Survey of Techniques for Deep Web Source Selection and Surfacing the Hidden Web Content

Khushboo Khurana
Assistant Professor, Shri Ramdeobaba College of Engineering and Management, Nagpur, India

Dr. M. B. Chandak
Professor, Shri Ramdeobaba College of Engineering and Management, Nagpur, India

***Abstract*—Large and continuously growing dynamic web content has created new opportunities for large-scale data analysis in the recent years. There is huge amount of information that the traditional web crawlers cannot access, since they use link analysis technique by which only the surface web can be accessed. Traditional search engine crawlers require the web pages to be linked to other pages via hyperlinks causing large amount of web data to be hidden from the crawlers. Enormous data is available in deep web that can be useful to gain new insight for various domains, creating need to access the information from the deep web by developing efficient techniques. As the amount of Web content grows rapidly, the types of data sources are proliferating, which often provide heterogeneous data. So we need to select Deep Web Data sources that can be used by the integration systems. The paper discusses various techniques that can be used to surface the deep web information and techniques for Deep Web Source Selection.**

Keywords—Deep Web; Surfacing Deep Web; Source Selection; Deep Web Crawler; Schema Matching

I. INTRODUCTION

Tremendous increase in collection of web content has created new opportunities for large-scale data analysis. Working of search engine is based on index creation by crawling web pages. The web crawler retrieves the contents of web pages and parses the web pages to get the data and hyperlinks. It then continues to crawl the found hyperlinks. Parsed data is sent to the indexer and stored in database. A search is performed by referring the search engine database, consisting of web page index [1].

Most of the search engines access only the Surface Web, which is a part of web that can be discovered by following hyperlinks and downloading the snapshots of pages for including them in the search engine's index [2]. As a traditional search engine crawler requires pages to be linked to other pages via hyperlinks or the page must be static, large amount of web data is hidden. Hidden web is also referred as Deep Web.

The deep Web is qualitatively different from the surface web. The term "Deep Web" refers to web pages that are not accessible to search engines. The existing automated web crawlers cannot index these pages, thus they are hidden from the Web search engines [3].

The data in digital libraries, various government organizations, companies is available through search forms. A deep web site is a web server that provides information maintained in one or more back-end web databases, each of which is searchable through one or more HTML forms as its query interfaces [4]. Deep web consists of following types of content:

- **Dynamic Data**: Data that can only be accessed through the query interface they support. These interfaces are based on input attribute(s), and a user query involves specifying value(s) for these attributes. In response to such a query, dynamically generated HTML pages returned as the output, comprising output attributes [5].
- **Unlinked Content**: Data that is not available during link analysis done by web crawlers.
- **Non-Text Content**: Various multimedia files, PDF and non-HTML documents.

The information in the deep web is about 500 times larger than the surface web, with 7,500 Terabytes of data, across 200,000 deep web sites [6]. This wealth of information is missed since the standard search engines cannot find the information generated by dynamic sites. So, there is a need to access the data that is deep by developing efficient techniques.

II. TRADITIONAL WEB CRAWLER

Traditional web crawlers are used to index the surface web. Fig. 1 shows the working of a traditional crawler. Initially a URL is selected as the start for the web crawler. Crawler then retrieves the web pages. From the web pages, data is extracted and resource discovery is done to extract the hyperlinks, which are further processed. Data is sent to the indexer which is used as an index during search. Hyperlinks are used as the new URL and the loop continues. Traditional crawler does not distinguish between pages with and without forms, structured and semi-structured data cannot be retrieved; hence form processing phase has to be added to the web crawler loop, to access the data present in the dynamic pages and web databases.

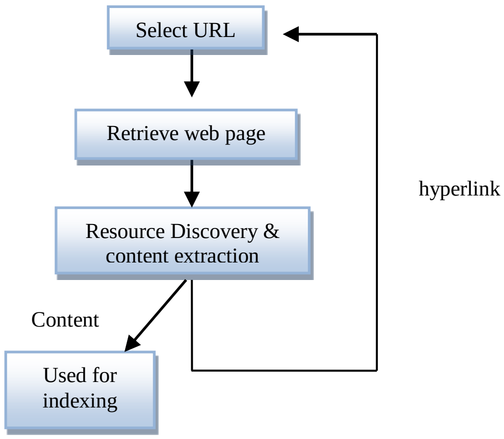

Fig. 1. Working of traditional crawler

III. ACCESSING THE DEEP WEB

Deep web data can be accessed by surfacing the web data that is not accessible to the traditional search engines. Following are the major steps to access the deep web content:

Step 1: Find the Data sources.

Step 2: Data source Selection

Step 3: Send the selected data source to the Data Integration System.

Data sources for accessing the deep web may be web databases, web servers and many other dynamic web pages. Depending upon the integration system, the data sources can be added. But all sources should not be included in the Integration System. The disadvantage of including all found data sources are as follows:

- Redundant data may be added
- Irrelevant data may be added reducing the overall quality of the Data Integration System
- Low quality data can be included
- High cost of including data since, networking and processing cost are associated with including a data source in the integration system.

