hidden web or the deep web because search engines generally cannot index the deep web pages and consequently do not show them in the results [1][2].

Every Internet user is using Internet for different needs and stores the privatized information in his or her private area on the web. Privatized information retrieval in deep web scenario means to provide desired and precise information to an authenticated user through deep web. This information is stored and distributed on different websites so there exist a possibility of saving time and resources if an authenticated user is provided combined search results about his or her privatized information from different sources, which are stored and distributed on different websites. Searching of specific information in the private area depends on site-specific search functionalities which are sometimes not much adequate for retrieving privatized information particularly with reference to deep web scenario. The other problem with the current web search engine is that there is almost no technique available to combine search results from private areas, stored and distributed on different websites in response to query word submitted to search engine particularly in deep web scenario. The main thrust area in this paper is to propose a framework for privatized information retrieval in deep web scenario for QIIIEP [3] with the facility of combining search results about privatized information from different sources distributed on different websites.

QIIIEP is a query intensive interface information extraction protocol proposed by the same authors for efficient deep web information retrieval. The architectural framework of QIIIEP is divided into various modules in which functioning of one of the module depend upon privatized information retrieval in deep web scenario. There is no appropriate standard available for providing privatized search with the facility of combining search results from different sources, stored and distributed on different websites particularly in deep web scenario. In this paper we have proposed a framework for privatized information
retrieval in deep web scenario with the facility of combining search results from different private sources, stored and distributed on different websites. The functioning of proposed framework is divided into two parts. The mechanism of the first part is based on “authenticated crawling” and the mechanism of the second part is based on “user authentication to domain mapping”.

The objective of this paper is to analyze these proposed mechanisms of privatized information search with the ability of combining search results from different private areas to find their usefulness and limitations particularly in deep web scenario. We believe that proposed work is important to make future contributions in the emerging field of deep web information retrieval, where various applications can utilize it.

The rest of this paper is organized as follows. Related work is discussed in section 2. Functioning of the proposed framework is discussed in section 3. The implementation detail of proposed framework is discussed in section 4 and further the results are presented in section 5 and finally a conclusion is given in section 6 with future scope.

II. RELATED WORK

As the web is expanding day by day and there is enormous amount of data stored and available on the web in present scenario. Numbers of persons who use websites for storing their private information through their various types of personal accounts available on different websites are also increasing. In today’s scenario website are not directly navigated through their web address but search engines are used more and more for referring and exploring the websites. In this paper stress is given to propose a framework for privatized information retrieval in deep web scenario with the additional facility of combining search results from different sources, stored and distributed on different websites. Current search engines have limitations in exploring the privatized information particularly from deep web with facility of combining search results from various sources. A very few attempt is done for providing the facility for combining the search results from different sources about privatized information retrieval particularly in deep web scenario.

Some attempt is done about the search from different repositories by oracle ultra search. Oracle ultra search is based upon oracle text technologies and oracle database. It provides uniform search and location ability over different repositories like oracle database, other compliant database html documents provided by a web server, IMAP mail servers and files on the disk etc. Oracle ultra search consists of a crawler that collects the documents. Crawler can be scheduled for the websites to be search. Documents are stored in the repositories and an index is made by using the collected information. The index is stored in the firewall in a specific oracle database. API is also provided by oracle ultra search for producing content management solution. Oracle ultra search consist of following three main components.

- Oracle ultra search crawler
- Oracle ultra search backend
- Oracle ultra search middle layer

Oracle ultra search crawler is a java application. Its working is handled by an oracle server. During activation period oracle ultra search crawler generates various processor threats. These generated processor threats extract document from different data sources and arrange them according to an index utilizing oracle text. This generated index is used for further querying process. The sources of data from where the documents are extracted may be database table, files, websites, and groups of server portal pages. The oracle ultra search backend can be divided into two parts i.e. oracle text and oracle ultra search repository. Oracle text does the two jobs i.e. indexing of text is done with the help of oracle text and it also provides search ability to indexing and querying process. Whereas oracle ultra search middle layer consist of API, administration tools for oracle ultra search and query application. Oracle ultra search middle layer work as a interface between oracle ultra search crawler and backend [4].

IBM Lotus extended search technology which is a scalable technology based on server does similar attempt in this area. It has the capability to search various parallel multiple contents and data sources and the results are returned against a query in the form of web application or notes.
This technology formerly known as domino extended search issued in 1997 to enhance the capability of domino search to various back ends data source such as oracle, DB2 and MS SQL Server database. IBM lotus extended search consist of various interconnected component. It handles various functions related to search request of client such as verification of data security, ranking of results, required number of search result, interpretation of source language, scaling of server and load balancing [5].

Authenticated crawling is the major issue in the privatized information retrieval in deep web scenario. One of the techniques for authenticated crawling is based on form based authentication (FBA). In form based authentication process, the authentication information is submitted by filling an html form by the user. It is an authentication mechanism for crawling, which uses a web form to enter the authentication information. There is no single universal standard available for FBA and various standards exist for FBA. Some techniques or standards send authentication information to the server and return a cookie in response whereas other techniques are more complex and they involve various redirects to various pages or they need that user get a cookie before supplying information about authentication to a server [6].

Kerberos authentication mechanism gives enhanced security over NTLM. It is the default authentication mechanism for MS Office share point server 2007 web applications. In this technology crawler is not able to crawl the sites, authenticated by Kerberos mechanism if the sites are configured over a non standard port. Any port other than SSL port 443(https) and TCP port 80(http) are non standard ports. Crawling process is affected by polling order of web applications. The crawler starts its process by polling the default zone. If the default zone is authenticated by Kerberos mechanism and it does not use SSL port 443 or TCP port 80, in such situations crawler does not authenticate by utilizing the further zone in the polling order sequence and no content of web application is crawled by the crawler, which means contents are not returned or indexed in the search results against the web query [7].

Dynamic URL tab [8] is a technique, developed for authenticated websites, which allow a site owner to supply yahoo search information with respect to dynamic parameters, appear on the site. Parameter of URL can be used in the private site to do the various jobs such as

- The changing of the content
- Tracking the user session through session ids
- Tracking of the sources which are responsible for providing referrals to the private pages and private sites through source tracker
- Printing of format through format modifier

By going through literature survey, we have concluded that a very few attempts are done on the privatized information retrieval with the facility to combine the search results from different sources, stored and distributed on different websites. So in this paper the main thrust is given on to propose a efficient, secure and authenticated technique to combine privatized search results from different sources, stored and distributed on different websites.

III. PROPOSED WORK

In this paper a framework is proposed for privatized information retrieval in deep web scenario with the facility of combining search results arise from different private sources, stored and distributed on different websites. First of all users supply credentials to the search engine of its own private area distributed on different sites as shown in figure 1.
A. Framework for privatized information retrieval

This framework then uses these credentials to crawl keywords from private area by authenticating on different sites. When the users search the keyword after login on search engine, search engine match this with private keywords of user. If user click on returned link, search engine authenticate it self on specific site and return the original page in the real time. So the step of manual authentication is eliminated in the proposed framework.

As shown in figure 2 every user who want to use privatized information search must be a registered user of deep web search engine. Deep web search engine collect information about the users of different private sites from authentication form as shown in figure 1. After authentication on search engine, user supply its credentials which are used for authentication purposes on different private site at the time of...
crawling. The crawler uses automatic login facility to authenticate itself on different sites and retrieve entire content from user site. After analyzing entire content, it only stores private keywords for the sake of privacy of individuals. When the user submits any query word, the deep web search engine not only matches this keyword with public data but also matches with the private keywords. When the user clicks on any link, deep web search engine returns links in combined form from different private sites. The search engine authenticates itself on that specific site in real time and provides original contents directly to the user by working as a tunnel between the user and http server.

There are two mechanisms that are used in our framework. First mechanism is based on user authentication to domain mapper and second mechanism based on agent based authenticated crawling.

A. User authentication to domain mapper

This module of crawler is responsible for mapping the login credentials provided by users to the crawler with the information provider domain. The main benefit of using this mapper is to overcome the hindrance of information retrieval between result link and information. The crawler uses the mapping information to allow the specific person to receive information contents directly from the domain by automatic login procedure and eliminate the step of separate login for user.

![Figure 3 Framework for user authentication to domain mapper](image)

As shown in figure 3, the user first provides its credentials to the deep web search engine. When the user clicks on any link returned by the search engine against the search, this module virtually logs in itself on the behalf of user to the user’s private site and provides the original contents from that site by working as a proxy in between user and site.

B. Agent for authenticated crawling

This module works when the information in a site hidden by the authentication form. It stores authentication credentials of every domain in its knowledge base situated at crawler which is provided by the individual user. At the time of crawling, it automatically authenticates itself on the domain for crawling the hidden contents. The crawler extracts and store keywords from contents and available to privatized search service due to privacy issue.
The working can be explained in four steps. In the first step, user login on deep web search engine and in the second step user submit credentials of its different private sites. During the third step, the deep web crawler authenticates this submitted information on private site and crawl entire contents from user’s private area. Then it identifies keywords from pages and stores them in the database.

IV. IMPLEMENTATION OF THE PROPOSED FRAMEWORK

The implementation of the proposed framework can be divided into two parts. In the first part, virtual authentication is done and during the second part, crawler acts as a tunnel between user and http server. Both the post and get method of http protocol is considered with session management in authentication process.

A. Implementation of user authentication to domain mapper

The working of user authentication to domain mapper consists of three major steps. In the first step the user login on the search engine and all the search terms evaluated against keywords retrieved from private area and accordingly the link is generated. The ranking of links are based on search engine’s own ranking algorithm overridden by private area keyword ranking. During the second major step, authentication is done on any link event at search engine as a virtual user on user’s site consisting of its private area. In this process, real time authentication is performed with above-mentioned procedure similar to agent for authenticated crawling. The last step responsible for generating appropriate results to be visualized to the user. This is done by extracting contents from user’s private area and sending it to user’s screen. In this process, the search engine works as a virtual tunnel in between user and user’s site. The user authentication to domain mapper is designed for providing exact wanted information to the user. Its working can be divided into three steps.

1. It provide the link list of different private areas to the user during the search process.
2. When the user clicks on a link, it authenticates search engine on the private area site by using user credentials.
3. After authentication, the crawler work as the proxy between the user and the private area site.

The authentication can be done by post or get request depending upon the user’s private area site implementation. Implementation of this module is based on post simulation using httpWebRequest [9] As HttpWebRequest = CType(WebRequest.Create(url), HttpWebRequest).
The code fragment is given below.

```vbnet
Dim requestStream As Stream
Try
    requestStream = httpWebRequest2.GetRequestStream
    Dim sChunks As Stream = httpWebRequest2.GetRequestStream
    sChunks.Write(bBuffer, 0, iBytesRead)
    sChunks.Close()
    requestStream.Close()
    Dim webResponse2 As WebResponse = httpWebRequest2.GetResponse
    Dim stream2 As Stream = webResponse2.GetResponseStream
    Dim reader2 As StreamReader = New StreamReader(stream2)
    Dim response As String = reader2.ReadToEnd
    If response != "" Then
        //Code for checking authentication success and crawling
        End If
    webResponse2.Close()
    httpWebRequest2 = Nothing
    webResponse2 = Nothing
    Catch ex As Exception
End Try
```

When the server responds successful authentication, which can identified by parsing the response data, this module finds the best matching link from its database for the keyword enganged by ranking algorithm and iterate private area until the page having query word were not found. At this moment, search engine starts working as a tunnel [10] between user and authenticated site. This will reduce the process of manual login as well as reduce the manual iteration process of user.

B. Implementation of agent for authenticated crawling

The agent for authenticated crawling is implemented by identifying session management mechanism of server and automating the authentication process. Some sites use a login page which simply send authentication information to the server by post request and return a cookie whereas others are more complex and perform redirections to other pages and require that the requesting party obtain a cookie before sending authentication information to a server.

The simple implementation of HTML Web form that is used to send authentication information to a server (HTML implementation of Login form) is given below:

```html
<form name="loginform" action="logincheck" method="POST">
    Username: <input type="text" name="uname" />
    Password: <input type="password" name="passw" />
    <input type="submit" value="Login" />
    <input type="reset" value="Reset" />
</form>
```

In a simple authentication mechanism, this form sends the username and password to the server and after matching with stored user name and password, server returns either login failed page or returns a cookie (or a set of cookies) to the requesting party. But some servers use more complex authentication process. It consists redirecting the user to the specific web form with cookies to enter the credentials. During some conditions, more than one redirects arise for the user. In such conditions, user submits the authentication form after entering the credentials of login. The browser sends the credentials and the cookie to the action url on the server. A single cookie or multiple numbers of cookies are returned back to the web browser by authentication server. The server then redirects the browser to the requested page.
The mechanism of the framework is given as follows. The agent accesses the root of the web server and obtains a session cookie. The crawler then analyzes the form for form method and action attribute. The crawler then forms a post query to the form action by GET or POST request with pre-supplied credentials. The Http Server matches the incoming data with database for validation. On a successful response the agent stores the cookie generated from the request. During crawling of pages form private area, the crawler attaches these cookies to the page request for each new link crawling.

The experiment has conducted on a machine having Intel Core 2 Duo T5870 @ 2.0 GHz with 1 GB of RAM. This machine was running with Windows 7 OS. Tests are performed using WAMP server equipped with php v. 5.2.6 and mysql v. 5.0.51b. and Microsoft Visual Basic .net 3.5. All of the tests were performed on Firefox 3.5.4. All tests were performed multiple times to minimize measurement errors.

We have tested authentication on some of the email service providers such as www.gmail.com, www.rediffmail.com and www.yahoo.co.in with three user name password pairs. The tables given below show the success rate achieved by this framework on querying different keywords.
As shown in above tables, most of the time users get some links in response to the query but in few cases no link is returned either due to no account of user on that specific site or no matched keyword found for that search string in user's private keyword database. There is a proper error handling procedure implemented for skipping content extraction procedure call if authentication fails which eliminate any captcha submission on multiple failures of authentication.

VI. CONCLUSIONS

The query intensive interface information extraction protocol is proposed for efficient deep web data extraction. Major data of deep web is available behind the authenticated sessions so the proposed framework definitely facilitates deep web information extraction. There is almost no security threat in proposed framework because the proposed framework is based on single sign-on at search engine and it searches multiple private areas through keyword and uses streamline cookies and session management on search engine. Proposed framework is tested our on three different sites i.e. gmail.com, rediffmail.com and yahoo.com. Users are successfully authenticated on these sites and fetched contents and stored to our private keyword database. Now some major sites are providing their own API for user level authentication automation such as Gmail content extraction API. The limitation of proposed framework is that it only stores keywords and does not stores the entire content at search engine so the results definitely depends upon the identification of keyword in contents of pages at the time of crawling from private area. But it enables peculiarity of privacy for the user.
A simple ranking algorithm is used for link ranking, which is based on keyword density and updating time. The future work includes to

REFERENCES


Dilip Kumar Sharma is B.Sc, B.E.(Honors)(CSE), M.Tech.(IT), M.Tech. (CSE) and pursuing Ph.D in Computer Engineering. He is life member of CSI Mumbai, IETE New Delhi, ISTE New Delhi, ISCA Kolkata, SSI and member of CSTA, USA. He has attended 18 short term courses/workshops/seminars organized by various esteemed originations such as IIT Kanpur, IIT Roorkee, IIT Delhi, IIT Khargpur, Microsoft, WIPRO, IBM, TCS etc. He has published 18 research papers in International Journals /Conferences of repute and participated in 18 International/National conferences. Presently he is working as Reader in Department of Computer Science & Engineering (NBA Accredited) at GLA Institute of Technology & Management, Mathura (U.P.) India,(AICTE Approved & Affiliated to U.P. Technical University, Lucknow) since March 2003. His research interests are deep web information retrieval, e-commerce, Digital Watermarking and Software Engineering. He has guided various projects and seminars undertaken by the students of undergraduate/postgraduate.

Prof. A. K. Sharma received his M.Tech. (Computer Sci. & Tech) with Hons. From University of Roorkee design an improved and efficient ranking algorithm to be used for link ranking.


[8] Dynamic URL tab , ‘Dynamic URL tab’, Obtained through the Internet in 2010 from


( Presently I.I.T. Roorkee) in the year 1990 and Ph.D (Fuzzy Expert Systems) from JMI, New Delhi in the year 2000. From July 1992 to April 2002, he served as Assistant Professor and became Professor in Computer Engg. at YMCA Institute of Engineering, Faridabad, Haryana in April 2002. He obtained his second Ph.D. in Information Technology form Indian Institute of Information Technology & Management, Gwalior in 2004. Presently he is working as Dean, Faculty of Engineering and Technology & Head, Dept of Computer Engineering at YMCA University of Science and Technology, Faridabad, India.

His research interest includes Fuzzy Systems, Object Oriented Programming, Knowledge Representation and Internet Technologies. He has guided 9 Ph.D. thesis and 8 more are in progress with about 175 research publications in International and National journals and conferences. The author of 7 books, is actively engaged in research related to Fuzzy logic, Knowledge based systems, MANets, Design of crawlers. Besides being member of many BOS and Academic councils, he has been Visiting Professor at JMI, IIT&M, and I.I.T. Roorkee. He is also project coordinator of Technical Education Quality Improvement Program under World Bank scheme.
A Comparative Analysis of Web Page Ranking Algorithms

Dilip Kumar Sharma
GLA University, Mathura, UP, India
todilipsharma@rediffmail.com

A. K. Sharma
YMCA University of Science and Technology, Faridabad, Haryana, India

todilipsharma@rediffmail.com

Abstract - Web is expending day by day and people generally rely on the search engine to explore the web. In such a scenario it is the duty of service provider to provide proper, relevant and quality information to the internet user against their query submitted to the search engine. It is a challenge for service provider to provide proper, relevant and quality information to the internet user by using the web page contents and hyperlink between the web pages. This paper deals with analysis and comparison of web page ranking algorithms based on various parameters to find out their advantages and limitations for the ranking of the web pages. Based on the analysis of different web page ranking algorithms, a comparative study is done to find out their relative strengths and limitations to find out the further scope of research in web page ranking algorithm.

Keywords Web page ranking, Page Rank, HITS, WCM, WSM, WUM.

I INTRODUCTION

As the volume of information on the internet is increasing day by day so there is a challenge for website owner to provide proper and relevant information to the internet user. Figure 1 [2] shows a working of a typical search engine, which shows the flow graph for a searched query by a web user.

An efficient ranking of query words has a major role in efficient searching for query words. There are various challenges associated with the ranking of web pages such that some web pages are made only for navigation purpose and some pages of the web do not possess the quality of self descriptiveness. For ranking of web pages, several algorithms are proposed in the literatures. The motive behind this paper to analyze the currently important algorithms for ranking of web pages to find out their relative strengths, limitations and provide a future direction for the research in the field of efficient algorithm for ranking of the web pages [1][2]. The remaining part of this paper is organized as follows: Related work is summarized in Section II. A tabular summary is presented in section III, which summarizes the techniques, advantages and limitations of some of the important web page ranking algorithms. Based on the literature analysis, a comparison of some of various web page ranking algorithms is presented in section IV and a conclusion is given in section V.

II RELATED WORK

Web mining is the technique to classify the web pages and internet users by taking into consideration the contents of the page and behavior of internet user in the past. Web mining helps the internet user about the web pages to be viewed in future. Web mining is made of three branches i.e. web content mining (WCM), web structure mining (WSM) and web usage mining (WUM). WCM is responsible for exploring the proper and relevant information from the contents of web. WSM is used to find out the relation between different web pages by processing the structure of web. WUM is responsible for recording the user profile and user behavior inside the log file of the web. The WCM mainly concentrates on the structure of the document whereas WSM explore the structure of the link inside the hyperlink between different documents and classify the pages of web. The number of out links i.e. links from a page and the number of in link i.e. links to a page are very important parameter in the area of web mining. The popularity of the web page is generally measured by the fact that a particular page should be referred by a large number of other pages and the importance of web pages may be adjudged by a large number of out links contained by a page. So WSM becomes a very important area to be researched in the field of web mining [1][2][3][4][5]. Figure 2 shows the general classification of web mining [2].
Two graph based page ranking algorithms i.e. google Page Rank proposed by Brin and Page in 1998 and Kleinberg’s hypertext induced topic selection (HITS) algorithm proposed by Kleinberg in 1998 are used successfully and traditionally in the area of web structure mining. Both of these algorithms give equal weights to all links for deciding the rank score.

**Page Rank Algorithm**

Page Rank algorithm is the most commonly used algorithm for ranking the various pages. Working of the Page Rank algorithm depends upon link structure of the web pages. The Page Rank algorithm is based on the concepts that if a page contains important links towards it then the links of this page towards the other page are also to be considered as important pages. The Page Rank considers the back link in deciding the rank score. If the addition of the all the ranks of the back links is large then the page then it is provided a large rank [1][6][7][8]. A simplified version of PageRank is given by:

$$PR(u) = \frac{\sum_{v \in B_u} PR(v)}{L(v)}$$

Where the PageRank value for a web page u is dependent on the PageRank values for each web page v out of the set B_u (this set contains all pages linking to web page u), divided by the number L(v) of links from page v.

An example of back link is shown in figure 3 below. U is the back link of V & W and V & W are the back links of X.

![Figure 3: Illustration of back links](image)

**HITS Algorithm**

HITS algorithm ranks the web page by processing in links and out links of the web pages. In this algorithm a web page is named as authority if the web page is pointed by many hyper links and a web page is named as HUB if the page point to various hyperlinks. An Illustration of HUB and authority are shown in figure 4.

![Figure 4: Illustration of Hub and Authorities](image)

HITS is technically, a link based algorithm. In HITS [9] algorithm, ranking of the web page is decided by analyzing their textual contents against a given query. After collection of the web pages, the HITS algorithm concentrates on the structure of the web only, neglecting their textual contents. Original HITS algorithm has some problems which are given below:

(i) High rank value is given to some popular website that is not highly relevant to the given query.

(ii) Drift of the topic occurs when the hub has multiple topics as equivalent weights are given to all of the outlinks of a hub page. Figure 5 shows an Illustration of HITS process.

![Figure 5: Illustration of HITS process](image)

To minimize the problem of the original HITS algorithm, a clever algorithm is proposed by reference [10]. Clever algorithm is the modification of standard original HITS algorithm. This algorithm provides a weight value to every link depending on the terms of queries and endpoints of the link. An anchor tag is combined to decide the weights to the link and a large hub is broken down into smaller parts so that every hub page is concentrated only on one topic.

Another limitation of standard HITS algorithm is that it assumes equal weights to all the links pointing to a webpage and it fails to identify the facts that some links may be more important than the other. To resolve this problem, a probabilistic analogue of the HITS (PHITS) algorithm is proposed by reference [11]. A probabilistic explanation of relationship of term document is provided by PHITS. It is able to identify authoritative document as claimed by the author. PHITS gives better results as compared to original HITS algorithm. Other difference between PHITS and standard HITS is that PHITS can estimate the probabilities of authorities compared to standard HITS algorithm, which can provide only the scalar magnitude of authority [1].

**Weighted Page Rank Algorithm**

Weighted Page Rank [1] Algorithm is proposed by Wenpu Xing and Ali Ghorbani. Weighted page rank algorithm (WPR) is the modification of the original page rank algorithm. WPR decides the rank score based on the popularity of the pages by taking into consideration the importance of both the in-links and out-links of the pages. This algorithm provides high value of rank to the more popular pages and does not equally divide the rank of a page among its out-link pages. Every out-link page is given a rank value based on its popularity. Popularity of a page is decided by observing its number of in links and out links. Simulation of WPR is done using the website of Saint Thomas University and simulation results show that WPR algorithm finds larger number of relevant pages compared to standard page rank algorithm. As suggested by the author, the performance of WPR is to be tested by using different
Websites and future work include to calculate the rank score by utilizing more than one level of reference page list and increasing the number of human user to classify the web pages.

**Weighted Links Rank Algorithm**

A modification of the standard page rank algorithm is given by Ricardo Baeza-Yates and Emilio Davis [13] named as weighted links rank (WLRank). This algorithm provides weight value to the link based on three parameters i.e. length of the anchor text, tag in which the link is contained and relative position in the page. Simulation results show that the results of the search engine are improved using weighted links. The length of anchor text seems to be the best attributes in this algorithm. Relative position, which reveal that physical position does not always in synchronization with logical position is not so result oriented. Future work in this algorithm includes, tuning of the weight factor of every term for further evolution.

**EigenRumor Algorithm**

As the number of blogging sites is increasing day by day, there is a challenge for service provider to provide good blogs to the users. Page rank and HITS are very promising in providing the rank value to the blogs but some limitations arise, if these two algorithms are applied directly to the blogs The rank scores of blog entries as decided by the page rank algorithm is often very low so it cannot allow blog entries to be provided by rank score according to their importance. To resolve these limitations, a EigenRumor algorithm [14] is proposed for ranking the blogs. This algorithm provides a rank score to every blog by weighting the scores of the hub and authority of the bloggers depending on the calculation of eigen vector.

**Distance Rank Algorithm**

An intelligent ranking algorithm named as distance rank is proposed by Ali Mohammad Zareh Bidoki and Nasser Yazdani [15]. It is based on reinforcement learning algorithm. In this algorithm, the distance between pages is considered as a punishment factor. In this algorithm the ranking is done on the basis of the shortest logarithmic distance between two pages and ranked according to them. The Advantage of this algorithm is that it can find pages with high quality and more quickly with the use of distance based solution. The Limitation of this algorithm is that the crawler should perform a large calculation to calculate the distance vector, if new page is inserted between the two pages.

**Time Rank Algorithm**

An algorithm named as TimeRank, for improving the rank score by using the visit time of the web page is proposed by H Jiang et al.[16] Authors have measured the visit time of the page after applying original and improved methods of web page rank algorithm to know about the degree of importance to the users. This algorithm utilizes the time factor to increase the accuracy of the web page ranking. Due to the methodology used in this algorithm, it can be assumed to be a combination of content and link structure. The results of this algorithm are very satisfactory and in agreement with the applied theory for developing the algorithm.

**TagRank Algorithm**

A novel algorithm named as TagRank [17] for ranking the web page based on social annotations is proposed by Shen Jie, Chen Chen, Zhang Hui, Sun Rong-Shuang, Zhu Yan and He Kun. This algorithm calculates the heat of the tags by using time factor of the new data source tag and the annotations behavior of the web users. This algorithm provides a better authentication method for ranking the web pages. The results of this algorithm are very accurate and this algorithm index new information resources in a better way. Future work in this direction can be to utilize co-occurrence factor of the tag to determine weight of the tag and this algorithm can also be improved by using semantic relationship among the co-occurrence tags.

**Relation Based Algorithm**

Fabrizio Lamberti, Andrea Sanna and Claudio Demartini [18] proposed a relation based algorithm for the ranking the web page for semantic web search engine. Various search engines are presented for better information extraction by using relations of the semantic web. This algorithm proposes a relation based page rank algorithm for semantic web search engine that depends on information extracted from the queries of the users and annotated resources. Results are very encouraging on the parameter of time complexity and accuracy. Further improvement in this algorithm can be the increased use of scalability into future semantic web repositories.

**Query Dependent Ranking Algorithm**

Lian- Wang Lee, Jung- Yi Jiang, ChunDer Wu and Shie-Jue Lee [19] have presented a query dependent ranking algorithm for search engine. In this approach a simple similarity measure algorithm is used to measure the similarities between the queries. A single model for ranking is made for every training query with corresponding document. Whenever a query arises, then documents are extracted and ranked depending on the rank scores calculated by the ranking model. The ranking model in this algorithm is the combination of various models of the similar training queries. Experimental results show that query dependent ranking algorithm is better than other algorithms.

**Ranking and Suggestive Algorithm**

M Vojnovic et al. [20] have proposed a ranking and suggestive algorithm for popular items based on user feedback. User feedback is measured by using a set of suggested items. Items are selected depending on the preferences of the user. The aim of this technique is to measure the correct ranking of the items based on the actual and unbiased popularity. Proposed algorithm has various techniques for suggesting the search query. This algorithm can also be used for providing tag suggestion for social tagging system. In this algorithm various techniques for ranking and suggesting popular items are studied and results are provided based on their performance. Results of this
algorithm demonstrate that randomized update and light weight rules having no special configurations provide better accuracy.

Comparison and Score Based Algorithm
NL Bhamidipati et al. [21] have proposed a more common approach whereby the scoring scheme may be perceived to be dissimilar if they induce identical ranking. In this algorithm a metric has been proposed to compare score vectors and the similarity and dissimilarity are measured on the basis of score fusion. Experimental results are in agreement with the theory applied and results demonstrate the various applications of the metric used in the theory applied for the proposed algorithm.

Algorithm for Query Processing in Uncertain Databases
Xiang Lian and Lei Chen [22] have proposed an algorithm for ranked query processing in uncertain databases. Uncertain database management is used in various areas such as tracking of mobile objects and monitoring of sensor data. Due to intrinsic difference between certain and uncertain data. To remove these limitations authors have proposed a novel algorithm. Uncertain database are not exact points and generally occurs within a limited region. Existing algorithms for rank query processing are generally developed for exact or certain data but they cannot be applied directly to uncertain database due to accelerate the probabilistic rank query with monotonic preference functions over the uncertain databases. Authors have proposed two effective techniques named as probabilistic and spatial to reduce the PRank search space. Exhaustive experiment results show that proposed algorithm is very effective and efficient with respect to number of PRank candidates to be refined and wall clock time.

Ranking of Journal based on Page Rank & HITS Algorithm
Su Cheng, Pan YunTao, Yuan JunPeng,Guo Hong, Yu ZhengLu and u ZhiYu [23] have used page rank and HITS algorithm for ranking of the journal and also compared ISI impact factor with page rank and HITS algorithm for ranking the journal. The advantages, limitations and scope of various algorithms used are discussed for ranking the journal. Impact factor is a very popular for ranking the journal but it has intrinsic limitations that the ranking is based on counting the in degrees of the nodes in the network and it does not consider the impact of prestige of the journal in which the citations are present. To minimize these limitations authors have used page rank and HITS algorithm for ranking the journal. Fundamentally the ranking of the journal is very similar to the ranking of the web page. So page rank and HITS algorithm can be used for ranking of the journal.

III SUMMARY OF VARIOUS WEB PAGE RANKING ALGORITHMS
By going through the literature analysis of some of the important web page ranking algorithms, It is concluded that each algorithm has some relative strengths and limitations. A tabular summary is given below in table 1, which summarizes the techniques, advantages and limitations of some of important web page ranking algorithms.

<table>
<thead>
<tr>
<th>Author/Year</th>
<th>Technique</th>
<th>Advantages</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>S. Brin et al. 1998</td>
<td>Graph based algorithm based on link structure of web pages. Consider the back links in the rank calculations.</td>
<td>Rank is calculated on the basis of the importance of pages.</td>
<td>Results are computed at the indexing time not at the query time.</td>
</tr>
<tr>
<td>Jon Kleinberg, 1998</td>
<td>Rank is calculated by computing hub and authorities score of the pages in order of their relevance.</td>
<td>Returned pages have high relevancy and importance.</td>
<td>With less efficiency and problem of topic drift.</td>
</tr>
<tr>
<td>Sung Jin Kim et al. 2002</td>
<td>This algorithm probabilistically estimates that clear semantics and the identified authoritative documents correspond better to human intuition [12].</td>
<td>Well defined semantics with clear interpretation. Efficiently provide answer to quantitative bibliometric questions.</td>
<td>Priors should be decided on the number of factors to model. Trades computational expense for the risk of getting stuck in local maxima.</td>
</tr>
<tr>
<td>Wenpu Xing et al. 2004</td>
<td>Based on the calculation of the weight of the page with the consideration of the outgoing links, incoming links and title tag of the page at the time of searching.</td>
<td>It gives higher accuracy in terms of ranking because it uses the content of the pages.</td>
<td>It is based only on the popularity of the web page.</td>
</tr>
<tr>
<td>Ricardo BaezaYates et al. 2004</td>
<td>This algorithm ranks the page by providing different weights based on three attributes i.e. relative position in page, tag where link is contained &amp; length of anchor text.</td>
<td>It has less efficiency with reference to precision of the search engine.</td>
<td>Relative position was not so effective, indicating that the logical position not always matches the physical position</td>
</tr>
<tr>
<td>Ko Fujimura et al. 2005</td>
<td>Use of the adjacency matrix, constructed from agent to object link not by page to page link. Three vectors i.e. hub, authority and reputation are needed for score calculation of the blog.</td>
<td>Useful for ranking of blog as well as web pages because input and output links are not considered in the algorithm.</td>
<td>Specifically suited for blog ranking.</td>
</tr>
<tr>
<td>Ali Mohammad Zareh Bidoki et al. 2007</td>
<td>Based on reinforcement learning which consider the logarithmic distance between the pages.</td>
<td>Algorithm consider real user by which pages can be found very quickly with high quality.</td>
<td>A large calculation for distance vector is needed, if new page inserted between the two pages.</td>
</tr>
</tbody>
</table>
IV COMPARISON OF VARIOUS WEB PAGE RANKING ALGORITHMS

Based on the literature analysis, a comparison of some of various web page ranking algorithms is shown in table 2 and in table 3. Comparison is done on the basis of some parameters such as main technique use, methodology, input parameter, relevancy, quality of results, importance and limitations.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Page Rank</th>
<th>HITS</th>
<th>Weighted Page Rank</th>
<th>Eigen Rumor</th>
<th>Web Page Ranking using Link Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Technique</td>
<td>Web Structure Mining</td>
<td>Web Structure Mining, Web Content Mining</td>
<td>Web Structure Mining</td>
<td>Web Content Mining</td>
<td>Web Structure Mining, Web Content Mining</td>
</tr>
<tr>
<td>Methodology</td>
<td>This algorithm computes the score for pages at the time of indexing of the pages.</td>
<td>It computes the hubs and authority of the relevant pages. It relevant as well as important page as the result.</td>
<td>Weight of web page is calculated on the basis of input and outgoing links and on the basis of weight the importance of page is decided.</td>
<td>Eigen rumor use the adjacency matrix, which is constructed from agent to object link not page to page link.</td>
<td>It gives different weight to web links based on 3 attributes: Relative position in page, tag where link is contained, length of anchor text.</td>
</tr>
<tr>
<td>Input Parameter</td>
<td>Back links</td>
<td>Content, Back and Forward links</td>
<td>Back links and Forward links.</td>
<td>Agent/Object</td>
<td>Content, Back and Forward links</td>
</tr>
<tr>
<td>Relevancy</td>
<td>Less (this algo. rank the pages on the indexing time)</td>
<td>More (this algo. Uses the hyperlink so according to Henzinger, 2001 it will give good results and also consider the content of the page)</td>
<td>Less as ranking is based on the calculation of weight of the web page at the time of indexing.</td>
<td>High for Blog so it is mainly used for blog ranking.</td>
<td>more (it consider the relative position of the pages )</td>
</tr>
<tr>
<td>Quality of results</td>
<td>Medium</td>
<td>Less than PR</td>
<td>Higher than PR</td>
<td>Higher than PR and HITS</td>
<td>Medium</td>
</tr>
<tr>
<td>Importance</td>
<td>High. Back links are considered.</td>
<td>Moderate. Hub &amp; authorities scores are utilized.</td>
<td>High. The pages are sorted according to the importance.</td>
<td>High for blog ranking.</td>
<td>Not specifically quoted.</td>
</tr>
<tr>
<td>Limitation</td>
<td>Results come at the time of indexing and not at the query time.</td>
<td>Topic drift and efficiency problem</td>
<td>Relevancy is ignored.</td>
<td>It is most specifically used for blog ranking not for web page ranking as other ranking like page rank, HITS.</td>
<td>Relative position was not so effective, indicating that the logical position not always matches the physical position.</td>
</tr>
</tbody>
</table>