Various deep web source selection algorithms are discussed in section (IV). The data sources or the deep web content can be accessed by one of the following techniques as shown in fig. 2.

a) Web Surfacing by Hidden Web Crawlers by form processing and querying the Web databases:

Huge amount of data is present in the hidden web and to access this data from the deep web sites, forms need to be filled and submitted, to get the data from the web databases. Deep web crawlers are discussed in section (V). General deep web crawlers do a breadth search on the deep web, for retrieving general web data; whereas vertical deep web crawlers do a depth based search, focusing on a particular domain to extract the deep web sites based on a specific topic.

b) Schema Matching for Web source Virtual Integration system

In schema matching, instead of filling the form of the deep web site and then extracting the data to find if they are relevant to the search, a schema of the required data is prepared and only those sites which match the schema are retrieved. This technique greatly reduces the cost of extraction of web pages and then processing them. Schema matching can be done by web source virtual integration system as discussed in section (VI).

c) Data extraction by deep web search using various techniques such as Data Mining

Various techniques can be used to extract relevant information from the deep web (Refer Section VII). In Vision based approach the web page is assumed to be divided into sections that contain particular type of information. Rather than extracting the complete web page information and then parsing it, only the section that contains the relevant information is extracted using this technique.

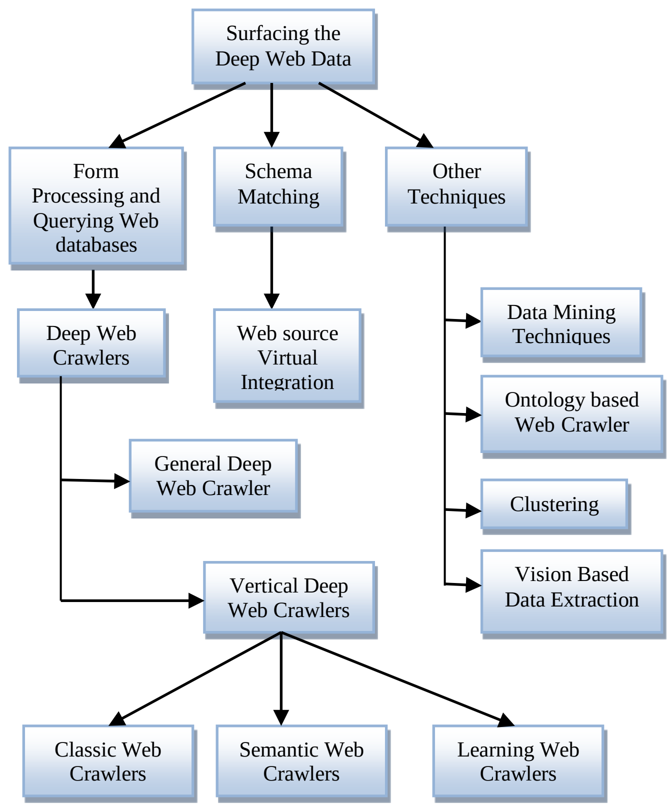

Fig. 2. Surfacing Hidden Web

IV. DATA SOURCE SELECTION BASED ON QUALITY PARAMETER

There may be hundreds and thousands of deep web data sources providing data for particular domain. The user may not want to include all the available data sources in the data integration system as there may be large number of data sources that may be of low quality. The data source selection can be broadly summarized to have following steps:

- Define quality dimensions for deep web
- Define the quality assessment model for deep web source.
- Depending on the source quality order the data sources
- Consider the highest quality set of deep web sources based on threshold.

After the web pages are extracted by the web crawler or using schema matching technique or any other technique, the web pages are checked for quality to decide whether the web page must be included or not. Fig. 3 illustrates this concept.

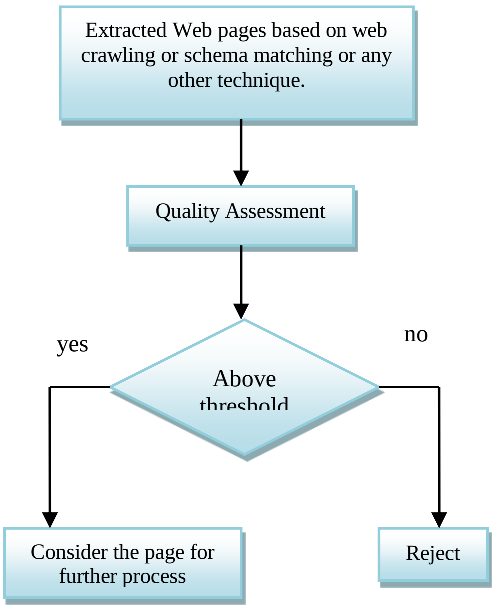

Fig. 3. Quality based Data Source Selection

In [7], an effective and scalable approach for selection of deep web source based on quality of data source is proposed for the deep web data integration system. Highest quality set of deep web sources related to particular domain are found by evaluating the quality dimensions representing the characteristics of the deep web source. Completeness, consistency, size, response time and available services are the quality dimensions considered.

The amount and type of data sources are proliferating. Data sources often provide heterogeneous or conflicting data, so we need to resolve data conflicts. There are several advanced techniques that consider accuracy of sources, freshness of sources and dependencies between sources to solve the conflicts. To improve the data fusion, a quality estimation model of Deep Web data sources (DSQ) is proposed in [8]. According to the characteristics of data fusion, the estimation model selects three dimensions of factors-data quality, interface quality and service quality as estimation criteria, and estimates the quality of data sources.

C. Hicks, et. al. [9], have proposed a new paradigm for discovery and cataloging of deep web sites. The approach divides the discovery into three phases. The first phase discovers potential deep Web sites based on a crawler configuration file for a given domain. The second phase consists of generating a set of probing queries using simple domain knowledge supplied in the query configuration file. If the submission of a probing query to a potential deep Web site is successful, the result page will be analyzed. If the result page contains the data types as specified, the site can be marked as successful identification. The third phase consists of creating a catalogue entry for that site. For this initial prototype, the catalogue entry can be as simple as the URL and the set of form parameters together with the associated values that are required for the successful query submission.

The process of data source selection can be automated by periodically analyzing different deep web sources and user can be given recommendations about a small number of data sources that will be most appropriate for their query. A data mining method to extract a high-level summary of the differences in data provided by different deep web data sources is proposed in [10]. Pattern of values are considered with respect to the same entity and a new data mining problem is formulated, referred as differential rule mining. An algorithm for mining such rules is developed. It includes a pruning method to summarize the identified differential rules. For efficiency, a hash-table is used to accelerate the pruning process.

V. Deep Web Crawler

Deep Web Crawlers are similar to traditional crawlers, but traditional crawlers do not distinguish between pages with and without forms. The results provided by the search engine are based on the copy of its local index. If additional steps are added to process pages, on which forms are detected and extraction of hidden information in databases is done then the crawler is termed as Hidden/ Deep Web Crawler [11], [4]. A user accesses the data in the Hidden Web by issuing a query through the search form provided by the web site, which in turn gives a list of links to relevant pages on the Web. The user then looks at the obtained list and follows the associated links to find interesting pages. Resource discovery and data extraction are the main task of Deep Web Crawler.

Fig. 4 shows the working of deep web crawler by addition of some extra steps. In this, the retrieved pages are checked if they have form. If form is present then it is processed to build an internal representation. Forms are filled with untried values and submitted. The returned page is then analyzed to check if a valid search result was returned. If the response page contains hypertext links, these are followed and the loop continues. Deep Web crawlers enable indexing, analysis and mining of hidden web content. The extracted content can then be used to categorize and classify the hidden databases. The Hidden Web crawler automatically process the search forms after downloading it form the hidden web site and submit the filled form so as to download the response pages which can then be used with existing index structures of the search engine.

To extract value from millions of HTML forms that span many languages and hundreds of domains, various deep web crawlers are designed. There are large numbers of forms that have text inputs and require valid input values to be submitted. In [12], an algorithm is presented for selecting input values for text search inputs that accept keywords and an algorithm for identifying inputs which accept values of specific type. HTML forms often have more than one input and hence very large number of URLs can be generated. An algorithm navigates the search space of possible input combinations to identify only those that generate URLs suitable for inclusion into the web search index.

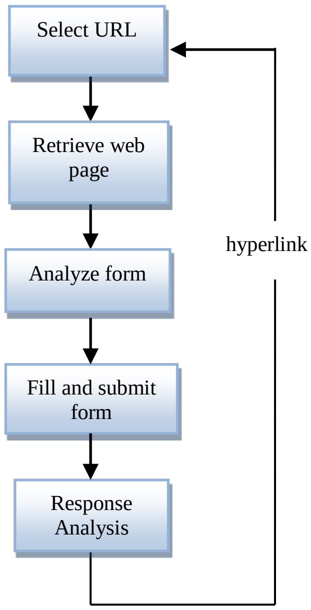

Fig. 4. Working of Deep web crawler

To meet the needs of deep web search, in [13], a new structure of crawler is designed to have innovative parts such as the mainframe extracting module and the algorithm to distinguish different websites with the same URL using improved Bayesian classification and to expand the function to AJAX form dealing. Dom Tree is also used to make easier and more visual analysis and treatment of downloaded web pages.

K.Bharati, P.Premchand, et. al., have proposed an effective design of a vertical Hidden Web Crawler that can automatically discover pages from the Hidden Web by employing multi-agent Web mining system. A framework for deep web with genetic algorithm is designed for resource discovery problem and the results show improvement in the crawling strategy and harvest rate. The focused crawler URL analysis model based on improved genetic algorithm proposed in this paper can improve accuracy rate, recall rate effectively, and avoid getting into the local optimal solution [14].

An entity extraction system, which extracts data from Deep Web automatically, is presented in [15]. A web crawler based on the characteristics of Deep Web is designed. Non- standard pages are normalized and the entity data from Deep Web are located and extracted accurately, based on the hierarchy and layout features in DOM tree, combined XPath with Regular Expression to locate entity data. Then the extracted entity attributes and attribute values are stored.

Crawling the Deep Web is requires huge amount of computing resources, but most of search engine companies hardly meet the needs. A design of the Grid-based middleware, OGSA-DWC for crawling the Deep Web is proposed in [16]. With the middleware, a Grid-based Deep Web crawling system can be implemented. It is based on two functions: Search Form Collecting and Deep Web Crawling.

A significant portion of deep web sites, including almost all online shopping sites, curate structured entities as opposed to text documents. Although crawling such entity-oriented content is clearly useful for a variety of purposes, existing crawling techniques optimized for document oriented content are not best suited for entity-oriented sites.

In [17], a prototype system is built that specializes in crawling entity-oriented deep web sites. A sub-problem is tailored to tackle important sub problems including query generation, empty page filtering and URL de-duplication in the specific context of entity oriented deep web sites. All information on the web is not in document or structured form. Multimedia data is also available is huge amount. Images can be a good source of information extraction from the deep web.

A focused crawler can miss a relevant page if there does not exist a chain of hyperlinks that connects one of the starting pages to that relevant page. Also, unless all the hyperlinks on the chain point to relevant pages, the crawler will give up searching in the relevant direction before it reaches the final target. Because of this limitation, crawlers using local search algorithms can only find relevant pages within a limited sub-graph of the Web that surrounds the starting URLs and any relevant pages outside this sub-graph will be ignored.