Table 2 Comparison of various web page ranking algorithms
Table 3 Comparison of various web page ranking algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Distance Rank</th>
<th>Time Rank</th>
<th>Tag Rank</th>
<th>Relational Based Page Rank</th>
<th>Query Dependent Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Technique</td>
<td>Web Structure Mining</td>
<td>Web Usages Mining</td>
<td>Web Content Mining</td>
<td>Web Structure Mining</td>
<td>Web Content Mining</td>
</tr>
<tr>
<td>Methodology</td>
<td>Based on reinforcement learning which consider the</td>
<td>In this algorithm the</td>
<td>Visitor time is used for</td>
<td>A semantic search engine</td>
<td>This paper proposed the</td>
</tr>
<tr>
<td></td>
<td>logarithmic distance between the pages.</td>
<td>visiting time is added</td>
<td>ranking. Use of sequential</td>
<td>would take into account</td>
<td>construction of the rank</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to the computational</td>
<td>clicking for sequence vector</td>
<td>keywords and would return</td>
<td>model by combining the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>score of the original</td>
<td>calculation with the uses of</td>
<td>page only if both keywords</td>
<td>results of similar type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>page rank of that page.</td>
<td>random surfing model.</td>
<td>are present within the page</td>
<td>queries.</td>
</tr>
<tr>
<td>Input Parameter</td>
<td>Forward links</td>
<td>Original Page Rank</td>
<td>Popular tags and related</td>
<td>High as it is keyword</td>
<td>High (because the model</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and Sever Log</td>
<td>bookmarks</td>
<td>based algorithm so it only</td>
<td>is constructed from the</td>
</tr>
<tr>
<td>Relevancy</td>
<td>Moderate due to the use of the hyperlinks.</td>
<td>High due to the</td>
<td>Less as it uses the keyword</td>
<td>returns the result if the</td>
<td>training queries.</td>
</tr>
<tr>
<td>Quality of results</td>
<td>High. It is based on distance between the pages.</td>
<td>updation of the</td>
<td>entered by the user and match</td>
<td>keyword entered by the</td>
<td>High</td>
</tr>
<tr>
<td>Importance</td>
<td>High, Consideration of the most recently</td>
<td>original rank</td>
<td>with the page title.</td>
<td>user match with the page.</td>
<td>High</td>
</tr>
<tr>
<td>Limitation</td>
<td>If new page inserted between two pages then the</td>
<td>visited pages.</td>
<td>It is comparison based</td>
<td>In this ranking algorithm</td>
<td>High number of</td>
</tr>
<tr>
<td></td>
<td>crawler should perform a large calculation to</td>
<td></td>
<td>approach so it requires</td>
<td>every page is to be</td>
<td>characteristics are used</td>
</tr>
<tr>
<td></td>
<td>calculate the distance vector.</td>
<td></td>
<td>more site as input.</td>
<td>annotated with respect to</td>
<td>to calculate the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>some ontology, which is</td>
<td>similarity.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>the very tough task.</td>
<td></td>
</tr>
</tbody>
</table>

V CONCLUSION

Based on the algorithm used, the ranking algorithm provides a definite rank to resultant web pages. A typical search engine should use web page ranking techniques based on the specific needs of the users. After going through exhaustive analysis of algorithms for ranking of web pages against the various parameters such as methodology, input parameters, relevancy of results and importance of the results, it is concluded that existing techniques have limitations particularly in terms of time response, accuracy of results, importance of the results and relevancy of results. An efficient web page ranking algorithm should meet out these challenges efficiently with compatibility with global standards of web technology.

REFERENCES


About the Authors
Dilip Kumar Sharma is B.Sc, B.E.(CSE), M.Tech.(IT), M.Tech. (CSE) and pursuing Ph.D in Computer Engineering. He is life member of CSI, IETE, ISTE., ISCA, SSI and member of CSTA, USA. He has attended 21 short term courses/workshops/seminars organized by various esteemed originations. He has published 22 research papers in International Journals /Conferences of repute and participated in 18 International/National conferences. Presently he is working as Reader in Department of Computer Science, IET at GLA University, Mathura, U.P. since March 2003 and he is also CSI Student branch Coordinator. His research interests are deep web information retrieval, Digital Watermarking and Software Engineering. He has guided various projects and seminars undertaken by the students of undergraduate/postgraduate.

Prof. A. K. Sharma received his M.Tech. (CST) with Hons. from University of Roorkee (Presently I.I.T. Roorkee) and Ph.D (Fuzzy Expert Systems) from JMI, New Delhi and he obtained his second Ph.D. in Information Technology form IITM, Gwalior in 2004. Presently he is working as the Dean, Faculty of Engineering and Technology & Chairman, Dept of Computer Engineering at YMCA University of Science and Technology, Faridabad. His research interest includes Fuzzy Systems, OOPS, Knowledge Representation and Internet Technologies. He has guided 9 Ph.D thesis and 8 more are in progress with about 175 research publications in International and National journals and conferences. The author of 7 books, is actively engaged in research related to Fuzzy logic, Knowledge based systems, MANETS, Design of crawlers. Besides being member of many BOS and Academic councils, he has been Visiting Professor at JMI, IIT&M, and I.I.T. Roorkee.
A Novel Ranking Algorithm of Query Words Stored in QIIIEP Server

Dilip Kumar Sharma
GLA University, Mathura, UP, India

A. K. Sharma
YMCA University of Science and Technology, Faridabad, Haryana, India

Abstract:

This paper proposes a novel algorithm for the ranking of query words stored in QIIIEP server which are used for posting the query to extract the contents from deep web (Sharma and Sharma, 2009). These words can be collected from either by auto query words extraction module or submitted by web master of third party sites. This paper analyze different existing algorithms for ranking of query words and suggest an improved algorithm for the same by including newer parameters for ranking of query words. An elaborate analysis is carried on the concept of query words ranking so as to come up with an improved algorithm with enhanced efficiency and one in conformance with the global standards. Proposed algorithm analyzes the context of web page with respect to the supplied keywords and frequency of simultaneous occurrence for same keyword on surface web to assign a numerical weighting to each query word with the purpose of "measuring" its relative importance within the set.

Keywords: QIIIEP, query word ranking, word clustering, density measure, term weighting, deep web.

1 INTRODUCTION

Ranking of a query word has a very significant role in providing authentic and useful information to the web user. Same concept is applicable in the case of deep web. A deep web is a part of World Wide Web, where information is generated on the basis of supplied query. The QIIIEP (query intensive interface information extraction protocol) is a protocol for crawling contents from deep web for search engine (Sharma and Sharma, 2009). A query word ranking algorithm is used for ranking the query words stored on QIIIEP server so that lesser post request has to be generated for crawling the deep web. It should extract out the authentic information against a specific query interface. The deep web (Bergman, 2001) has higher complexity compared to traditional hyperlink web pages as reflected by higher number of ontology modeling languages. Bharat and Broder (Bharat and Broder, 1998) proposed a technique for handing random query to a set of search engine for calculating the relative size and overlapping of their indices. Barbosa and Freire (Barbosa and Freire, 2004) analyze the technique for generating multi keywords queries. A very large fraction of document collection is returned by these multi keywords queries. Proposed work differs from the previous in respect that it provides a framework for analyzing the process of ranking query word for a search interface and examines the results to find out its effectiveness and limitations compared to the methods presented in the previous work. The remaining part of paper is organized as: Related work is discussed in section II. A novel algorithm of query word ranking is proposed in section III. Experimental results are analyzed in section IV. Finally the conclusion is presented in section V.

2 RELATED WORK

One of the complex tasks in ranking of query words is to find out the relevance of query words with query interface. Some of the important algorithmic models for ranking the query word column are discussed below.

2.1 Density Measure Technique

A user try to find a certain degree of detail in the representation of knowledge related to the concept during the searching for the associated concept. This process may involve to find the details of the specification of the concept
to be searched. The example of the specification for this purpose is the number of siblings etc. Density measure techniques consider all of these specifications in its searching process. Density measure is proposed to approximate the representational density or information content of classes and on the basis of this, it approximates the level of knowledge detail. Density measure technique has limitations with respect to the numbers of siblings, subclasses, super-classes, and direct relations. It does not consider the number of instances from this measure because it may deviate the results with reference to populated ontologies, which may not be true reflection of the quality of the schema itself (Alani and Brewster, 2005) (Alani and Brewster, 2006) (Alani, Brewster, and Shadbolt, 2006). Advantage of density measure is that it improves the quality of the information content related to the concept to be searched. Limitation of the density measure is that in some cases irrelevant words are provided with good density measure in the document.

2.2 Term Weighting Technique

Many techniques for term weighting are proposed in the literature (Jones, 1972) (Robertson and Jones, 1976) (Robertson et al., 1999) (Salton and Buckley, 1988) (Singhal et al., 1999) (Singhal, 2001). Term Weighting techniques are proposed using the probabilities models so the term weighting techniques highly rely on accurate calculation of various probabilities. Some of the term weighting techniques are developed using the vector space model and generally these techniques are the outcome of experiments based on the large scale experiment with the system. In both of these, term weighting techniques which are based on probabilistic and vector space model, three main parameters are used in the final term weight formulation. These parameters are document length, document frequency and term frequency. One of the examples of term weighting based on document score is given below (Singhal, 2001).

\[
\text{tf} \text{ is the term’s frequency in document}
\]
\[
\text{qtf} \text{ is the term’s frequency in query}
\]
\[
N \text{ is the total number of documents in the collection}
\]
\[
df \text{ is the number of documents that contain the term}
\]
\[
dl \text{ is the document length (in byes), and}
\]
\[
\text{avdl is the average document length}
\]

Okapi weighting based document score: (Robertson and Jones, 1976)

\[
\sum_{t \in Q,D} \ln \frac{N - df + 0.5}{df + 0.5} \cdot \frac{(k_1 + 1)tf}{k_1(1 - b) + b \frac{dl}{avdl} + tf} \cdot \frac{(k_3 + 1)qtf}{k_3 + qtf}
\]

\[
k_1 \text{ (between 1.0-2.0), } b \text{ (usually 0.75), and } k_3 \text { (between 0-1000) are constants.}
\]

Pivoted normalization weighting based document score (Salton and Buckley, 1988).

\[
\sum_{t \in Q,D} \frac{1 + \ln(1 + \ln(tf))}{(1 - s) + s \frac{df}{avdl}} \cdot \text{qtf} \cdot \ln \frac{N + 1}{df}
\]

s is a constant (usually 0.20).

Advantage of this technique is that it does not respond to the repeated words in the documents and it provides better weighting matrix. Limitation of this technique are that lengthy document get high score due to presence of large number of words and words with multiple repetitions and accuracy of this method highly dependent on the estimation of probability.

2.3 Adaptive Technique

Adaptive technique (Ntoulas et al., 2004) (Ntoulas et al., 2005) recognize the keyword that is most probabilistic to return the large number of document by analyzing the documents returned against the previous query submitted to the deep web database. One can extract out the most important query using this process by repeating process. For extracting out the most significant query, this process includes to calculate the number of new documents to be downloaded if it issues the query qi as the next query. Assuming that issued queries ranges from q1 to qi, then one have to calculate P(q1 to qi), for each potential next counting of how many times qi appears in the pages from q1 to qi. That means one have to calculate only P(qi) to estimate P(q1 to qi). In this process, efficiency is calculated by the following relation:
Efficiency \( (q_i) = \frac{P_{\text{new}}(q_i)}{\text{Cost}(q_i)} \)

Here, \( P_{\text{new}}(q_i) \) represents the amount of new documents returned for \( q_i \) (the pages that have not been returned for previous queries). \( \text{Cost} (q_i) \) corresponds to the cost of issuing the query \( q_i \). Advantage of adaptive technique is that this algorithm extracts new terms and keywords from the downloaded contents and it results in the large number of documents using a lesser number of queries. Limitation of this technique as that it identifies the keywords related to the topic of the website in response to the specific topic.

### 2.4 Generic Algorithm Based Technique

The generic algorithm for ranking the query word based on the analysis of a generic document quantity which is collected, say from the web. It calculates the generic frequency distribution of each keyword. Based on this generic distribution, results are derived by issuing the most frequent keyword to the deep web database. Further it is required to continue to the second-most frequent keyword and repeat this process until all downloaded resources are exhausted. The aim behind this process is that the frequent keywords in a generic corpus will also be frequent in the deep web database, which returns various matching documents (Ntoulas et al., 2004) (Ntoulas et al., 2005).

Advantage of this technique is that keywords are selected with reference to their decreasing frequency in order to their decreasing frequency in the document. Limitation of this technique is that performance of the generic algorithm degrades compared to the adaptive technique in the case of sets with specific topic.

By going through literature review, it can be concluded that a ranking can be improved by including newer and more parameters for ranking of query words. In the next section, a novel technique for ranking is presented with special reference to QIIIEP server.

### 3 PROPOSED QUERY WORD RANKING ALGORITHM

In proposed algorithm i.e. randomized probabilistic selection algorithm, the ranking is based on the relevance of context of web page with respect to the supplied keywords and assigns a numerical weighting to each query word with the purpose of measuring its relative importance within the set. Context for initial grouping of query words is used. Let the number of query words is \( n \) for filling any element in query interface then the group is formed on the basis of category \( c \) in which the word belongs. The model uses random selection strategy for choosing one query word from one group. After analyzing the generated results, the priority of group will be identified by assigning a rank for each group of query word. The relevance is computed in three steps i.e. (1) Root words are created for query terms (2) In second step the identification of context is takes place. (3) Relevance weighting is conducted for a number of parameters (Donahue, 2009).

\[
\sum n_c = n \quad \text{Where } n_c \text{ denotes the number of query words in each category.} \\
S_i = \text{Rand} (x, (p>n_c)) \quad \text{Where } x \text{ is seed, } p \text{ is element of group and } S_i \text{ is initial selection.} \\
S_{i+1} = \text{Rand} (x, (p>n_p)) \quad \text{Where } n_p \text{ groups having high priority derived from context seed and } S_{i+1} \text{ is next selection.}
\]

The flow graph of query word selection algorithm is shown in figure 1.
Randomized probabilistic selection algorithm is used for selecting the query word after analyzing the exact search term and stemmed search term. A random number is used as an arbitrary input to channelize its behavior for producing desired result. The priority identification assimilated with various quality measures, which have been contemplated out in order to improve the results. All these measures tend to affect the ranking results in real time.

Algorithm for randomized probabilistic selection and priority generation, $R_P_{SR}(W,L)$

Input:
- $W$ // List of Query Words
- $L$ // Link of Query interface

Assumptions:
1. Query q is a single word of English language or numerals or a combination of both.
2. QIIIEP server must have information of Query interface and the contents of web document must be in HTML.

Output: ubiquitous
A list of words arranged by priority

1. Identify context of query interface through link $L$.
2. Divide list of query words in groups based of category generated on the basis of context derived from query interface link.
3. Select random population of $S_i$ from groups containing $n_c$ query words from equation 2.
4. Evaluate the fitness $f(x)$ by queering and result analysis of each query word $p$ in the population
5. Repeat following steps to assign priority for each group until every word is used for query.
   5.1 Select those groups that produce better results and assign a higher priority according to their fitness (the better is the fitness, the better are the chances to be selected)
   5.2 Select more query words form high priority group and randomly select the group with low priority and select few query words from them.
   5.3 Evaluate the fitness $f(x)$ by queering and analyze the result of each query word $p$ in the population.

Flow graph of the proposed algorithm can be broadly divided into three sub processes that are, Root-words stemming generation, Context Identification & clustering and fitness evaluation.

3.1 Root-words stemming generation
Words are converted to their base or root words through the stemming technique. In a very fundamental way, it tries to find singular term with respect to plural form of search term, which is in plural form and vice-versa. This process
can be implemented by eliminating “s” or “es” from the search words in English language. In most of the situations search terms are stemmed before starting of the actual search. Frequently it is required to mention that one is going to execute exact search or a stemmed search. There are various available techniques based on linguistic analysis to extract the root form of a word. Examples of such technique are porter algorithm and trivial algorithm which is basically based on techniques of affix removal. The root word stemming is required because deep web interface generally have singular entry label. 1. Advantages of this technique are that it reduces the redundancy of the word having different modifications (such as –s, es, ing in the suffix) in the QIIIEP server. It helps in the formation of good cluster and efficiency of searching is fast because there are fewer words to be search in the QIIIEP server after root word stemming.

3.2 Context identification and clustering

The proposed methods exploit the techniques of context identification. A word context mainly made of text surrounding the given words, hyperlink documents, documents title and so on. The characteristic of the context is very important because it help to measure the relevance of the search term. Term selection is one of the methods of extracting the terms that can be used as a context for the query interface. In order to quantify the importance of a term t in a surface web page collection, a weight is provided to the term. After calculating the weight of a term, these words are stored according to their weight in context table. By experimenting of isuvidha.com, context words shop, market, category, purchase and online having high weight are collected. In next section of this paper, the cluster occurrence of given term t is found with stored query word by using surface web search engine. Context identification helps in finding the relevance of the search term in the document and clustering help in making the groups of relevant words. Let us take an example. Assume a set of words is stored in query word knowledge base for specific element having Label Product i.e. printer, scanner, mp3 player, speaker, and monitor. In this case, following three word groups are required: i.e.

<table>
<thead>
<tr>
<th>Context words</th>
<th>Label</th>
<th>Query word</th>
</tr>
</thead>
<tbody>
<tr>
<td>shop</td>
<td>Product</td>
<td>Printer</td>
</tr>
<tr>
<td>market</td>
<td></td>
<td>Scanner</td>
</tr>
<tr>
<td>category</td>
<td></td>
<td>mp3 player</td>
</tr>
<tr>
<td>purchase</td>
<td></td>
<td>Speaker</td>
</tr>
<tr>
<td>online</td>
<td></td>
<td>Monitor</td>
</tr>
</tbody>
</table>

In this algorithm, the chi-square ($\chi^2$) value is used. The chi-square value is calculated as follows: Denote the number of pages containing both $w_1, w_2$ and $w_3$ as $a$.
Thereby, the expected frequency of ($w_1, w_2$) is $(a+c)(a+b)/N$. Eventually, chi-square is calculated as follows (Manning and Schutze, 2002) (Matsuo et al, 2006).

$$b' = \sum_{w \in W, w \neq w_2} f_{w_1, w}, \quad c' = \sum_{w_1 \in W, w \neq w_2} f_{w_2, w},$$
$$d' = \sum_{w_1, w \in W, w \neq w_2} f_{w_1, w}, \quad N' = \sum_{w_1, w \in W} f_{w_1, w}.$$

Where W represents a given set of words. Then chi-square (within the word list W) is defined as

$$X^2w(w_1, w_2) = \frac{N' \times (a \times d' - b' \times c')^2}{(a + b') \times (a + c') \times (b' + d') \times (c' + d')}$$

3.3 Fitness evaluation

In the proposed algorithm occurrence of search term within search result is analyzed and consequently weights are assigned for different factors. Occurrence of stem terms and exact terms is also taken into consideration. Relative weights can be assigned to different result fields. Higher weights can be assigned to the collection with higher importance and recent results. Priority of the groups is assigned for the ranking of the groups according to the
relative weight. The parameters such as density of search term, position of search term, ordinality of search term and proximity of search term are also taken into consideration as per given algorithms.