In [18], Tunneling technique is proposed which addresses the problems of local search. It is a heuristic based method that solves simple global optimization problem. A focused crawler using Tunneling will not give up probing a direction immediately after it encounters an irrelevant page. Instead, it continues searching in that direction for a pre set number of steps. This allows the focused crawler to travel from one relevant Web community to another when the number of irrelevant pages between them is within a limit.

Semantic Crawlers [19] are a variation of classic focused crawlers. Download priorities are assigned to pages by applying semantic similarity criteria for computing page-to-topic relevance: a page and the topic can be relevant if they share conceptually (but not necessarily lexically) similar terms. Learning Crawlers can be used to assist, visiting of web pages based on priorities. A learning crawler is supplied with a training set consisting of relevant and not relevant Web pages which is used to train the learning crawler [20] [21]. Higher visit priority is assigned to links extracted from web pages classified as relevant to the topic. Methods based on Context Graphs [22] and Hidden Markov Models (HMM) [23] can be used which consider the page content with the corresponding classification of web pages as relevant or not relevant to the topic, the link structure of the Web and the probability that a given page leading to a relevant page within a small number of steps.

Image extractor for extracting images from the result pages of deep web called AIE is proposed in [24]. Images from deep web result pages are extracted along with the images from the deep web which has no images on the result pages but has images on the detailed data record pages. The extractor can also get the images from the surface web sites which have some relations with the records on deep web. Multimedia data provide large amount of useful information. Using focused vertical crawler, image/video data can be extracted. Relevant videos can be extracted based on video annotations [25,26], based on the domain the crawler is designed for.

Some attributes in Query co-occur and some are exclusive. To generate a valid query, we have to reconcile the key attributes and their semantic relations. To address the problem, a method based on the HTML codes is presented in this paper. Different kinds of semantic containers can be got through analyzing the codes of the query interface. A Query based approach is proposed in [27].

VI. Web Source Virtual Integration

The virtual integration system, also called information mediation system [28], tries to discover relevant web sources to user's query and avoids the user to ask each web source separately using their own vocabulary. The mediation system uses a logical schema based on the source description and called the mediated schema. The source description is all the properties of the data source that the mediated schema need to know to access and to use their data.

Virtual Data Integration System creates specific virtual schema for each domain and map the fields of the search forms in that domain to the attributes of the virtual schema. This enables the user to query over all the resources in its domain of interest just by filling a single search form in the domain. Search systems using such vertical schema are called vertical search engines.

Another vision of a deep web virtual integration system uses a mediated schema built with a relational schema describing each deep web [29]. The paper proposes an approach to extract a relational schema describing a deep web source. The key idea is to analyze two structured information: the HTML Form and the HTML Table extracted from the deep web source to discover its data structure and to allow us to build a relational schema describing it. A knowledge table is also used to take profit of the learning experience on extracting relational schema from deep web source.

To automatically accomplish deep web sources discovery a method is proposed by importing focused crawler. Web sites for domain specific data sources based on focused crawling are selected. These web sites are checked if there exists deep web query interface in the former three depths. Lastly, the deep web query interface is judged to check if they are relevant to the given topic. Importing focused crawling technology makes the identification of deep web query interface locate in a specific domain and capture relative pages to a given topic instead of pursuing high overlay ratios. This method dramatically reduces the quantity of pages for the crawler to identify deep web query interfaces [30].

There are two types of virtual integration approach. The first one is the vertical search engine that integrates the same kind of information from multiple web sources like a flight ticket search engine for all flight companies. And the second one is the topical search portal that integrates all information about a particular domain. For example a topic search portal for travel will provide user data about all what concern our travel organization: flight ticketing, hotel, car rental, monuments to visit, security information etc. [31]

In [32] the authors have designed a conceptually novel approach by viewing schema matching as correlation mining, for the task of matching Web query interfaces to integrate the myriad databases on the Internet. DCM framework, which consists of data preparation, dual mining of positive and negative correlations, and finally matching selection, is proposed. The algorithm cares both positive and negative correlations, especially the subtlety of negative correlations, due to its special importance in schema matching.

Statistical Schema matching is proposed in [33]. A general statistical framework MGS for such hidden model discovery, which consists of hypothesis modeling, generation, and selection, is proposed. The algorithm targets synonym discovery and schema matching, by designing a model that specially captures synonym attributes.

Various other approaches of schema matching such as schema-only based, content based, hybrid and composite matchers are explained in [34].

Holistic Schema Matching (HSM) to find matching attributes across a set of Web database schemas of the same domain is proposed in [35]. HSM takes advantage of the term occurrence pattern within a domain and can discover both simple and complex matching efficiently with-out any domain knowledge.

VII. Data Extraction

A. Data Mining on Deep Web

Data mining on the deep web can produce important insights. For example, to show the price of electronic gadgets from different web sites and offer the customer with the site providing the requested gadget in the lowest price. Data mining on deep web must be performed based on sampling of the datasets. The samples, in turn, can only be obtained by querying the deep web databases with certain inputs. Data Mining is applied on the data that is obtained by querying the deep web database.

In [36], a stratified sampling method to support association rule mining and differential rule mining on the deep web is proposed. A pilot is selected at random from the deep web for identifying interesting rules. Then, the data distribution and relation between input attributes and output attributes are learnt from the pilot random sample. Greedy stratification approach is then applied, which processes the query space of a deep web data source recursively, and considers both the estimation error and the sampling costs. The optimized sample allocation method integrates estimation error and sampling costs.

Dasgupta et al.[6] proposed HDSampler, a random walk scheme over the query space provided by the interface, to select a simple random sample from hidden database.