3.3.1 To measure the density of search term – Density of keywords is measured by counting the occurrence of particular keywords in contents of a page. The density of search term gives an indication of about the probability of finding the relevant contents with respect to given query.

The algorithm for measuring the density of search term is given below.

\[
D_{S\_T}(k,q)
\]

1. String str <- k.data
2. String token[ ] <- parse(str,#)
3. String qstr<-q.data
4. String qtoken[ ] <- parse(qstr,#)
5. while (qtoken[j]==!NULL)
6. while (token[i]==!NULL)
   a. if ( token [i] = q[j])
      c <- c + 1
   b. i  <- i + 1
   end while
   j<-j+1
end while
7. return c

3.3.2 To measure the position of the search term – Position of search term refer to that, where the search terms is located within particular field. Special considerations are given that a search term is located at the first word position, last word position or relative position between first and last words.

The algorithm for measuring the position of search term is given below.

\[
P_{S\_T} (k,q)
\]

1. String str <- k.data
2. String token[ ] <- parse(str,#)
3. String qstr<-q.data
4. String qtoken[ ] <- parse(qstr,#)
5. while (qtoken[j]==!NULL)
6. while (token[i]==!NULL)
   if ( token [i] = q[j])
      pos[j][c] <- i
      c <- c + 1
   i  <- i + 1
   end while
   j<-j+1
   c <- 0
end while
7. return pos[ ][ ]

3.3.3 To measure the ordinality of the search term – Ordinality of the search term refers to the situation that if the search term are in the same order as given in search expression or search term are not in order in conjunction with search expression. Ordinality of search term can play an important role in searching the query words and should be provide a higher weight to the terms with the same order. Similarly ordinarily related to multiple occurrences can also be a important factor.

The algorithm for measuring the Ordinality of search term is given below.

Algorithm for two query words
\[
O_{S\_T}(k,q)
\]

1. j=0,p=0
2. String str <- k.data
3.3.4 To measure the proximity of search term – The proximity of search term refers to the facts that how closely the search term are related to each other. While estimating the search term proximity, the search term within the query expression and the distance between reoccurring of search term are considered. The proximity of search term can play a significant role in searching the query words. The algorithm for measuring the proximity of search term is given below.

Algorithm for two query words

\[ X_{S_T}(k,q) \]

1. \( j=0, p=0 \)
2. String str <- k.data
3. String qstr<-q.data
4. String qtoken[ ] <- parse(qstr, #)
5. while(str!=null)
6. while (qtoken[j]!=NULL)
7. \( y=str.indexof(qtoken[j]) \)
8. if(qtoken[j+1]!=!NULL)
9. \( z=str.indexof(qtoken[j+1]) \)
10. if z>y
11. ord[p]=true
12. else
13. ord[p]=false
14. p++
15. endif
16. endif
17. j++
18. str=str.substring(y+ qtoken[j]).length+1)
19. endwhile
20. endwhile
21. return ord

These parameters are considered for the relevant ranking information which used to evaluate fitness function \( f(x) \).

4 EXPERIMENTAL RESULTS

Web site http://www.isuvidha.com and http://shopping.indiatimes.com are used as a test bed. In http://shopping.indiatimes.com/ site, there is product listing which is searched by supplying different keywords according to their preference. 50 different keywords are used for filling a single element of the form. These query words have been collected in the knowledge base of QIIIEP server. There are one text elements in the search form, i.e. product and search respectively. Main interest lies in selecting appropriate keywords through a proper ranking.
score for product by matching the context with isuvidha.com. The context words are stored according to the weights in taken context table. By experimenting of http://isuvidha.com and http://shopping.indiatimes.com/, context words “shop”, “market”, “category”, “purchase”, “online” and “computers and peripherals”, “mobile and accessories”, “electronics”, “movies” respectively having high weight are collected. Further the performance of randomized probabilistic selection algorithm is tested for ranking of key words. A set of words stored on QIIIEP server will be clustered first into groups so that the similar words are in the same cluster. It is done by querying each pair of words to a search engine, which results in a co-occurrence matrix. Web counts of the query words are shown in table 1 for case-1.

<table>
<thead>
<tr>
<th>Query Words</th>
<th>Web Count</th>
<th>Query Words</th>
<th>Web Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>shop product scanner</td>
<td>368,000</td>
<td>category product speaker</td>
<td>32,300,000</td>
</tr>
<tr>
<td>shop product mp3 player</td>
<td>7,000,000</td>
<td>category product monitor</td>
<td>45,600,000</td>
</tr>
<tr>
<td>shop product speaker</td>
<td>9,430,000</td>
<td>purchase product printer</td>
<td>39,500,000</td>
</tr>
<tr>
<td>shop product monitor</td>
<td>8,090,000</td>
<td>purchase product scanner</td>
<td>270,000</td>
</tr>
<tr>
<td>market product printer</td>
<td>8,040,000</td>
<td>purchase product mp3 player</td>
<td>7,690,000</td>
</tr>
<tr>
<td>market product scanner</td>
<td>289,000</td>
<td>purchase product speaker</td>
<td>19,300,000</td>
</tr>
<tr>
<td>market product mp3 player</td>
<td>286,000</td>
<td>purchase product monitor</td>
<td>12,900,000</td>
</tr>
<tr>
<td>market product speaker</td>
<td>8,360,000</td>
<td>online product printer</td>
<td>12,400,000</td>
</tr>
<tr>
<td>market product monitor</td>
<td>10,600,000</td>
<td>online product scanner</td>
<td>359,000</td>
</tr>
<tr>
<td>category product printer</td>
<td>65,500,000</td>
<td>online product mp3 player</td>
<td>7,030,000</td>
</tr>
<tr>
<td>category product scanner</td>
<td>3,490,000</td>
<td>online product speaker</td>
<td>17,300,000</td>
</tr>
<tr>
<td>category product mp3 player</td>
<td>30,900,000</td>
<td>online product monitor</td>
<td>21,600,000</td>
</tr>
</tbody>
</table>

Table 1: Number of Web counts versus query words for case 1

For clustering the words two algorithms i.e. PMI and Chi-square algorithm are used. Here The Chi-square algorithm is used because cluster of the words made by it are more relevant than the PMI method. Here web counts values are divided by one thousands.

The formula for Chi square is given below:

\[
b' = \sum_{w \in W, w \neq w_2} f_{w_1, w}, \quad c' = \sum_{w \in W, w \neq w_1} f_{w_2, w};
\]

\[
d' = \sum_{w, w' \in W, w \neq w_1 \text{ and } w \neq w' \text{ nor } w_2} f_{w, w'}, \quad N' = \sum_{w \in W} f_{w, w'};
\]

Where \( W \) represents a given set of words. Then chi-square (within the word list \( W \)) is defined as

\[
X^2 w(w_1, w_2) = \frac{N' \times (a \times d' - b' \times c')^2}{(a + b') \times (a + c') \times (b' + d') \times (c' + d')}
\]

Where

- \( w_1 = \text{shop product} \) and \( w_2 = \text{printer} \)
- \( w = \text{Set of all the words} \)
- \( b' = \text{sum of all web count words of in the row of shop products except } w_2 = \text{printer} \)
\[ c' = \text{sum of all web count words of in the column of printer except } w_1 = \text{shop product} \]
\[ d' = \text{sum of web count of all words except the } (\text{(sum of all web count words of in the row of shop products)} + \text{(sum of all web count words of in the row of shop products)}) \]
\[ N' = \text{sum of web count of all words} \]

Computed Chi square value given in table 2 for case-1.

<table>
<thead>
<tr>
<th>Context Words/Query words</th>
<th>printer</th>
<th>scanner</th>
<th>mp3player</th>
<th>Speaker</th>
<th>monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>shop product</td>
<td>736000</td>
<td>13000</td>
<td>1152000</td>
<td>348000</td>
<td>183000</td>
</tr>
<tr>
<td>market product</td>
<td>572000</td>
<td>10000</td>
<td>4137000</td>
<td>930000</td>
<td>2357000</td>
</tr>
<tr>
<td>category product</td>
<td>155000</td>
<td>1326000</td>
<td>3248000</td>
<td>4225000</td>
<td>32000</td>
</tr>
<tr>
<td>purchase product</td>
<td>8330000</td>
<td>688000</td>
<td>1561000</td>
<td>103000</td>
<td>5113000</td>
</tr>
<tr>
<td>online product</td>
<td>6510000</td>
<td>235000</td>
<td>228000</td>
<td>1710000</td>
<td>4143000</td>
</tr>
</tbody>
</table>

Table 2: Chi square value matrix

Taking high value of chi square values for making the group for each context words, Groups can be formed as shown in figure 2:

shop product: mp3player, printer
market product: mp3 player, monitor
category product: speaker, mp3 player, monitor
purchase product: printer, monitor, mp3player
online product: printer, monitor, speaker

Case 2 Example:
- site name: http://shopping.indiatimes.com/
- context of the site is:
  1. computers and peripherals
  2. mobile and accessories
  3. electronics
  4. movies

Suppose the query word can be
  - mouse, maachis, nokia, Bluetooth, radio

Table: Computed Chi-square values for each group for each context word under test for case-1
Web counts of the query words are shown in table-3 for case-2.

<table>
<thead>
<tr>
<th>Context Words/ Query Words</th>
<th>mouse</th>
<th>Maachis</th>
<th>nokia</th>
<th>Bluetooth</th>
<th>Radio</th>
</tr>
</thead>
<tbody>
<tr>
<td>computers and peripherals search</td>
<td>27200</td>
<td>41</td>
<td>180000</td>
<td>260000</td>
<td>283000</td>
</tr>
<tr>
<td>mobile and accessories search</td>
<td>67900000</td>
<td>22100</td>
<td>94600000</td>
<td>85900000</td>
<td>32500000</td>
</tr>
<tr>
<td>electronics search</td>
<td>38900000</td>
<td>12700</td>
<td>7920000</td>
<td>11800000</td>
<td>76100000</td>
</tr>
<tr>
<td>movies search</td>
<td>895000</td>
<td>20700</td>
<td>76100000</td>
<td>31900000</td>
<td>416000000</td>
</tr>
</tbody>
</table>

Table 3: Number of Web counts versus query words for case 2

Computed Chi square value is given in table 4 for case-2.

<table>
<thead>
<tr>
<th>Context Words/ Query Words</th>
<th>mouse</th>
<th>maachis</th>
<th>nokia</th>
<th>Bluetooth</th>
<th>Radio</th>
</tr>
</thead>
<tbody>
<tr>
<td>computers and peripherals search</td>
<td>9000</td>
<td>1000</td>
<td>6181000</td>
<td>0.0</td>
<td>35820000</td>
</tr>
<tr>
<td>mobile and accessories search</td>
<td>12925000</td>
<td>1000</td>
<td>3086000</td>
<td>22742000</td>
<td>42858000</td>
</tr>
<tr>
<td>electronics search</td>
<td>76881000</td>
<td>7000</td>
<td>9226000</td>
<td>493000</td>
<td>63090000</td>
</tr>
<tr>
<td>movies search</td>
<td>83992000</td>
<td>1000</td>
<td>7000</td>
<td>19059000</td>
<td>69966000</td>
</tr>
</tbody>
</table>

Table 4: Computed Chi square value for case-2

Taking high value of chi square values for making the group for each context words, Groups can be formed as shown in figure 3:

**Group:**
- computers and peripherals search: nokia, radio
- mobile and accessories search: nokia, Bluetooth, radio, mouse
- electronics search: mouse, Bluetooth, radio
- movies search:

The major advantage of grouping of keywords is to minimize the useless query posting and decreasing the load as well bandwidth of deep web server. The next parts of the algorithm randomly select the query words from groups to post the actual query on deep web. The returned results are analyzed to find out the relevance of that group’s keyword to the deep web query interface. If the returned result is good, the group gets high priority.
5 CONCLUSION

A very sophisticated ranking algorithm is required for selecting the appropriate query word from QIIIEP server for posting to crawl deep web resource. The ranking algorithm for deep web must consider a large number of parameters as far as possible. Some of the main parameters that are considered in proposed algorithm are stemming, context of the query interface, density of search term, position of search term, ordinality of search term and proximity of search term. The proposed algorithm assigns relative weights based on the relationship between the search term, context of page and frequency of occurrence of simultaneous words on surface web. In the proposed algorithm, the weight assigning process heavily depends upon the occurrence of context word, query label and simultaneous keyword on surface web. By including all these factors, ranking can be improved to optimum value for a deep web search as reflected in the experimental results of the proposed algorithms. This algorithm can be modified by including other factors affecting the result analysis to generate next population of genetic approach.

REFERENCES


About the Authors

Dilip Kumar Sharma is B.Sc, B.E.(CSE), M.Tech.(IT), M.Tech. (CSE) and pursuing Ph.D in Computer Engineering. He is life member of CSI, IETE, ISTE., ISCA, SSL and member of CSTA, USA. He has attended 21 short term courses/workshops/seminars organized by various esteemed originations. He has published 23 research papers in International Journals /Conferences of repute and participated in 18 International/National conferences. Presently he is working as Reader in Department of Computer Science, IET at GLA University, Mathura, U.P. since March 2003 and he is also CSI Student branch Coordinator. His research interests are deep web information retrieval, Digital Watermarking and Software Engineering. He has guided various projects and seminars undertaken by the students of undergraduate/postgraduate.

Prof. A. K. Sharma received his M.Tech. (CST) with Hons. from University of Roorkee (Presently I.I.T. Roorkee) and Ph.D (Fuzzy Expert Systems) from JMI, New Delhi and he obtained his second Ph.D. in Information Technology form IIITM, Gwalior in 2004. Presently he is working as the Dean, Faculty of Engineering and Technology & Chairman, Dept of Computer Engineering at YMCA University of Science and Technology, Faridabad. His research interest includes Fuzzy Systems, OOPS, Knowledge Representation and Internet Technologies. He has guided 9 Ph.D thesis and 8 more are in progress with about 175 research publications in International and National journals and conferences. The author of 7 books, is actively engaged in research related to Fuzzy logic, Knowledge based systems, MANETS, Design of crawlers. Besides being member of many BOS and Academic councils, he has been Visiting Professor at JMI, IIIT&M, and I.I.T. Roorkee.
A QIIIEP Based Domain Specific Hidden Web Crawler

Dilip Kumar Sharma
Dept of Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura, UP, India
+91-9927031755
todilipsharma@rediffmail.com

A. K. Sharma
Dept of Computer Engineering
YMCA University of Science and Technology
Faridabad, Haryana, India
ashokkale2@rediffmail.com

ABSTRACT
For context based surfing of World Wide Web in a systematic and automatic manner, a web crawler is required. The World Wide Web consists interlinked documents and resources that are easily crawled by general web crawler, known as surface web crawler. But for crawling the hidden web data, in which the data is hidden behind the html forms requires special type of crawler, known as hidden web crawler. For efficient crawling of hidden web data, the discovery of relevant and proper html forms is very important step. For this purpose a technique for domain specific hidden web crawler is proposed in this paper. The proposed technique is based on the domain specific crawling of World Wide Web. In this approach, a link is followed in a step by step manner, which results in a large source of hidden web databases. Experiential results verify that the proposed approach is quite effective in crawling the hidden web data contents.

Categories and Subject Descriptors
H.3.3 Information Search and Retrieval

General Terms

Keywords
Deep web, hidden web, invisible web, QIIIEP, web crawler.

1. INTRODUCTION
The size of hidden web information is much larger than the surface web and major amount of hidden web information is inaccessible to the users. So there is a challenge to provide the hidden web information to the general user. Present crawlers extract the information from the publically indexable web. There are two different types of forms exists in a publically indexable web. First is authentication form and second is searchable interface. These databases can be accessed through html searchable forms. There are various challenges associated with the discovery of searchable interfaces such as to find the algorithm to define the domain of the databases and how to fill the forms automatically to extract out the most useful information for the users. To meet out these challenges, a domain specific hidden web crawler is proposed in this paper, which does not require human intervention to crawl the hidden web information. In the proposed approach, first the initial links are taken into consideration using already existing techniques to extract the urls then the crawler focuses on these links. The crawler discovers the search interface for various domains. This accelerates the process of searching of the hidden search interfaces very efficiently. The proposed crawler possesses a knowledge base for each domain. So the users can get the proper and relevant information as they want. The duty of the site administrator is only to provide the initial seed set. This initial seed set is taken into the consideration by the crawler for crawling of the relevant links. Crawled relevant links are saved in a database. The database contains the form related pages in a form repository [4][11][13]. Figure 1 shows the barrier in information extraction in the form of search form or login form.

Rest of the paper is organized as follows .In section 2, related work is discussed. Proposed work is presented in section 3. Experimental results are discussed in section 4. Finally a conclusion is presented in section 5.
2. RELATED WORK

A lot of work is done to crawl efficiently the surface web. But a little work is done in area of crawling of hidden web data. Some of the prominent work done in this area is discussed here.

Raghavan et al. proposed an extraction technique which is task/application specific. In this technique, extraction of irrelevant pages is minimized and crawling is not precise due to missing of some pages [11]. Barbosa et al. proposed a focused approach with automatic form filling. In this approach crawling is very relevant and saves resources and time but quality of forms is compromised [3]. Ntoulas et al. proposes technique based on automatic query generation form. In this, crawling is efficient due to crawler generated query but it tries not to involve frequently used multi attribute databases [9]. Alvarez et al. proposes a modified task specific approach based on domain definition which produces effective results for real words data collecting tasks [2]. Lu et al. proposes a technique based on sampling data from the database but it has low efficiency in case of large sample and pool size [7].