A novel query-oriented, mediator based biological data querying tool, SNPMiner, is proposed in [5]. It is a domain specific search utility, which can access and collect data from the deep web. The system includes three important components, which are the web server interface, the dynamic query planner, and the web page parser. The web server interface can provide end users a unified and friendly interface. The dynamic query planner can automatically schedule an efficient query order on all available databases according to user's query request. The web page parser analyzes the layout of HTML pages and extracts desired data from those pages.

B. *Clustering*

Clustering can be performed on Web sources to process only domain specific content. A novel method DWSemClust, is proposed in [37] to cluster deep web databases based on the semantic relevance found among deep web forms by employing a generative probabilistic model Latent Dirichlet Allocation (LDA) for modeling content representative of deep web databases. A document comprises of multiple topics, the task of LDA is to cluster words present in the document into topics. Parameter estimation is done to discover the document's topic and tell about its proportionate distribution in documents. Deep web has a sparse topic distribution. Due to this LDA is used as a clustering solution for the sparse distribution of topics.

C. *Ontology Assisted Deep Web Search*

Deep web has huge amount of information, hence the number of relevant Web pages returned might be very less. It is necessary to develop a methodology that increases the number of relevant pages returned by a search. Ontologies can assist the web search, to reduce the number of irrelevant web pages returned by the search engine.

Domain ontologies can be integrated with the web search engine for efficient search. Combining of Deep Web information with ontology is suggested in [38]. The paper considers the problem of constructing domain ontologies that support users in their Web search efforts and that increase the number of relevant Web pages that are returned. A semi-automatic process to interpret information needed against the backdrop of the Deep Web is designed to utilize domain ontologies to meet Web users' needs.

In [39], a novel approach is proposed that combines Deep Web information, which consists of dynamically generated Web pages that cannot be indexed by the existing automated Web crawlers, with ontologies built from the knowledge extracted from Deep web sources. The Ontology based search is divided into different modules. The first module constructs attribute-value ontology. Second module constructs the attribute-attribute ontology. Third module formulate the user query, fills the search interface using domain ontology, extract results by looking into the index database.

G.Liu, K.Liu, et.al., [40] have put forward an automatic method for Deep Web discovery. The information from specific fields of Deep Web entry form is used to establish domain ontology, then the crawler extracts a URL from links queue as the start link, and using Bayesian Classifier to do theme classification. If the page belongs to the theme then the Form viewer module is used to check the HTML code to determine whether these have a form. If form exists in the HTML, then entry found modules are used to calculate weights between the ontology and the attributes of the form, if the value is according to the requirements, the page is downloaded.

Duplicate entity identification is done to discover the duplicate records from the integrated Web databases. However, most of existing works address this issue only between two data sources. That is, one duplicate entity matcher trained over two specific Web databases cannot be applied to other Web databases. In addition, the cost of preparing the training set for n Web databases is higher than that for two Web databases. A holistic solution to address the new challenges posed by deep Web, whose goal is to build one duplicate entity matcher over multiple Web databases is proposed in [41].

D. *Visual Approach*

Web information extraction based on visual approach is programming language independent. This approach utilizes the visual features of the web pages to extract data from deep web pages including data record extraction and data item extraction. There are many semi-automatic and automatic methods for visual based information extraction.

In [42], a vision-based approach is proposed to extract the structured data, including data records and data items automatically, from the deep Web pages. Given a sample deep Web page from a Web database, its visual representation is obtained and it is transformed into a Visual Block tree. Then data records are extracted from the Visual Block tree. The extracted data records are then partitioned into data items and the data items are aligned based on semantics. Finally, visual wrappers for the Web database based on sample deep is generated, such that both data record extraction and data item extraction can be preformed.

A coordinate system can be built for every Web page. The origin locates at the top left corner of the Web page. The X-axis is horizontal left-right, and the Y-axis is vertical topdown. Suppose each text/image is contained in a minimum bounding rectangle with sides parallel to the axes. Then, a text/image can have an exact coordinate (x, y) on the Web page. Here, x refers to the horizontal distance between the origin and the left side of its corresponding rectangle, while y refers to the vertical distance between the origin and the upper side of its corresponding box. The size of a text/ image is its height and width. The coordinates and sizes of texts/images on the Web page make up the Web page layout. Fig. 5a shows a popular presentation structure of deep Web pages and Fig. 5b gives its corresponding Visual Block tree. The technical details of building Visual Block trees can be found in [43]. An actual Visual Block tree of a deep Web page may contain hundreds even thousands of blocks.

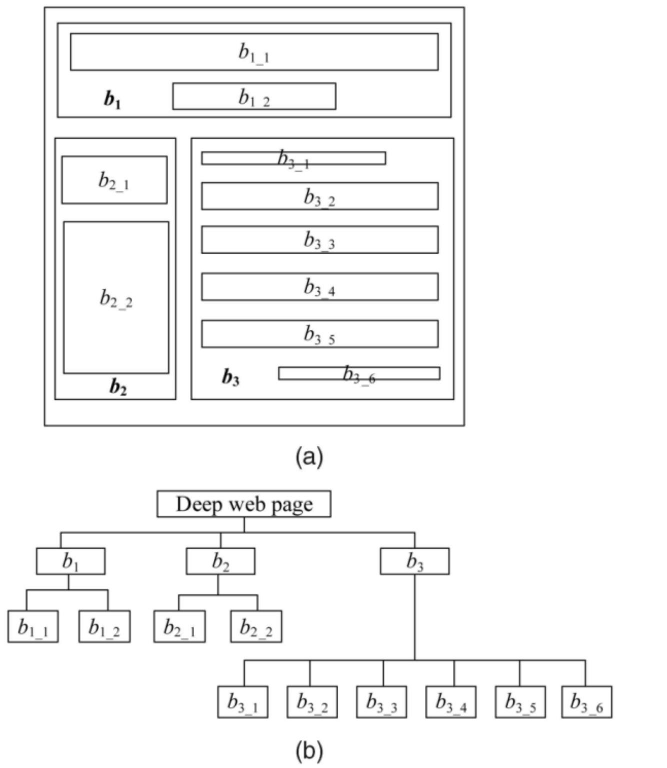

Fig. 5. (a) The presentation structure and (b) its Visual Block tree (Referred from [43])