Akilandeswari et al. proposes the use of multi agent system on a large database. It is time efficient, fault tolerant but cost is high [1]. Wang et al. proposes a focused crawling in which results are stored in specific domain. It takes few pages for crawling but there is a possibility of wrong semantics [16]. Peisu et al. proposes a model of form with form submission with four additional modules with LVS table in which form submission becomes more accurate with the use of LVS table [10]. Liu et al. proposes a technique based on the concept of minimum executable pattern (MEP). It reduces the problem of data islands but results are not satisfactory in the case of website having limited size of result set [6]. Madaan et al. proposes a technique which regularly updates web repository [8]. Jiang et al. proposes a reinforcement learning based technique which permits the use of efficient characteristics of query words [5].

After going through the related work, it can be concluded that a lot of improvement can be done on the existing architectures in order to improve the searching efficiency.

3. PROPOSED WORK

Downloading of the web pages during crawling can be improved by making it multithreaded. Concept of QIIIEP knowledge base can be used to efficiently process the automatic form filling and form submission [12]. The proposed technique for domain specific hidden web crawling incorporates important features of various researches like automatic query word matching, multithreading concept, task-specific approach, focused crawling, domain-specific crawl etc. Therefore, this technique for domain specific hidden web crawling is a practical implementation of the new concepts as well as the concepts given in the previous researches. Thus it can be predicted that proposed technique for domain specific based hidden web crawling work efficiently according to the need. The technique of the proposed domain specific hidden web crawling is shown in figure 2.

3.1 Module-wise explanation of the proposed technique for domain specific hidden web crawler

3.1.1 Multithread Web Page loader Module
It simply used to download the web content of the urls. The functioning of this module is multithreaded as multiple threads are generated for each url distinctly.

3.1.2 Web Page Analyzer Module
The page analyzer is used to parse the contents of the web page. It analyzes the web page for searchable interfaces.

3.1.3 Form Id Manager
This module extracts the forms according to different domains and provides help to the QIIIEP server in maintaining knowledge base.

3.1.4 Form Submitter Module
This module simply sends the filled form to the http server to inquire for further information.

3.1.5 Query Word Extractor
This module extracts meaningful queries to issue to the query interface. It stores each and every query word associated with specific element of form [15].

3.1.6 Crawl Frontier
Crawl frontier contains all the links which are yet to be fetched from the HTTP server or the links obtained after URL filter. It starts with a list of URLs to visit, called the seeds.

3.1.7 Link Extractor Module
Link extractor extracts the links or the hyper links from the text file for the further retrieval from the http server. The extraction of links is done as per the link identified by the web page analyzer, that are likely to lead to pages that contain searchable form interfaces in one or more steps.
3.1.8 QIIIEP Server
The QIIIEP (Query Intensive Interface Information Extraction Protocol) is an application-level protocol for semantic ontology based query word composition; basically it is a knowledge base which is made with the help of query word extractor module and form manager. This knowledge base is useful for the auto filling of the forms while submitting the query. Whenever a user fires a new query, it is automatically saved into this knowledge base [12].

3.1.9 Link Ranker Module
It is required to rank the links accordingly so that more information is gathered from each link. This is based on link ranking algorithm.

3.1.10 Link Indexer Module
This module plays an important role in the indexing of the generated keywords to the content database. Indexer collects, parses, and stores data to facilitate fast and accurate information retrieval.

3.1.11 Content Database Module
Content database stores all the generated links or keywords in the content database. When user puts any query into the user interface, the indexes are matched with the corresponding links and information is displayed to the user for further processing.

3.1.12 Searching Agent Module
It provides the searching interface on which user places the query. This involves the searching of keywords and other information stored in the content database which is actually stored in it after the whole process of authenticated crawling.

3.1.13 Form DB Module
Form database module is used to store the forms.

3.1.14 URL Repository Module
URL repository module is used for temporary storage of link.

3.1.15 Link Composer Module
This module composes the links on querying by the user. On querying, link composer extract links from the database. After that the links are presented to the user through the user interface.

3.1.16 Interface Generator Module
Interface generator is used to give the view of the contents stored in the content database after the search is complete. The Interface generator for example shows the list of relevant links indexed and ranked by link ranker module and link indexer module respectively.

3.2 Working procedure of domain specific hidden web crawler
1. A web page is fetched by the crawler after requesting to the web server.
2. Fetched web page is analyzed and parsed for finding the relevant links.
3. Web page analyzer analyzes the links for proper selection and filtering.
4. Filtered links are sent to the crawl frontier. Crawl frontier further selects a link and sent it towards the web page fetcher.
5. Crawler analyzes the form, containing the form id to identify the search interface.
6. If the form is associated with the required domain then the fetched web page are indexed and saved in the database.
7. Form analyzer analyzes, extract and save the form in the form database. Simultaneously the relevant query word is fetched and saved in the knowledge base of the QIIIEP server.
8. Form is taken from the database and filled with the query words produced by QIIIEP server [14].
9. Crawler forwards the filled form to the http server by placing the received query words.
10. Crawler starts crawling the contents produced by that query word.
11. Finally extracted pages are saved in the database of the search engine.

3.3 User interface
1. User enters the query to search.
2. The validation of query words takes place.
3. The link composer then fetches links from the database.
4. The content is then produced to the user with the help of interface generator.

4. EXPERIMENT RESULTS
Implementation of proposed technique for domain specific hidden web crawling is done in java platform and by using various APIs of java. Experimental results are given below in table 1.

<table>
<thead>
<tr>
<th>Site Name</th>
<th>Links Crawled</th>
<th>Number of form detected</th>
<th>% Success Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.deeppeep.org/books/">www.deeppeep.org/books/</a></td>
<td>22</td>
<td>9</td>
<td>40.9%</td>
</tr>
<tr>
<td><a href="http://www.readbookonline.net/">www.readbookonline.net/</a></td>
<td>23</td>
<td>19</td>
<td>82.6%</td>
</tr>
<tr>
<td><a href="http://www.onlinecomputerbooks.com/">www.onlinecomputerbooks.com/</a></td>
<td>76</td>
<td>39</td>
<td>51.3%</td>
</tr>
<tr>
<td><a href="http://www.infibeam.com/books/">www.infibeam.com/books/</a></td>
<td>15</td>
<td>5</td>
<td>33.3%</td>
</tr>
<tr>
<td><a href="http://www.bookfinder.com/">www.bookfinder.com/</a></td>
<td>54</td>
<td>15</td>
<td>27.7%</td>
</tr>
</tbody>
</table>

Figure 3 shows the links crawled versus number of forms detected for five sites of book domain.
Figure 4 shows the efficiency versus number of seed links.

![Figure 4. Efficiency v/s no. of seed links](image)

Experimental results show that the efficiency can be made better if a good initial seed set is selected. However the learning power can be improved by making it multithreaded. The graph between time taken for finding the forms in the web shows how the proposed crawler is adaptable as per the number of websites visited.

5. CONCLUSION

Proposed crawler discovers the search interface related to various domains through user interface. Proposed crawler incorporates multi threaded function with multi attributes form processing. Proposed domain specific crawling approach accelerates the process of searching of hidden web search interface very effectively. The proposed crawling technique for domain specific hidden web crawling contains knowledge base which facilitates the extraction of useful and relevant information to the user through automatic form filling. The searchable interfaces are discovered in a lesser time. It can be concluded that proposed technique for domain specific hidden web crawling and protocols are effective to the point of interest of expected results.

6. REFERENCES


TECHNICAL REVIEW OF WEBSITE ANALYTICS TOOLS

Dilip Kumar Sharma
GLA Institute of Technology & Management, Mathura, UP
todilipsharma@rediffmail.com
A. K. Sharma
YMCA University of Science and Technology, Faridabad (Haryana)

Abstract
In today’s scenario internet is changing in a way of presentation to social connectivity. Website analytics is the process of measurement, collection, analysis and reporting of internet data for purposes of understanding and optimizing web use. Website analytics is not only used for estimating the website traffic but can also be used for commercial applications. Results of website analytics applications can also be very helpful for companies for advertising their products. Website analytics collects and analyzes information about the number of visitors, page views etc of the websites which can be further utilized for various purposes. In this paper, a technical review of various web analytics tools has been presented.

Keywords: Website analytics, web browser, URL, JavaScript, IP address.

Introduction

Web analytics tools conventionally have been used to help website owners to study more about clients’ online behavior in order to improve website structural design and online marketing actions. Most of today’s web analytics solutions offer a range of analytical and statistical data, ranging from fundamental traffic reporting to individual-level data that can be linked with price, response and profit data. Understanding web navigational data is crucial tasks for web analysts as it can influence the website upgrading process.

Website analytics tools is the measurement, collection, analysis and reporting of internet data for purposes of understanding and optimizing web usage. There are two broad categories of website analytics i.e. off-site and on-site website analytics. Off-site website analytics refers to web measurement and analysis of website irrespective of whether one own or maintain a website. It includes the measurement of a website's prospective audience (opportunity), share of voice (visibility), and buzz (comments) that is happening on the internet as a whole. On-site website analytics measure a visitor's journey once on the website. On-site website analytics requires the use of drivers and conversions. One illustrations of this process can be to know that which landing pages facilities the subscribers for purchasing. On-site website analytics measures the performance of the website in a commercial context.
The administrator user register himself on the website then statistical website analytics provides the one line script code to the administrator user then he adds that into the body of their all web pages which help to calculate the statistics by using the Ajax technology. Cross-site communication becomes possible whenever the user visit the administrator website. The code which the administrator users paste in his website gets executed by website analytics tool. Therefore calculated statistics get stored in database which update to the administrator user on time to time. Statistical website analytics help to calculate the statistics of other website like entry, exit pages, location of user, color depth, IP addressed these features are sent to administrator user through email or administrator user can login in his/her account to see the statistics of their website. Figure 1 shows the basic framework for web analytics process.

![FRAMEWORK FOR WEB ANALYTICS](image)

Figure 1: Framework for web analytics

**Need for website web analytics tools**

The current need of any successful web portal is to identify visitors preference and for this purpose various amount of information is required such as most popular pages, entry pages of portal, exit pages of portal, the URL from where they come on website, searched keywords by which they came from any search engine, the pages viewed by visitor in one session, duration of visit on per page, no of revisiting users and their duration, visitors geographical location such as country/state/city/ISP, visitor’s browsers/system/IP address etc.

First of all administrator of any website registers for web analytics service for analyzing statistics of website. It is like a normal registration and authentication mechanism by which every client gets its separate profile. After that clients provide the project details for which they require statistics. Finally they download JavaScript service code which must be embedded with that website.

Now for every unique instance of web page fetched by visitor with java script enabled browser service code is executed and it sends information about the visit to service engine which do the following functions:
- Count the number of visit of a particular page lead to compute popular pages.
- Note the reference of previous website from which visitors comes to compute the entered pages.
Note the hit on external link through which it exit lead to compute the exit page.
Note the country through which it entered into website.
Extraction of keywords from search engine reference URL.
Count the difference of time in between initial request of page and next link visit to calculate visitor duration.
Client site cookies used for identifying re-visit of user.
IP to location database are used for finding out the geographical information of visitors.
Suggestive schedule mailing to the client is done for enhancing visitor satisfaction.

**Features of Web Analytics tools**

Description of general parameters of web analytics is given below:

A parameter that log size determines how many of the page loads are included in detailed web analysis. Invisible counter option parameter becomes significant, if user wants to include code on his/her website without disturbing the design of his/her site. Configurable counter parameter allows the visitor to view total page loads. Configurable summary stat is designed to take an idea about the number of pageloads, unique visitors, returning visitors and first time visitors of the website. Drill down is a very strong and most advance feature of new analytics tools by which one can drill down and can view the related visitors. Using the magnifying tool one can zoom in an individual visitor to get a detail report about the source and setting of the user.

Popular pages feature allows seeing the pages of the website that are ranked by popularity. This feature gives the idea of most heavily visited pages. An entry page is the first page at which a visitor arrives first during the visit of the website. Similarly exit pages give the idea that which pages are mostly used as exit pages by the user. Came from feature allows to know that how visitors are finding the information about the website as a whole. Keyword analysis is the tool that allows to know the keywords used to reach the website by ranking their popularity. Recent keyword analysis features of keywords used to reach the website up-to-the-second reporting. Search engine war tool allows to see which search engines generate the most traffic for the website owner. This can be useful in determining if the website is performing as well as it is in specific engines, or if certain important engines are missing.

Visitor path tool provide the details of recent visitors to the website and the navigation path they took through the site. Visit length tool provide the details about how long visitors are spending on the website, to find out what percentage of visitors stay for various periods of time. A country/state/city stats tool allows to identify the geographical location of visitors. ISP stats tool provides the details about the ISPs used by visitors by ranking their popularity. One can drill down on an ISP and show visitors who use that ISP.

Browser stats is an important tool that allows to see which browsers visitors use to view the website, and which are most popular. Resolution stats tool identifies the display resolutions used by visitors ranked by popularity. This can be very useful in ensuring the website looks its best for the majority of users.
Comparison of different existing website analytics tools

<table>
<thead>
<tr>
<th>Site Name Features</th>
<th>Statcounter.com</th>
<th><a href="http://www.phpmyvisites.us">www.phpmyvisites.us</a></th>
<th>2enetworx.com</th>
<th>Google analytics</th>
<th>Thefreecountry.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Reliable invisible web tracker.</td>
<td>PhpMyVisites has maximum protection against intrusions and external attacks.</td>
<td>Powerful than a simple page counter.</td>
<td>Complete Statics</td>
<td>Highly reliable due to inbuilt security features of object oriented language.</td>
</tr>
<tr>
<td>Proprietary</td>
<td>Detailed web stats. Free.</td>
<td>Free open source.</td>
<td>Filtered by year, month, week, day or hour. No code is available.</td>
<td>Many filter and clean interface, free.</td>
<td>Periodically deliver the details but no code is available.</td>
</tr>
<tr>
<td>Language</td>
<td>Statcounter is available in more than 30 languages.</td>
<td>phpMyVisites is available in more than 30 languages.</td>
<td>2enetworx .com is available only in English.</td>
<td>Multiple Language</td>
<td>Thefreecountry.com is available only in English.</td>
</tr>
<tr>
<td>Visibility</td>
<td>Invisible Tracking, no advertisements on the website.</td>
<td>Invisible Tracking, no advertisements on the website.</td>
<td>Invisible tracking is not possible.</td>
<td>Invisible Tracking, no advertisements on the website.</td>
<td>Invisible tracking is not possible.</td>
</tr>
</tbody>
</table>

Table1: Comparison of different existing web analytics tools

Conclusions

Markets want to know if their websites are attracting visitors and whether or not their investment is paying off. With web analytics, one can identify website trends and also able to understand how visitors interact with their website. One can identify the navigational chokepoints that prevent visitors from completing their conversion goals.

All vendors are improving their own web analytics capabilities with respect to extended functionalities. This accelerates competition among pure-play (dedicated) web analytics, customer relationship management, Enterprise software companies or new comers that may enter the market through acquisitions. The challenge for vendors is to accurately and usefully address and analyze not only web data in greater depth but also provide multichannel activity in real time. The huge amount of data to be processed drives the web analysts to demand more visual and explorative systems to get a deeper insight in their data.
References

[5] Internet Marketing-Web Analytics Methodology, ©Bruce Clay, Inc
[6] Rick Tansun,"10 web analytics tools For Tracking Your Visitors"
http://www.sitepoint.com/blogs/2009/03/26/10-web-analytics-packages-for-tracking-your-visitors
[8] www.gostats.com
ABSTRACT

The volume of information on the web is increasing day by day. The information in the web can be broadly categorized into two types i.e. surface web and deep web. The surface web pages can be easily indexed through conventional techniques but the deep web, whose size assumed to be thousand times larger than surface web, cannot be indexed through conventional search technique. The first stage of the extraction of the deep web information is the detection of deep web search interface. A search interface is generally consisting of html forms. The conventional techniques of searching the deep web information is done by filling the html forms on the search interface manually but recently the research is going on automatic accessing and understanding of html forms. Being the first stage of deep web extraction process, the detection of deep web search interface becomes one of the important module of deep web information retrieval. In this paper a technical analysis of some of the important deep web search interface detection techniques is done to find out their relative strengths and limitations with reference to current development in the field of deep web information retrieval technology.

Keywords Deep web, hidden web, search interface detection, crawler, random forest.

INTRODUCTION

The whole process of extraction of information from deep web can be broadly categorized into four steps i.e. query interface analysis, values allotment, response analysis & navigation and relevance ranking. Query interface analysis is the first and most important step for deep web information retrieval. In query interface analysis, a request of fetching a web page from a web server is made by a crawler. After completion of the fetching process, an internal representation of the web page is produced after parsing and processing of html forms based on the developed model. Further the query interface analysis can be broken into the some modules that are detection of hidden web search interface, search form schema matching and domain ontology identification. In these module the detection of hidden web search interface is the first and foremost step towards deep web information retrieval. As expected, a human user can easily identify a deep web search interface but to understand a deep web search interface through an automatic technique without human intervention is a challenging task [1][2][3][4][5]. Figures 1 depict the different types of search interfaces.

![Figure 1 Different types of search interface](image-url)
RELATED WORK

Random Forest Algorithm Based Approach

One of the prominent works for detection of deep web search interface is done by Leo Breiman (2001)[6] in form of random forest algorithm. A random forest algorithm detects the deep web search interface by using a model, based on decision trees classification. A random forest model can be defined as a collection of decision trees. A decision tree can be generated by bootstrapping processing of the training data. Various classification trees can be generated through random forest algorithm. To classify a new object from its input vector, the sample vector is passed to every tree defined in algorithm. A decision for classification is given by every tree. A decision about most voted classification is done by using all of the classification results of the individual trees. The advantages of random forest algorithm are that it exhibits a substantial performance improvement over single tree classifiers and injecting of the right kind of randomness makes accurate classifiers and regulators. The disadvantage of this algorithm is that it may select unimportant and noisy features in the training data, as a result a bad classification results because of its random selection feature.