A page division method is proposed in [44], which divides the pages into separate parts, after analyzing source codes and visual information of pages, into several segments by applying block-division algorithm. After that the parts which don't contain search interfaces are removed. At last topic-specific queries are constructed to obtain results and distinguish deep web interfaces by analyzing the results.

Some of these approaches perform only data record extraction but not data item extraction, such as Omini [45], RoadRunner [46]. These methods do not generate wrappers, i.e., they identify patterns and perform extraction for each Web page directly without using previously derived extraction.

Similar structures are recognized when comparing the differences between two Web pages in [47]. Visual structural information of Web pages are recognized. The technique is based on a classification of the set of html – tags which is guided by the visual effect of each tag in the whole structure of the page. This allows translating the web page to a normalized form where groups of html tags are mapped into a common canonical one. A metric to compute the distance between two different pages is also introduced.

VIII. Comparative Analysis

Table 1 shows the comparative analysis of the two techniques widely used for surfacing the hidden web-form processing and querying the deep web by Hidden Web Crawlers and Schema Matching for Virtual Integration systems. The comparison is based on various parameters such as- technique, type, usefulness, main challenges, cost, storage requirement, advantages, and limitations with their solutions, as discussed in the previous sections.

The major requirement for any of these systems is huge computing requirement. For this Grid based middleware can be used as discussed in section-V.

TABLE I. Comparative Analysis of Surfacing Techniques

<table>
<tr><th>Parameter</th><th colspan="2">Form processing and querying the deep web by Hidden Web Crawlers</th><th colspan="2">Schema Matching for Virtual Integration systems</th></tr>
<tr><td>Technique</td><td colspan="2">1. Give the initial URL to start the process
2. Extract the pages
3. Fill the form and submit
4. Extract the data and index them.</td><td colspan="2">1. Construct the schema based on the requirement.
2. Extract those sites which match the schema
3. Send the web pages to virtual integration system</td></tr>
<tr><td rowspan="3">Types</td><td rowspan="2">General deep web crawlers</td><td rowspan="2">Vertical / focused web crawlers</td><td colspan="2">Various techniques exist for schema matching such as logical, relational, statistical, co-relation based schema matching, element or structure based, content based, etc.</td></tr>
<tr><td>Vertical search engine</td><td>Topical search engine</td></tr>
<tr><td>They perform a breadth search on the deep web, for retrieving general web data.</td><td>They perform a depth based search, focusing on a particular domain to extract the deep web sites based on a specific topic.</td><td>Integrates the same kind of information from multiple web sources.</td><td>Integrates all information about a particular domain.</td></tr>
<tr><td>Use</td><td>Can be used when surfacing for generalized search engine on web rather than a domain specific search engine.</td><td>Vertical crawling can be used to generate data for an individual user.</td><td colspan="2">Used for specific data extraction.</td></tr>
<tr><td rowspan="2">Main challenges / issues.</td><td colspan="2">1. Decide which form inputs to fill when submitting queries to a form
2. To find appropriate values to fill in these inputs.</td><td colspan="2" rowspan="2">1. Design an approach to extract the deep web source description needed by the mediated schema or extract a relational schema describing the deep web source.
2. Design a virtual integration system that accepts the web pages based on the schema.</td></tr>
<tr><td>3. It needs to determine the relevance of a retrieved web page.
4. Design an algorithm that balances the trade-off between number of URLs</td><td>3. Predict and identify potential URLs that can lead to relevant pages.
4. Rank and order</td></tr>
</table>

	and to achieve high coverage of the site's content.	the relevant URLs so the crawler knows exactly what to follow next.	
Cost of web page extraction	Cost of web page extraction is high because in this method the all the deep web pages are extracted following the link analysis.	Cost is less than general web crawler since only pages related to a domain are extracted. Also the probability of an unvisited page being relevant or not is calculated before actually downloading the page.	This technique greatly reduces the cost of extraction of web pages.
Processing cost	All the extracted pages are analyzed to remove the irrelevant data, hence processing cost is more.		Processing cost is less
Storage Requirement	Very high storage is required	Compared to general crawler, the storage requirement is reduced.	Very less storage. Only schema needs to be stored and the extracted pages.
Advantages	Starting from popular seed pages, leads to collecting large-Page Rank pages early in the crawl [48]	Higher density of value pages	Maximum relevant pages are retrieved based on the schema.
Limitations	1. HTML forms typically have more than one input and hence a naive strategy of enumerating the entire Cartesian product of all possible inputs can result in a very large number of URLs being generated. 2. Crawling too many URLs will drain the resources of a web crawler preventing the good URLs from getting crawled, and posing an unreasonable load on web servers hosting the HTML forms. 3. Large number of empty result pages	1. Focused crawlers have a limitation of local search because it may not follow a path that does not have relevant content. (as discussed in section V) 2. Classical focused crawler fail to associate documents with semantically similar but lexically different terms	1. Schema generation for a large domain can be time consuming 2. Dynamic changes, so greedy algorithms may fail. 3. Simple matching 4. Domain knowledge required
Probable Solutions	1. URL de-duplication 2. Use DOM trees and pruning method to reduce the number of URLs to be searched. 3. Combine techniques discussed in section (VII) to improve. Ontologies can be combined with any type of deep web crawler to get only relevant results. 4. Data mining technique, association rule mining, clustering, etc can be used. 5. Tunneling can be a solution to limitation-1 of focused web crawler. Other techniques discussed in section V. 6. Semantic focused crawlers and Learning crawlers can be used for better results. (for limitation-2 of focused crawler)		1. Faster design process for schema and updating to accommodate the dynamic nature. 2. Holistic Schema Matching (HSM), DCM and other techniques discussed in section VI.