Task-Specific, Human-Assisted Based Approach

One of the deep web crawler architecture is proposed by Sriram Raghavan and Hector Garcia-Molina (2001) [7]. In this paper, a task-specific, human-assisted approach is used for crawl the hidden web. There are two basic problems related to deep web search, firstly the volume of the hidden web is very large and secondly there is a need of such type of crawlers which can handle search interfaces efficiently, which are designed mainly for humans. In this paper a model of task specific human assisted web crawler is designed and relized in HiWE (hidden web exposure). The HiWE prototype built at Stanford which crawl the dynamic pages. HiWE is designed to automatically process, analyze, and submit forms, using an internal model of forms and form submissions. HiWE uses a layout-based information extraction (LITE) technique to process and extract useful information. The advantages of HiWE architecture is that its application/task specific approach allows the crawler to concentrate on relevant pages only and with the human assisted approach automatic form filling can be done. Limitations of this architecture are that it is not precise with response to partially filled forms and it is not able to identify and respond to simple dependency between form elements.

Hidden Web Agents Based Approach

A technique for collecting hidden web pages for data extraction is proposed by Juliano Palmieri Lage et al. (2002) [8]. In this technique the authors have proposed the concept of web wrappers. A web wrapper is programs which extract the unstructured data from web pages. It takes a set of target pages from the web source as an input. These set of target pages are automatically generated by an approach called “Spiders”. Spiders automatically traverse the web for web pages. Hidden web agents assist the wrappers to deal with the data available on the hidden web. The advantage of this technique is that it can access a large number of web sites from diverse domains and limitation of this technique is that it can access only that web site that follow common navigation patterns. Further, modification can be done in this technique to cover navigation patterns based on these mechanisms.

Single Tree Classifiers Based Approach

A technique for automated discovery of search interface from a set of html forms is proposed by Jared Cope, Nick Craswell and David Hawking (2003) [9]. This paper defined a novel technique to automatically detect search interface from a group of html forms. A decision tree was developed with the C4.5 learning algorithm using automatically generated features from html markup that can give a classification accuracy of about 85% for general
web interfaces. Advantage of this technique is that it can automatically discover the search interface. Limitation of this technique is that it is based on single tree classification method and number of feature generation is limited due to use of limited data set. As a future work, modification is suggested that a search engine can be develop using existing methods for other stages along with the proposed one with a technique to eliminate false positives.

2P Grammar & Best Effort Parser Based Approach

A technique for understanding web query interfaces through best effort parsing with hidden syntax is proposed by Zhen Zhang et al. (2004) [10]. This paper addresses the problem of understanding web search interfaces by presenting a best-effort parsing framework. The paper presented a form extractor framework based on 2P grammar and the best effort parses in a language parsing framework. It identifies the search interface by continuously producing fresh instances by applying productions until attaining a fix-point, when no fresh instance can be produced. Best effort parser technique minimizes wrong interpretation as much as possible in a very fast manner. It also understands the interface to a large extent. Advantage of this technique is that it is a very simple and consistent technique with no priority among preferences and it can handle missing elements in form and limitation of this technique is that establishment of single global grammar that can be interacted to the machine globally is a critical issue.

Automatic Query Generation Based Approach

A technique named as “siphoning hidden web data through key word based interface” for retrieval of information from hidden web databases through generation of a small set of representative keywords and build queries is proposed by Luciano Barbosa and Juliana Freire (2004) [11]. This technique is designed to enhance coverage of deep web. Advantage of this technique is that it is a simple and completely automated strategy that can be quite effective in practice, leading to very high coverage of deep web. Limitation of this technique is that it is not able to achieve the coverage for collection whose search interface fixes a number of results. Further the authors have advised that modification can be done in this algorithm to characterize search interfaces techniques in a better way so that different notions and levels of security can be achieved.

Weighted Feature Selection Algorithm Based Approach

An improved version of random forest algorithm is proposed by Deng et al. (2008) [12]. In this improved technique a weighted feature selection algorithm is proposed to generate the decision trees. The advantage of this improved algorithm is that it minimizes the problem of classification of high dimension and sparse search interface using the ensemble of decision trees. Disadvantage of this improved algorithm is that it is highly sensitive towards the changes in training data set.

Feature Weighted Random Forest Algorithm Based Approach

Further improvement in random forest algorithm is done by Yunming Ye et al. (2009) [13] by using feature weighting random forest algorithm for detection of hidden web search interface. This paper had presented a feature weighting selection process rather than random selection process. Advantage of this technique is that it makes a weighted feature selection process instead of random selection hence reduces the chances of noisy feature selection and limitation of this techniques is that features available only in the search forms were used. Future modification suggested in random forest algorithm to investigate more feature weighting methods for construction of random forests.

Naive Bayesian Web Text Classification Algorithm Based Approach

An algorithm named as “The naive bayesian web text classification algorithm” is proposed by Ping Bai and Junqing Li (2009) [14] for automatic and effective classification of web pages with reference to given model for machine
learning. In the conventional techniques, category abstracts are produced using the inspection by domain experts
either through semiautomatic method or artificial method. All the items are provided equal important according to
conventional common bayesian classifier whereas according to improved naive bayesian web text classification
algorithm, whole of the items in every title are provided higher importance to others. The strength of this technique
is that text classification results are very accurate and further scope in this algorithm is suggested to make the
classification process automatic in an efficient way.

**Domain Ontology Based Approach**

An approach for automatic detection and unification of web search query interfaces using domain ontology is
proposed by Anuradha and A.K.Sharma (2010) [15]. The technique proposed in this paper works by concentrating
the crawler on the given topic considering the domain ontology. This technique results in the pages which contains
the domain specific search form. The strengths of this technique are that results are produced from multiple sources,
human effort is reduced and results are very accurate in less execution time. Limitation of this technique is that it is
domain specific.

**SUMMARY OF VARIOUS TECHNIQUES FOR DETECTION OF DEEP WEB SEARCH INTERFACE**

By going through the literature survey of some deep web search interface detection techniques, it is concluded that
each techniques for detection of deep web search interface have some relative strengths and limitations. A tabular
summary is given below in table 1, which summarizes the techniques, strengths and limitations of some of important
detection techniques for deep web search interface.
<table>
<thead>
<tr>
<th>Authors</th>
<th>Technique</th>
<th>Strengths</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leo Breiman (2001)</td>
<td>Forest of regression trees as classifiers.</td>
<td>A substantial improvement in performance over single tree classifiers.</td>
<td>May include un-important or noisy features.</td>
</tr>
<tr>
<td>Palmieri Lage et al.</td>
<td>Hidden Web Agents.</td>
<td>Wide coverage of distinct domains.</td>
<td>Restricted to web sites that follow common navigation patterns.</td>
</tr>
<tr>
<td>Jared Cope et al.</td>
<td>Single tree classifiers.</td>
<td>Automatically discovery of search interface, performed well when rules are generated on the same domain.</td>
<td>Long rules, large size of feature space in training samples, Over fitting. Classification precision is not very satisfying.</td>
</tr>
<tr>
<td>Zhen Zhang et al.</td>
<td>2P Grammar and Best effort Parser.</td>
<td>Very simple and consistent, No priority among preferences, Handling of missing elements in form.</td>
<td>Critical to establish single global grammar that can be interacted to the machine globally.</td>
</tr>
</tbody>
</table>
CONCLUSION

Deep web search interface are the entry point for the searching of the deep web information. A deep web crawler should understand and detect the deep web search interface efficiently to facilitate the further process of deep web information retrieval. An efficient detection of deep web search interface may results towards a significant retrieval of deep web information so the first and foremost step of deep web information retrieval is the efficient understanding and detection of deep web search interface. In this paper a technical analysis of some of the techniques for detection of deep web search interface is done and it is concluded that each of them have some relative strengths and limitations in detecting of deep web search interface. To explore the deep web information efficiently, an efficient technique for detection of deep web search interface should be designed which should have strengths simultaneously and particularly in terms of wide coverage of different domains, automatic procedure, resistant to noisy and unwanted features, ability to consider the features as per their importance, application specific approach as per requirement and user friendly approach. Finally the technique for detection of deep web search interface should be compatible with current web technology.

REFERENCES


Dilip Kumar Sharma is B.Sc, B.E.(CSE), M.Tech.(IT), M.Tech. (CSE) and pursuing Ph.D in Computer Engineering. He is life member of CSI, IETE, ISTE., ISCA, SSI and member of CSTA, USA. He has attended 21 short term courses/workshops/seminars organized by various esteemed originations. He has published 21 research papers in International Journals /Conferences of repute and participated in 18 International/National conferences. Presently he is working as Reader in Department of Computer Science, IET at GLA University, Mathura, U.P. since March 2003 and he is also CSI Student branch Coordinator. His research interests are deep web information retrieval, Digital Watermarking and Software Engineering. He has guided various projects and seminars undertaken by the students of undergraduate/postgraduate.

Prof. A. K. Sharma received his M.Tech. (CST) with Hons. from University of Roorkee (Presently I.I.T. Roorkee) and Ph.D (Fuzzy Expert Systems) from JMI, New Delhi and he obtained his second Ph.D in Information Technology form IIITM, Gwalior in 2004. Presently he is working as Dean, Faculty of Engineering and Technology & Chairman, Dept of Computer Engineering at YMCA University of Science and Technology, Faridabad. His research interest includes Fuzzy Systems, OOPS, Knowledge Representation and Internet Technologies. He has guided 9 Ph.D thesis and 8 more are in progress with about 175 research publications in International and National journals and conferences. The author of 7 books, is actively engaged in research related to Fuzzy logic, Knowledge based systems, MANETS, Design of crawlers. Besides being member of many BOS and Academic councils, he has been Visiting Professor at JMI, IIIT&M, and I.I.T. Roorkee.

Dilip Kumar Sharma
G.L.A. Institute of Technology and Management, Mathura, U. P., India,
todilipsharma@rediffmail.com
A. K. Sharma
YMCA University of Science & Technology, Faridabad, Haryana, India

ABSTRACT:

Web crawlers specialize in downloading web content and analyzing and indexing from surface web, consisting of interlinked HTML pages. Web crawlers have limitations if the data is behind the query interface. Response depends on the querying party’s context in order to engage in dialogue and negotiate for the information. In this paper, the authors discuss deep web searching techniques. A survey of technical literature on deep web searching contributes to the development of a general framework. Existing frameworks and mechanisms of present web crawlers are taxonomically classified into four steps and analyzed to find limitations in searching the deep web.

Keywords: Crawler; Deep Web; OAI-PMH; Ontology; Schema Mapping; SRU; Z39.50

INTRODUCTION

The versatile use of the internet has proved a remarkable revolution in the history of technological advancement. Accessibility of web pages starting from zero in 1990 has reached more than 1.6 billion during 2009. It is like a perineal stream of knowledge. The more we dig the more thirst can be quenched. Surface data is easily available on the web. Surface web pages can be easily indexed through conventional search engine. But the hidden, invisible and non indexable contents which cannot be retrieved through conventional methods used for surface web and whose size is estimated to be thousands of times larger than the surface web is called deep web. The deep web consist of a large database of useful information such as audio, video, images, documents, presentations and various other types of media. Today people really heavily depend on internet for numerous applications such as flight and train reservations, to know about new product or to find any new locations and job etc. They can evaluate the search result and decide which of the bits or scraps reached by the search engine is most promising (Galler, Chun, & An, 2008).

Unlike the surface web, the deep web information is stored in searchable databases. These databases produce results dynamically after processing the user request (BrightPlanet.com LLC, 2000). Deep web information extraction first uses the two regularities of the domain knowledge and interface similarity to assign the tasks that are proposed from users and chooses the most effective set of sites to visit by ontology inspection. The conventional search engine has limitations in indexing the deep web pages so
there is a requirement of an efficient algorithm to search and index the deep web pages (Akilandeswari & Gopalan, 2008). Figure 1 shows the barrier in information extraction in the form of search form or login form.

**Contributions** This paper attempts to find the limitations of the current web crawlers in searching the deep web contents. For this purpose a general framework for searching the deep web contents is developed as per existing web crawling techniques. In particular, it concentrates on survey of techniques extracting contents from the portion of the web that is hidden behind search interface in large searchable databases with the following points.

- After profound analysis of entire working of deep web crawling process, we extracted qualified steps and developed a framework of deep web searching.
- Taxonomic classification of different mechanisms of the deep web extraction as per synchronism with developed framework.
- Comparison of different algorithms web searching with their advantages and limitations.
- Discuss the limitations of existing web searching mechanisms in large scale crawling of deep web.

![Figure 1. Query or credentials required for contents extraction](image)

**CURRENT DEEP WEB INFORMATION RETRIEVAL FRAMEWORK:**

After exhaustive analysis of existing deep web information retrieval processes, a deep web information retrieval framework is developed, in which different tasks in deep web crawling are identified, arranged and aggregated in sequential manner. This framework is useful for understanding entire working of deep web crawling mechanisms as well as it enables the researcher to find out the limitations of present web crawling mechanisms in searching the deep web. The taxonomical steps of developed framework can be classified into following four major parts.
• **Query Interface Analysis:** First of a crawler will request for any web server to fetch a page. After fetching process, it parses and process the form to build an internal representation of web page based on the model developed.

• **Values Allotment:** It provides appropriate value to each and every input element by using different combinations of keywords, which will be allocated by using some string matching algorithms by analyzing the form labels using knowledge base.

• **Response Analysis and Navigation:** Crawler analyzes the response web pages to check if the submission yielded valid search results. Crawler uses this feedback to tune the values assigned and crawl the hypertext links iteratively, received by response web page to some pre-specified depth.

• **Relevance Ranking:** Relevance ranking means the order in which search engine should return the URLs, produced in response to a user’s query, to show more relevant pages on priority basis. During this step, the deep web is a completely different from traditional web. In deep web there is none of those <A href> links to content and no association of links to be followed. So in deep web retrieval process, quality of a page cannot be predicted with its reference. This need is definitely highly demanded in the future framework to be developed to increase the quality of deep web contents.

These above steps are represented in following figure 2 which demonstrates the flow of control over the deep web contents extraction mechanism.
Exhaustive literature analysis of the above taxonomically classified steps are given below.

**QUERY INTERFACE ANALYSIS**
Query interface analysis can be taxonomically partitioned into the following steps.

**Detection of Hidden Web Search Interfaces**

Pages for search interfaces are commonly HTML forms which is filled and submitted by users and server respond appropriately according to filled forms. But every form is not search interfaces. The problem is to identify a form which is a search interface. Search interface identification can be taxonomically categorized in the following three ways:

**Based on Heuristic Rules**

One of the simplest methods of search interface identification is done by using heuristic rules. “Heuristic” here refers to a general problem-solving rule or set of rules that do not guarantee the best solution or even any solution, but serves as a useful guide for interface matching. Automatic search interface detection was first defined by (Raghavan & Garcia-Molina, 2001), whose crawling system use heuristic rules to detect the hidden databases. The paper (Lage et al., 2004) use two heuristic rules and utilizes a pre-existing data repository to identify the contents of deep web. This paper exploits the advantage of some patterns that is available in websites to find out the navigation path to be followed.

**Decision trees classification**
One of the approaches for search interface identification is based on decision trees classification models to detect search interface. One of such example is random forest algorithm. A random forest model contains a group of decision trees defined by bootstrapping the training data. An improved version of random forest algorithm (RFA) known as improved random forest algorithm (IRFA) is proposed by Deng et al. In IRFA the original RFA is extended with a weighted feature selection method to select more representative subset of features for building each decision trees. These can be vulnerable to changes in the training dataset. IRFA eliminates the problem of classification of high dimensional and sparse search interface data through the ensemble of decision trees (Deng, Ye, Li & Huang, 2008). Future work in this regard is to identify other techniques for feature waiting for the generation of random forest. Currently this paper uses the features available in the search form themselves. The paper (Jared Cope et al., 2003) defined a novel technique to automatically detect search interface from a group of HTML forms. Future work in this regard will be to develop a technique to eliminate false positives.

**Best-Effort Parsing Framework**

It identifies the search interface by continuously producing fresh instances by applying productions until attaining a fix-point, when no fresh instance can be produced. An example is shown in Figure 3 (Zhang, He & Chang, 2004) in which, the parser starts from a group of tokens to iteratively generate fresh instances and finally generates parse trees. A complete parse tree related to a unique instance of the start symbol QI that take cares of all tokens. But, due to the significant ambiguities and incompleteness, the parser may not derive any complete parse tree and only conclude with multiple incomplete parse trees. Best effort parser technique minimizes wrong interpretation as much as possible in a very fast manner. It also understands the interface to a large extent.

![Figure 3. Best-Effort (Fix point) parsing process](image)

Table 1 depicts the comparison of different hidden web search interfaces detection mechanism.

<table>
<thead>
<tr>
<th>Search Techniques</th>
<th>Authors Name</th>
<th>Advantages</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Based on Heuristic Rules</td>
<td>S. Raghavan et al.</td>
<td>Effective label extraction technique</td>
<td>No auto-learn capability, unstable, un-scalable to</td>
</tr>
</tbody>
</table>
Table 1. Comparison of different hidden web search interfaces detection techniques

<table>
<thead>
<tr>
<th>Search Form Schema Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>After detection of hidden web search interface, the next task is to identify accurate matching for finding semantic correspondences between elements of two schemas. Schema extraction of query interface is one of the very prime research challenges for comparing and analysis of an integrated query interface for the deep web. Many algorithms are suggested in last few years which can be classified in two groups.</td>
</tr>
</tbody>
</table>

**Heuristic**

Heuristic techniques for Search form Schema Mapping are based on guessing relations which may consider similar labels or graph structures and can be further taxonomically classified in two groups.

- **Element-level explicit techniques**
  Explicit techniques use the semantics of labels such as used in precompiled dictionary (Cupid, COMA), Lexicons (S-Match, CTXmatch) e.g. sedan: Car is a hypernym for Four Wheeler, therefore, Car $\subseteq$ Four Wheeler.

- **Structure-level explicit techniques**
  These are based on taxonomic structure format (Anchor-Prompt, NOM) e.g. DEPTT and Department can be found as an appropriate match.

**Formal**

Formal techniques are based on model-theoretic semantics which is used to justify their results.

- **Element-level explicit techniques**
  It uses the OWL properties (NOM) e.g. same Class as constructor explicitly states that one class is equivalent to the other such as hybrid car = Car or CNG based car.

- **Structure-level explicit techniques**
The approach is based on to translate the matching problem, namely the two graphs (trees) and mapping queries into propositional formula and then to check it for its validity.

- **Modal SAT (S-Match)**
  The idea is based on to enhance propositional logics with modal logic operators. Therefore, the matching problem is translated into a modal logic formula which is further checked for its validity using sound and complete search procedures.

Many automatic or semi-automatic search form schema mapping systems meticulous in a simple 1:1 matching are proposed such as Cupid method (Madhavan, Bernstein & Rahm, 2001). This method proposed a new technique by including substantial linguistic matching step and by biasing matches by leaves of a schema. Future work in this regard include integrating cupid transparently with an off-the-shelf thesaurus using schema notations for the linguistic matching and automatic tuning of control parameter.

Do and Rahm (2002), develops the COMA schema matching system to combine multiple matchers. It uses COMA as a framework to evaluate the effectiveness of different matcher and their combination for real world schemas. Future work in this regard is to add other match and combination algorithm in order to improve match quality. LSD method (Doan, Domingos & Levy, 2000), proposes a initial idea for automatic learning mappings between source schemas and mediated schemas. (Doan, Domingos & Halevy, 2001) describe LSD a system that employs and extends current machine learning techniques to semi-automatically find semantic mappings between the source schema and mediated schema.

The paper (Melnik, Garcia-Molina & Rahm, 2002) present a matching algorithm based on a fixed point computation. It uses two graphs such as schemas catalogs as input and produces as output a mapping between corresponding nodes of the graph. The paper (Kaljuvee, Buyukkokten, Molina & Paepcke, 2001) proposed a technique for automatically and dynamically summarize and organize web pages for displaying on a small devices such as PDA. This paper proposed eight algorithms for performing label-widget matches in which some algorithms based on n-gram comparisons and others based on common form layout specifications. Results can be improved by using syntactic and structural feature analysis.

For schema extraction, (He, Meng, Yu & Wu, 2005) consider the non-hierarchical structure of query interface assuming that a query interface has a flat set of attributes and the mapping of fields over the interfaces is 1:1, which neglects the grouping and hierarchical relationships of attributes. So the semantics of a query interface cannot be captured correctly.

Literature (Wu, Yu, Doan & Meng, 2004) proposed a hierarchical model and schema extraction approach which can group the attributes and improve the performance of schema extraction of query interface, but they show the poor clustering capability of pre-clustering algorithm due to the simple grouping patterns and schema extraction algorithm and possibly outputs the subsets inconsistent with those grouped by pre-clustering algorithm. Semantic matching is based on two ideas: (i) To discover an alignment by computing semantic relations (e.g., equivalence, more general); (ii) To determine semantic relations by analyzing
the meaning (concepts, not labels). Although this paper provides a good accuracy but it can be improved by investigating the interaction to help break ties when the ordering based strategy does not work. Another improvement can be done by investigating the use of automatic interface model procedure into the proposed approach.

Most of the proposed search form schema mapping techniques require human involvement and not suitable for dynamic large scale data sets. Other approaches such as DCM framework (He & Chang, 2006) and MGS framework (He et al., 2003) pursues a correlation mining approach by exploiting the co-occurrence patterns of attributes and proposes a new correlation measures while other (Zhong, Fu, Liu, Lin & Cui, 2007) hypothesizes that every application field has a hidden generative model and can be viewed as instances generated from models with possible behaviors. There are certain issues in this algorithm that can be improved such as how to select the appropriate measure to filter out false matching and how to design a dynamic threshold to apply it to all domains. Table 2 depicts the comparison of different schema mapping techniques.

New schema extraction algorithm Extr (Qiang, Xi, Qiang &, Zhang, 2008), which is based on three metrics (LCA) precision, (LCA) recall, and (LCA) F1 are employed to evaluate the performance of schema extraction algorithm by pre-clustering of attributes P by using MPreCluster than all the subsets in P are clustered once again according to spatial distance. Finally all singleton clusters are merged according to n-way constrained merging operation as algorithm Ncluster. So the result is a hierarchical clustering H over the attributes of query interface on the deep web. The experimental results indicate that proposed algorithm can obviously improve the performance of schema extraction of query interfaces on the deep web and avoid resulting inconsistencies between the subsets by pre-clustering algorithm and those by schema extraction algorithm.

<table>
<thead>
<tr>
<th>Name</th>
<th>Techniques</th>
<th>Advantages</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artemis</td>
<td>Affinity-based analysis and hierarchical clustering</td>
<td>Effective label extraction technique, with high submission efficiency.</td>
<td>Falls into the alignments as likeness clues category.</td>
</tr>
<tr>
<td>Cupid</td>
<td>Structural schema matching techniques</td>
<td>Emphasize the name and data type similarities present at the finest level of granularity (leaf level).</td>
<td>Lack of integrating cupid transparently with an off-the-shelf thesaurus using schema notations for the linguistic matching and automatic tuning of control parameter.</td>
</tr>
<tr>
<td>DCM</td>
<td>Hybrid n:m</td>
<td>Identifies and clusters synonym elements by analyzing the co-occurrence of elements. DCM framework can find complex matching in many domains.</td>
<td>Ineffective against ‘noisy’ schema. DCM cannot differentiate frequent attributes from rare attributes. DCM tries to first identify all possible groups and then discover the matching between them.</td>
</tr>
</tbody>
</table>
Table 2. Comparison of different schema mapping techniques

<table>
<thead>
<tr>
<th>GLUE</th>
<th>Composite n:m</th>
<th>It uses a composite approach, as in LSD, but does not utilize global schema.</th>
<th>Accuracy of the element similarity depends on training.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSD</td>
<td>Corpus-based Matching</td>
<td>It provides a new set of machine-learning based matchers for specific types of complex mappings expressions. It provides prediction criterion for a match or mismatch.</td>
<td>Performance depends upon training data.</td>
</tr>
</tbody>
</table>

The last but not least correlated-clustering framework works in four phases. In first phase, it finds frequent attributes in the input attribute groups. In second phase, it discovers group where positively correlated attributes to form potential attribute groups, according to positive correlation measure and defined threshold. In third phase, it partitions the attributes into concepts, and cluster the concepts by calculating the similarity of each two concepts. At last, it ranks to discover matching and then use a greedy matching selection algorithm to select the final matching results.

Komal Kumar Bhatia et al. (Bhatia & Sharma, 2008) presented in his research literature where mapping is done by using domain specific interface mapper in which search interface repository will work for matching purpose. It includes extensible domain specific matcher library. The multi-strategy interface matching is done in three steps: parsing, semantic matching and semantic mapping generation, in step one SI parser is used to extract interface schema. In second step each tuple has mapped by fuzzy matching, domain specific thesaurus and data type matching. Finally SVM generator creates matrices of mapping that are identified by the matching library. DSIM also used mapping knowledge base for avoiding repetition in map effort.

Future work may include testing the schema extraction algorithm on real world data and testing the efficiency of schema matching and schema merging over variety of query interfaces.

Domain ontology identification

Ontology is a formal specification of a shared conceptualization (Niepert, Buckner & Allen, 2007). This step is required for analyzing area or specialization of web page so that in further steps appropriate data set will be efficiently placed in query part of the page. This can be taxonomically classified into four different groups.

RDF annotations based ontology identification

Deitel et al. (Deitel, Faron & Dieng, 2001) present an approach for learning ontology from resource description framework (RDF) annotations of web resources. To perform the learning process, a particular approach of concept formation is adopted, considering ontology as a concept hierarchy, where each concept is defined in extension by a cluster of resources and in intension by the most specific common description of these resources. A resource description is a RDF sub graph containing all resources reachable from the considered
resource through properties. This approach leads to the systematic generation of all possible clusters of descriptions from the whole RDF graph incrementing the length of the description associated to each particular concept in the source graph.

**Metadata annotations based ontology identification**

Stojanovic et al. (Stojanovic, Stojanovic & Volz, 2002) presents an approach for an automated migration of data-intensive web sites into the semantic web. They extract light ontologies from resources such as XML Schema or relational database schema and try to build light ontologies from conceptual database schemas using a mapping process that can form the conceptual backbone for metadata annotations that are automatically created from the database instances.

**Table Analysis based ontology identification**

The paper (Tijerino et al., 2005) presents an approach Table Analysis for Generating Ontologies (TANGO) to generate ontology based on HTML table analysis. TANGO discovers the constraints, match and merge mini-ontology based on conceptual modeling extraction techniques. TANGO is thus a formalized technique of processing the format and content of tables that can aim to incrementally build a appropriate reusable conceptual ontology.

**DOM Based ontology identification**

Zhiming Cui et al. published his research (Cui, Zhao, Fang & Lin, 2008) which works in following steps.

- Use the query interfaces information to generate a mini-ontology model.
- To draw the instances from the intermediate result pages.
- Use of various sources to generate ontology mappings and merging ontology.

In first step it employs vision-based approach to extract query interface (Zhao et al., 2007), and in second step data region discovery is done by employing a DOM parser to generate the DOM parsed trees from the result pages. Based on parent length, adjacent and normalized edit distance, extraction of hierarchical data is done by vision-based page segmentation algorithm. In final step merging is done by label instances pairs mined from the result pages into the domain ontology. They also give an idea, of absent attribute annotation for finding absent attributes in the data records. The next generation semantic web framework is required to be able for handling knowledge level querying and searches. Main area to be focused for research is concept relations learning to increase the efficiency of the system.

**VALUES ALLOTMENT**

Values allotment techniques can be taxonomically classified in the following groups.

**Integrating the Databases for Values mapping**
Integration of the databases with the query interfaces is done in this process. The search form interface brings together the attributes and this step will analyze appropriate data values by structure characteristics of the interface and the order of attributes in the area as much as possible. The integration of query interfaces can provide a unified access channel for users to visit the databases which belong to the same area. For integrating interfaces, the core part is dynamic query translator, which can translate the users’ query into different form (Meng, Yin & Xiao, 2006) (He, Meng, Yu & Wu, 2003). There is also some scope for improvement by using open directory hierarchy to detect more hypernymy relationship.

**Fuzzy comprehensive evaluation methods**

In this mapping is done by fuzzy comprehensive evaluation (Chen, Wen, Hu & Li, 2008) which map the attribute of the form to the data values. First it analyzes a form mapping with a view the data range mapping is the key issues and then it select the optimum matching result. Further scope of work in this area includes finding a model to detect the optimal configuration parameter to produce the results with high accuracy.

**Query Translation Technique**

Query translation technique is used to get query across different deep web sources to translate queries to sources without primary knowledge. The framework takes source query form and target query form as input and output a query for target query.

Some methods can be concerned such as type-based search-driven translation framework by leveraging the “regularities” across the implicit data types of query constraints.

**Patterns Based**

(He & Zhang et al., 2005) found that query constraints of different concepts often share similar patterns i.e. same data type (page title or author name etc.) and encode more generic translational knowledge for each data type. This indicates explicit declaration of data type that localities the translatable patterns. Therefore, getting translation knowledge for each data types are more generic rather identifying translation based on source of information. For translation purpose it uses extensible search-driven mechanism which uses type based translation.

**Hierarchical relations based**

Other approach published by Hao Liang et al (Liang, Zuo, Ren & Sun, 2008) will map on the basis of three constraints:

i. The same word in different schemas of query forms generally has the same semantics, with or without the same formalizations.

ii. The words of two different forms may have same meaning. For this purpose use of thesaurus or dictionaries is required. In addition to this for dealing some special subjects like computer and electronics, some specialized dictionaries related to that subject is required.

iii. There may be some hierarchical relations between the words, e.g., X is a hypernym of Y if Y is a kind of X, and on the other hand Y is hyponym of X. For example, bus is a vehicle it
indicates that vehicle is hypernym of bus, vehicle is also equivalent to automobile. Thus bus is equivalent to automobile semantically.

Domain ontology is widely used in different areas. There is a scope of lot of work to be done for building ontology. One of the issues is to make automatic domain ontology inspection for some particular domain to gain information about the domain.

**Type-based predicate mapping**

Type-based predicate mapping method (Zhang et al., 2005) proposed by Z. Zhang focusing on text type attribute with some constraint. The constraint is restricted to the query condition, for example, any, all or exactly. The minimum search space is computed but the cost of the query is not considered.

**Cost model based**

Another method of query translation is proposed by Fangjiao Jiang, Linlin Jia, Weiyi Meng, Xiaofeng Meng which is based on a cost model for range query translation (Jiang et al., 2008) in deep web data integration. This paper proposes a multiple regression cost model based on statistical analysis for global range queries that involve numeric range attributes. It works on the basis of following concepts.

i. Using a statistical-based approach for translating the range query at the global level after proposing a multiple-regression cost model (MrCoM).

ii. For selecting significant independent variables into the MrCoM, a pre-processing-based stepwise algorithm is defined.

iii. Global range queries are classified into three types and different models are proposed for each of the three types of global range query.

iv. Experimental process is done to verify the efficiency of the proposed method.

After going through the above process conclusion is that MrCoM has good fitness and query strategy selection of MrCoM is highly accurate.

**RESPONSE ANALYSIS AND NAVIGATION**

Techniques for Response analysis and navigation can be taxonomically categorized in the following parts.

**Data Extraction Algorithm**

Data extraction is another important aspect of deep web research, which involves in extracting the information from semi-structured or unstructured web pages and saving the information as the XML document or relationship model. The paper (Crescenzi, Mecca, & Merialdo, 2001) have done a lot of work in this field. Additionally, in some papers, such as (Arlotta, Crescenzi, & Mecca et al., 2003) and (Song, Giri, & Ma, 2004), researchers have paid more attention to the influence of semantic information on deep web.
Jufeng Yang et al. (Yang, Shi, Zheng & Wang, 2007) has published his literature on data extraction in which web page is converted into a tree, in which the internal nodes represent the structure of the page and the leaf nodes preserve the information and compared with configuration tree. Moreover, structure rules are used to extract data from HTML pages and the logical and application rules are applied to correct the extraction results. The model has four layers, among which the access schedule, extraction layer and data cleaner are based on the rules of structure, logic and application. Proposed models are tested to three intelligent system i.e. scientific paper retrieval, electronic ticket ordering and resume searching. The results show that the proposed method is robust and feasible.

**Iterative Deepening Search**

Generally the dynamic web search interface generates some output if they are given with some input. The generation of output by some input can be visualized as a graph which is based on the keyword relationship. After getting the interface information about the targeted resource, primary query keywords are applied to generate new keywords, which can be used for further extraction. Iterative deepening search (Ibrahim, Fahmi, Hashmi & Choi, 2008) is proposed by Ahmad Ibrahim et al. His work is based on probability, iterative deepening search and graph theory. It has two phases. The first phase is about classification or identification of a resource behind search interface into some certain domain and in the second phase, each resource is queried according to its domain. Even if the deep web contains a large database but there is need of efficient technique for extracting information from deep web in relatively short time. Presently most of the techniques does not work on real time domain and lot of time is consumed in processing to find the desired result.

**Object Matching Method**

Object matching process has vital role for integration of deep web sources. For integration of database information, a technique was proposed in (Hernandez et al., 1995) to identify the same object from variety of sources using well defined specific matching rules. This paper gives the solution of the merge/ purge problem i.e. the problem of merging data from multiple sources in an effective manner. The sorted neighborhood method is proposed for the solution of merge/purge problem. An alternative technique based on clustering method is also proposed with the comparison with sorted neighbor method.

**String transformation Based Method**

A technique to compare the same parameters of similar objects was proposed in literature (Tejada, Knoblock & Minton, 2002) through string transformation which is independent of application domain but uses application domain to gain the knowledge about attributes of weights through a very little user interaction. There are several future research areas with regard to this algorithm such as how to minimize the noise in the labels given by the user.

**Training Based Method**
A technique PROM was defined in (Doan, Lu, Lee & Han, 2003) to increase the accuracy of matching by using the constraint among the attributes available through training procedure or expert domain. It uses the objects from different sources having different attributes. Using the segmentation of pages into small semantic blocks defined on basis of HTML tags. The future work in this regard to implement the profilers generated in matching task to other related matching task to see the effect of transferring such knowledge.

**Block Similarity Based Method**

A technique proposed by (Ling, Liu, Wang, Ai & Meng, 2006), which segment pages into small semantic blocks based on html tags and change the problem of object matching into problem of block similarity. The method is based on high accuracy record and attributes extraction. Due to the limitation of existing information extraction technology, extracted object data from html pages is often incomplete.

**Text Based Method**

A new method of object matching is proposed by (Zhao, Lin, Fang & Cui, 2007) which is text-based standard TF/IDF cosine-similarity calculation method to calculate the object similarity, and further expended his framework to record-level object matching model, attribute-level object matching model and hybrid object matching model, which considers structured and unstructured features and multi-level errors in extraction. This paper compare the performance of the unstructured, structured and hybrid object matching models and concludes that hybrid method has the superior performance.

**RELEVANCE RANKING**

Surface web crawlers normally do the page-level ranking but this does not fulfill the purpose of vertical search for entity oriented. The need of entity level ranking for deep web resource has initiated a large amount of research in the area of entity-level ranking. Previously most of the approaches concentrate on the ranking the structured entities based on the global frequency of relevant documents or web pages. Method of relevance ranking can be taxonomically categorized in the following parts.

**Data Warehouse Based Method**

Many researchers such as (Nie, Ma, Shi, Wen & Ma, 2007) initiated the use of web data warehouse to pre-store all of entities having the capability of handling structured queries. This paper proposed various language models for web object retrieval such as an unstructured object retrieval model, structural object retrieval model and hybrid model having both structured and unstructured features. This paper concludes that hybrid model is the superior one with extraction errors at changing labels.

**Global Aggregation and Local Scoring Based Method**

A data integration methods based on the local uncertainties of entities is proposed in various literature. But nearly all of the method does not have the capability for local scoring
of entities or aggregation of variety of web sources in a global environment. Literature survey indicates that various search engines are built for focusing on clear indication of entity type and context pattern in user request as illustrated in reference (Cheng, Yan & Chang, 2007). This paper concentrates on the ranking of entities by extracting its underline theoretical model and producing a probabilistic ranking framework that can be able to smoothly integrate both global and local information in ranking.