IX. Conclusion

The paper discusses the way to extend the traditional web crawlers to surface the Deep Web. Hidden Web content can be accessed by Deep Web Crawlers that can fill and submit forms to query the online databases for information extraction. In this technique the extracted content is analyzed to check if it is relevant. Schema Matching has proved to be an efficient technique for extracting relevant content. Data from the Deep Web can be extracted by applying various techniques such as mining, building ontology to assist domain specific data retrieval. Visual approach is an efficient technique to extract only the required data. The paper also shows the comparative analysis of the two techniques widely used for surfacing the hidden web form processing and querying the deep web by Hidden Web Crawlers and Schema Matching for Virtual Integration systems. Depending upon the application area the surfacing technique can be selected and be combined with other techniques to overcome the drawbacks in the original method.

References

[1] H. T. Yani Achsan, W. C. Wibowo, "A Fast Distributed Focused-web Crawling," Procedia Engineering 69 (2014), pp. 492-499.

[2] M. Bergman, "The deep Web: surfacing hidden value", in the Journal Of Electronic Publishing 7(1) (2001).

[3] Y.J. An, J. Geller, Y.T. Wu, S. Chun, "Automatic Generation of Ontology from the Deep Web," in Database and Expert Systems Applications, 2007. DEXA'07. 18th International Workshop, IEEE, pp. 470-474.

[4] A. Ntoulas, P. Zerfos, J.Cho., "Downloading Textual Hidden Web Content Through Keyword Queries," in Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL'05, Denver, USA, Jun 2005 IEEE , pp. 100-109.

[5] F.Wang, G.Agrawal, R. Jin, and H. Piontkivska, "Snpminer: A domain-specific deep web mining tool," In Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering, 2007. BIBE 2007, IEEE, pp. 192-199.

[6] A. Dasgupta, X. Jin, B. Jewell, N. Zhang, and G. Das, "Unbiased Estimation of Size and Other Aggregates Over Hidden Web Databases," in SIGMOD '10, Proceedings of the 2010 international conference on Management of data, New York, NY, USA, 2010, ACM, pp. 855–866.

[7] X.F. Xian, P.P. Zhao, W. Fang, J. Xin, "Quality Based Data source selection for Web-scale Deep Web Data Integration," in Machine Learning and Cybernetics, 2009 International Conference,IEEE, Vol. 1, pp. 427-432.

[8] M Sun, H Dou, Q Li, Z Yan, "Quality Estimation of Deep Web Data Sources for Data Fusion," Procedia Engineering 29 (2012), pp. 2347-2354.

[9] C. Hicks, M. Scheffer, A.H. Ngu, and Q.Z. Sheng, "Discovery and cataloging of deep web sources," in Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference ,pp. 224-230.

[10] T Liu, F Wang, J Zhu, G Agrawal, "Differential Analysis on Deep Web Data Sources," In Data Mining Workshops (ICDMW), 2010 IEEE International Conference, pp. 33-40.

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

[12] J Madhavan, D Ko, Ł Kot,V. Ganapathy, "Google's Deep-Web Crawl," Proceedings of the VLDB Endowment,1(2),2008,pp. 1241-1252.

[13] W. Ma, X. Chen, and W. Shang. "Advanced deep web crawler based on Dom," in Fifth International Joint Conference on Computational Sciences and Optimization (CSO), 2012, IEEE, pp. 605-609.

[14] K. F. Bharati, P. Premchand, A. Govardhan, K. Anuradha, and N. Sandhya, "A Framework for Deep Web Crawler Using Genetic Algorithm," International Journal of Electronics and Computer Science Engineering, IJECSE, 2(2), pp.602-609.

[15] H Yu, JY Guo, ZT Yu, YT Xian, "A Novel Method for Extracting Entity Data from Deep Web Precisely," in Control and Decision Conference (2014 CCDC), The 26th Chinese, IEEE, pp. 5049-5053.

[16] J. Song, D.H. Choi, Y.J. Lee, "OGSA-DWC: A Middleware for Deep Web Crawling Using the Grid," in IEEE Fourth International Conference on eScience, 2008, pp. 370-371.

[17] Y. He, D. Xin, V. Ganti, S. Rajaraman, N. Shah, "Crawling Deep Web Entity Pages," in Proceedings of the sixth ACM international conference on Web search and data mining, pp. 355-364.

[18] D. Bergmark, C. Lagoze and A. Sbityakov, "Focused Crawls, Tunneling, and Digital Libraries," in Proceedings of the 6th European Conference on Digital Libraries, Rome, Italy, 2002.

[19] M. Ehrig, A. Maedche, "Ontology-Focused Crawling of Web Documents". Proc. of the Symposium on Applied Computing (SAC 2003). March 9-12, 2003.