One of the latest techniques named as LG-ERM proposed by (Kou, Shen, Yu & Nie, 2008) for the entity-level ranking based on the global aggregation and local scoring of entities for deep web query purpose. This technique uses large number of parameters affecting the rank of entities such as relationship between the entities, style information of entities, the uncertainty involved in entity retrieval and the importance of web resources. Unlike traditional approaches, LG-ERM considers more rank influencing factors including the uncertainty of entity extraction, the style information of entities and the importance of Web sources, as well as the entity relationship. By combining local scoring and global aggregation in ranking, the query result can be more accurate and effective to meet users’ needs. The experiments demonstrate the feasibility and effectiveness of the key techniques of LG-ERM.

FEW PROPOSED PROTOCOL FOR DEEP WEB CRAWLING PROCESS

Some examples of frameworks designed for extraction of deep web information are given below.

Search/Retrieval via URL

Search/Retrieval via URL (SRU) protocol is a standard XML-focused search protocol for internet search queries that uses contextual query language (CQL) for representing queries. SRU is very flexible. It is XML-based and the most common implementation of SRU via URL, which uses the HTTP GET for message transfer. The SRU uses the representational state transfer (REST) protocol and introduces sophisticated technique for querying databases, by simply submitting URL-based queries For example URL?version=1.1&operation=retrieve &query=dilip&maxRecords=12
This protocol is only considered useful when the information about the resource is predictable i.e. the query word is already planned from any source. With reference to our previous discussion the task of SRU is comes under values allotment and data extraction.

Z39.50

Z39.50 (ANSI/NISO Z39.50, Information Retrieval: Application Service Definition and Protocol Specification, 2003) is an ANSI/NISO standard that specifies a client/server-based protocol for searching and retrieving information from remote databases. Clients using the Z39.50 protocol can locate and access data in multiple databases. The data is not centralized in any one location. When a search command is initiated, the search is normally sent simultaneously in a broadcast mode to the multiple databases. The results received back are then combined into one common set. In a Z39.50 session, the Z39.50 client software that initiates a request for the user is known as the origin. The Z39.50 server software system that
responds to the origin’s request is called the target. This protocol is useful for extracting data from multiple sources simultaneously but here the search phrase must also be defined using any other knowledge source. The task of the protocol can be considered in our values allotment and data extraction phase.

**OPEN ARCHIVES INITIATIVE PROTOCOL FOR METADATA HARVESTING**

The open archives initiative (OAI) (The Open Archives Initiative Protocol for Metadata Harvesting (Protocol Version 2.0), 2003) protocol for metadata harvesting (OAI-PMH) provides an interoperability framework based on the harvesting or retrieval of metadata from any number of widely distributed databases. Through the services of the OAI-PMH, the disparate databases are linked by a centralized index. The data provider agrees to have metadata harvested by the service provider. The metadata is then indexed by the harvesting service provider and linked via pointers to the actual data at the data provider address. This protocol has two major drawbacks. It does not make its resources accessible via dereferencable URIs, and it provides only limited means of selective access to metadata. This protocol not only provides the proper values allotment for query but also provides knowledge harvested from different sources so it increases the accuracy by decreasing the unmatched query load.

**PROLEARN QUERY LANGUAGE**

The ProLearn Query Language (PLQL) developed (Campi, Ceri, Duvall, Guinea, Massart, & Ternier, 2008) by the PROLEARN "Network of Excellence", is query language, for repositories of learning objects. PLQL is primarily a query interchange format, used by source applications (or PLQL clients) for querying repositories or PLQL servers. PLQL has been designed with the aim of effectively supporting search over LOM, MPEG-7 and DC metadata. However, PLQL does not assume or require these metadata standards. PLQL is based on existing language paradigms like the contextual query language and aims to minimize the need for introducing new concepts.

For a given XML binding form, all relevant metadata standards for learning objects, it was decided to express exact search by using query paths on hierarchies by borrowing concepts from XPath. Thus, PLQL combines two of the most popular query paradigms, allowing its implementations to reuse existing technology from both fields i.e. approximate search (using information retrieval engines such as Lucene) and exact search (using XML-based query engines). This is simple protocol and work as a mediator for transforming data extracted from one object to other. It is applicable in the phase of querying the resource with predefined query words.

**HOST LIST PROTOCOL**

The Host-List Protocol (HLP) model (Khattab, Fouad, & Rawash, 2009) is a periodical script designed to provide a way to inform web search engines about hidden hosts or unknown hosts. The virtual hosting feature, applied in apache web server allows one Apache installation to serve different actual websites. This virtual hosts feature will be the target during the design process of this model. The algorithm of the HLP model is such that it extracts hidden hosts, in the form of virtual hosts from apache web server using one of the PHP scripting language based open source technologies which is utilizing an open standard technology in the form of XML language, building a frontier of extracted hosts then sending
such hosts frontier to the web search engines that support this protocol via HTTP request in an automatic fashion through a cron job. Hosts frontier is an XML file that lists virtual hosts extracted from the configuration file of the apache web server "httpd.conf" after verifying its configuration to take a decision about from where to extract virtual hosts from "httpd.conf". This protocol is designed to reduce the task or identifying virtual host on any server. It generate host list in XML format for crawler and provide the path for data extraction.

REALLY SIMPLE SYNDICATION

RSS stands for "Really Simple Syndication" (Grossnickle et al., 2005). It is a technique to easily distribute a list of headlines, update notices, and sometimes content to a wide number of people. It is used by computer programs that organize those headlines and notices for easy reading. Most people are interested in many websites whose content changes on an unpredictable schedule. RSS is a better technique to notify the new and changed contents. Notifications of changes to multiple websites are handled easily, and the results are presented to user are well organized and distinct from email. RSS works through the website author to maintain a list of notifications on their website in a standard way. This list of notifications is called an "RSS Feed". Producing an RSS feed is very simple and lakhs of websites like the BBC, the New York Times and Reuters, including many weblogs now providing this feature. RSS provides very basic information to do its notification. It is made up of a list of items presented in newest to oldest order. Each item usually consists of a simple title describing the item along with a more complete description and a link to a web page with the actual information being described. This mechanism of extracting contents is very effective for site updating there contents daily but the loop whole is again generating proper feed from combination of database and generated page links. It comes into the category of values allotments and data extraction.

SITEMAP PROTOCOL

The sitemaps protocol (Sitemaps, 2009) allows a webmaster to inform search engines about URLs on a website that are available for crawling. A sitemap is an XML file that lists the URLs for a website. It allows webmasters to include additional information about each URL such as when it was last updated, how often it changes and how important it is in relation to other URLs in the website. This allows search engines to crawl the site more intelligently. Sitemaps are a URL inclusion protocol and complement robots.txt, a URL exclusion protocol. Sitemaps are particularly advantageous on websites where some contents of web site is not linked with public pages and webmasters use rich Ajax or Flash contents that are not normally processed by search engines. Sitemaps helps to find out the hidden contents when submitted to crawler and it do not replace the existing crawl-based mechanisms that search engines already use to discover URLs. Sitemap protocol will come under the taxonomical categorization of values allotment and data extraction. Table 3 depicts the comparison of different protocols for deep web information extraction.

<table>
<thead>
<tr>
<th>Name</th>
<th>Techniques</th>
<th>Advantages</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRU</td>
<td>Uses the REST protocol, send encoded query words through http get request.</td>
<td>Simple xml based request. Independent of underlining database.</td>
<td>Limitation of 256 character query. Responding server must be equipped for analyzing query.</td>
</tr>
<tr>
<td>Protocol</td>
<td>Description</td>
<td>Advantages</td>
<td>Limitations</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Z39.50</td>
<td>ANSI/NISO client/server-based protocol.</td>
<td>The data can be extracted from multiple locations. Access data from multiple databases. Various results combined into one result set, regardless of their original format.</td>
<td>Complex technique and the searching is limited to the speed of the slowest server, No updates and support.</td>
</tr>
<tr>
<td>OAI-PMH</td>
<td>XML response over http.</td>
<td>Disparate databases are linked by a centralized index, Simple http based request. Much faster and independent of database.</td>
<td>Evaluation on xml. limitation in object access, exchange and transfer, No technique available for the user to know when the metadata was last harvested.</td>
</tr>
<tr>
<td>PROLEARN QUERY LANGUAGE</td>
<td>XML XQuery and Xpath based</td>
<td>Best suited for approximate search and exact search, supports hierarchical metadata structures.</td>
<td>Complex to implement and application development model is not mature.</td>
</tr>
<tr>
<td>HOST LIST PROTOCOL</td>
<td>Apache and xml</td>
<td>Retrieval of url from virtual host.</td>
<td>Extracting hidden hosts are limited to the root.</td>
</tr>
<tr>
<td>RSS</td>
<td>XML</td>
<td>Short bunch of information, real time update, generation of RSS and reading mechanism is easy to implement.</td>
<td>Dependent upon site administrator to generate these feed. Because auto feed generator will read only anchor tag.</td>
</tr>
<tr>
<td>SITEMAP PROTOCOL</td>
<td>XML based, Site administrator’s protocol</td>
<td>Easy to generate, Simple structure and node hierarchy.</td>
<td>Site administrator’s involvement is needed, if the pages are hidden by Ajax or flash, explicit mentioning must required.</td>
</tr>
</tbody>
</table>

**Table 3.** Comparison of different protocols in the context of deep web information extraction

**COMPARATIVE STUDY OF DIFFERENT SEARCH ENGINES**

We have conducted search on different search engines some of them are surface web search engine which crawler frontier that retrieves only anchor tag and other are deep search engines which retrieves deep web information. The results are shown below in Table 4 and Table 5. From the analysis of these results, it is clear that surface search engines show more number of results compared to deep web search engine because still at present deep web search engine are not more efficient due to the lack of technological advancement and standards. Results count for query words related to technical literature are relatively more in number deep web search engines while results count of general query words are more in surface web search engines due to the numerous public posting and search engine optimization work. Figure 4 depicts the variation of results count versus query words for different search engines. Figure 5 depicts the snap shots showing results by different surface web search engines in response
to query words. Figure 6 depicts the snap shots showing results by different deep web search engines in response to query words.

Table 4. Query words versus Results counts for surface web search engines

<table>
<thead>
<tr>
<th>Query words</th>
<th>Google.com</th>
<th>Yahoo.com</th>
<th>Bing/live.com</th>
<th>Ask .com</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloud computing</td>
<td>34,200,000</td>
<td>1,510,000</td>
<td>15,300,000</td>
<td>13,100,000</td>
</tr>
<tr>
<td>global warming</td>
<td>30,200,000</td>
<td>2,800,000</td>
<td>14,100,000</td>
<td>7,582,000</td>
</tr>
<tr>
<td>optical modulation</td>
<td>1,800,000</td>
<td>57,800</td>
<td>1,600,000</td>
<td>1,290,000</td>
</tr>
<tr>
<td>walmart</td>
<td>20,800,000</td>
<td>194,000,000</td>
<td>16,300,000</td>
<td>3,340,000</td>
</tr>
<tr>
<td>best buy</td>
<td>296,000,000</td>
<td>44,200,000</td>
<td>551,000,000</td>
<td>188,000,000</td>
</tr>
<tr>
<td>Dictionary</td>
<td>191,000,000</td>
<td>7,880,000</td>
<td>68,400,000</td>
<td>27,400,000</td>
</tr>
<tr>
<td>Astrology</td>
<td>25,300,000</td>
<td>40,400,000</td>
<td>22,600,000</td>
<td>7,790,000</td>
</tr>
<tr>
<td>Insurance</td>
<td>363,000,000</td>
<td>41,600,000</td>
<td>262,000,000</td>
<td>49,730,000</td>
</tr>
</tbody>
</table>

Table 5. Query words versus results counts for deep web search engines

<table>
<thead>
<tr>
<th>Query words</th>
<th>science.gov</th>
<th>deepdyve.com</th>
<th>biznar.com</th>
<th>worldwide.science.org</th>
<th>complete.planet.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloud computing</td>
<td>32,318,709</td>
<td>1,229,954</td>
<td>108,609</td>
<td>52,763</td>
<td>5000</td>
</tr>
<tr>
<td>global warming</td>
<td>14,436,593</td>
<td>563,694</td>
<td>380,240</td>
<td>214,452</td>
<td>3702</td>
</tr>
<tr>
<td>optical modulation</td>
<td>1,774,195</td>
<td>1,513,665</td>
<td>116,407</td>
<td>442,284</td>
<td>621</td>
</tr>
<tr>
<td>Walmart</td>
<td>5,960,514</td>
<td>87,143</td>
<td>234,180</td>
<td>244</td>
<td>41</td>
</tr>
<tr>
<td>best buy</td>
<td>315,040,211</td>
<td>72,574</td>
<td>2,613,762</td>
<td>111,272</td>
<td>2208</td>
</tr>
<tr>
<td>dictionary</td>
<td>56,975,239</td>
<td>101,835</td>
<td>397,173</td>
<td>122,385</td>
<td>1087</td>
</tr>
<tr>
<td>astrology</td>
<td>14,101,342</td>
<td>87,964</td>
<td>132,384</td>
<td>3,404</td>
<td>290</td>
</tr>
<tr>
<td>insurance</td>
<td>92,056,043</td>
<td>170,433</td>
<td>2,795,986</td>
<td>291,304</td>
<td>5000</td>
</tr>
</tbody>
</table>

Figure 4: Variation of results count versus query words for different search engines
Figure 5: Snap shots showing results by different surface web search engines in response to query words
CONCLUDING REMARKS

After going through the exhaustive literature survey, a general framework is developed for understanding the entire web crawling mechanisms and to find out the limitations of general web crawling mechanism in searching the deep web. The entire working of web crawling mechanisms in general is divided into four parts in the developed framework. The present web crawling mechanisms are analyzed and merged as per developed framework to understand their advantages and limitations in searching the deep web. Some existing deep web information extraction protocols are analyzed with their comparative study. We have also done a comparative study of surface web search engines and deep web search engines.

In query interface analysis, heuristic rules based techniques are the simplest one but does not guarantee even a solution. IRFA eliminates the problem of classification of high dimensional and sparse search interface data. Future work is to identify other techniques for feature waiting for generation of random forest. Various techniques of search form schema mapping are described and can be categorized as heuristic and formal techniques. In this regard an appropriate measure has to be defined to filter out false matching. In domain ontology identification the main area to be focused for research is concept relation learning to increase the efficiency of the system. In values allotment there is a scope for improvement by using hierarchy to detect more hypernymy relationship. In fuzzy comprehensive evaluation methods, the future research area include to find a model to detect the optimal configuration.
parameter to produce the results with high accuracy but MrCOM model has good fitness with highly accurate query strategy selection. In response analysis and navigation, most of the techniques do not work on real time domain and a large time is consumed in processing to find the desired result. In relevance ranking one of the latest techniques is LG-ERM which considers a larger number of ranking influencing factors such as the uncertainty of entity extraction, the style information of entity and the importance of web sources as well as entity relationship. By combining the techniques of local scoring and global aggregation in ranking, the query result can be made more accurate and effective. There exists an improper mapping problem in the process of query interface analysis. Over traffic load problem arises from unmatched semantic query. Data integration suffers due to the lack of cost models at the global level in the process of proper values allotment. In distributed websites there is challenge for query optimization and proper identification of similar contents in the process of response analysis and navigation. The ranking of deep web contents are difficult due to the lack of reference to other sites.

On the basis of taxonomic classification and consequent analysis of different algorithms a conclusion is made that an open framework based deep web information extraction protocol is required to eliminate the limitations of present web crawling techniques in searching the deep web. Future work includes the design, analysis and implementation of open framework protocol for deep web information extraction, considering the fact that it must allow simple implementation without much modifying present architecture of web.

REFERENCES


http://www.sitemaps.org


**Dilip Kumar Sharma** is B.Sc, B.E.(Honors)(CSE), M.Tech.(IT), M.Tech. (CSE) and pursuing Ph.D in Computer Engineering. He is life member of CSI Mumbai, IETE New Delhi, ISTE New Delhi, ISC Kolkata, SSI and member of CSTA, USA. He has 18 short term courses/workshops/seminars organized by various esteemed originations such as IIT Kanpur, IIT Roorkee, IIT Delhi, IIT Kharagpur, Microsoft, WIPRO, IBM, TCS etc. He has published 18 research papers in International Journals/Conferences of repute and participated in 18 International/National conferences. Presently he is working as Reader in Department of Computer Science & Engineering (NBA Accredited) at GLA Institute of Technology & Management, Mathura (U.P.),(AICTE Approved & Affiliated to U.P. Technical University, Lucknow) since March 2003. His research interests are deep web information retrieval, e-commerce, Digital Watermarking and Software Engineering. He has guided various projects and seminars undertaken by the students of undergraduate/postgraduate.

**Prof. A. K. Sharma** received his M.Tech. (Computer Sci. & Tech) with Hons. From University of Roorkee (Presently I.I.T. Roorkee) in the year 1990 and Ph.D (Fuzzy Expert Systems) from JMI, New Delhi in the year 2000. From July 1992 to April 2002, he served as Assistant Professor and became Professor in Computer Engg. at YMCA Institute of Engineering, Faridabad, Haryana in April 2002. He obtained his second Ph.D. in Information Technology form Indian Institute of Information Technology & Management, Gwalior in 2004. Presently he is working as Dean, Faculty of Engineering and Technology at YMCA University of Science and Technology, Faridabad.

His research interest includes Fuzzy Systems, Object Oriented Programming, Knowledge Representation and Internet Technologies. He has guided 8 Ph.D. thesis and 8 more are in progress with about 175 research publications in International and National journals and conferences. The author of 7 books, is actively engaged in research related to Fuzzy logic, Knowledge based systems, MANets, Design of crawlers. Besides being member of many BOS and Academic councils, he has been Visiting Professor at JMI, IIFT&M, and I.I.T. Roorkee. He is also project coordinator of Technical Education Quality Improvement Program under World Bank scheme.