[20] G. Pant and P. Srinivasan, "Learning to Crawl: Comparing Classification Schemes". ACM Transactions on Information Systems (TOIS). 23(4), 2005, pp.430-462.

[21] Li, Jun, K. Furuse, and K. Yamaguchi, "Focused Crawling by Exploiting Anchor Text Using Decision Tree". Proceedings of the 14th International World Wide Web Conference. 2005, pp. 1190-1191.

[22] M. Diligenti, F. Coetzee, S. Lawrence, C. Giles and M. Gori, "Focused Crawling Using Context Graphs.". Proc. 26th International Conference on Very Large Databases (VLDB 2000). 2000, pp. 527–534.

[23] H. Liu, J. Janssen, and E. Milios, "Using HMM to Learn User Browsing Patterns for Focused Web Crawling". Data & Knowledge Engineering. 59(2), 2006, pp.270–29.

[24] J. Li, D. Shen, and Y. Kou, "AIE: An Automatic Image Extractor for Deep Web and Surface Web," in Web Information Systems and Applications Conference (WISA), 7th, IEEE, 2010,pp. 137-141.

[25] K. Khurana, and M.B. Chandak, "Video annotation methodology based on ontology for transportation domain," International Journal of Advanced Research in Computer Science and Software Engineering, 3(6), 2013, pp. 540-548.

[26] K. Khurana, and M.B. Chandak, "Study of Various Video Annotation Techniques," International Journal of Advanced Research in Computer and Communication Engineering, 2(1),pp. 909-914.

[27] H Liang, J Chen, W Zuo, Y Mao, "Generating the Semantic Containers for the Query Interfaces of Deep Web," In Management and Service Science, 2009. MASS'09. International Conference, IEEE, pp. 1-4.

[28] J. Madhavan, S. Jeffery, S. Cohen, X. Dong, D. Ko, et.al., "Web-scale data integration: You can only afford to pay as you go," CIDR, January, 2007.

[29] Y. Saissi, A. Zellou, A. Idri, "Extraction of relational schema from deep web sources: a form driven approach," in Complex Systems (WCCS), 2014 Second World Conference, pp. 178-182.

[30] Y. Wang, W. Zuo, T. Peng, F. He, "Domain-Specific Deep Web Sources Discovery," in Natural Computation, 2008, Fourth International Conference, ICNC'08, IEEE, Vol. 5, pp. 202-206.

[31] A. Doan, A. Halevy, Z. Ives, Principles of Data Integration, Elsevier, 2012.

[32] B. He, K. Chen-Chang, and J. Han, "Discovering complex matchings across web query interfaces: a correlation mining approach," Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, pp. 148–157.

[33] B. He, K. Chen-Chang, "Statistical schema matching across web query interfaces," in Proceedings of the 2003 ACM SIGMOD international conference on Management of data, 2003, pp. 217–228.

[34] E. Rahm and P. A.Bernstein, "A survey of approaches to automatic schema matching," The International Journal on Very Large Data Bases, vol. 10,2001, pp. 334–350.

[35] W. Su, J. Wang, F. Lochovsky, "Holistic Schema Matching for Web Query Interface", Advances in Database Technology-EDBT 2006. Springer Berlin Heidelberg, 2006. pp. 77-94.

[36] T. Liu, F.Wang, G. Agrawal. "Stratified Sampling for Data Mining on the Deep Web." Frontiers of Computer Science 6.2, 2012, pp. 179-196.

[37] U. Noor, et.al., "Latent Dirichlet Allocation Based Semantic Clustering of Heterogeneous Deep Web Sources," in Intelligent Networking and Collaborative Systems (INCoS), 2013 5th International Conference, IEEE, pp. 132-138.

[38] Y.J. An, S. Chun, K. Huang, J. Geller, "Assessment for Ontology-supported Deep Web Search," In E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, 2008 10th IEEE Conference, pp. 382-388.

[39] A .K. Sharma, "Accessing the Deep Web Using Ontology," in Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference, pp. 565-568.

[40] G Liu, K Liu, Y Dang, "Research on discovering Deep web entries Based ontopic crawling and ontology," in Electrical and Control Engineering (ICECE), 2011 International Conference, IEEE, pp. 2488-2490.

[41] W. Liu, X. Meng, "A Holistic Solution for Duplicate Entity Identification in Deep Web Data Integration," In Semantics Knowledge and Grid (SKG), 2010 Sixth International Conference, IEEE, pp. 267-274.

[42] S.J. Pusdekar, S.P. Chhaware, "Using Visual Clues Concept for Extracting Main Data from Deep Web Pages," In Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conferenc, IEEE, pp. 190-193.

[43] W Liu, X Meng, W Meng, "ViDE: A Vision-Based Approach for Deep Web Data Extraction," Knowledge and Data Engineering, IEEE Transactions on, 22(3), pp. 447-460.

[44] X. Du, Y. Zheng, Z. Yan, "Automate Discovery of Deep Web Interfaces," in Information Science and Engineering (ICISE), 2010 2nd International Conference, IEEE, pp. 3572-3575.

[45] D. Buttler, L. Liu, and C. Pu, "A Fully Automated Object Extraction

System for the World Wide Web," Proc. Int'l Conf. Distributed Computing Systems (ICDCS), 2001, pp. 361-370.

[46] V. Crescenzi, G. Mecca, and P. Merialdo, "RoadRunner: Towards Automatic Data Extraction from Large Web Sites," Proc. Int'l Conf. Very Large Data Bases (VLDB),2001, pp. 109-118.

[47] M. Alpuente, D. Romero, "A Visual Technique for Web Pages Comparison," Electronic Notes in Theoretical Computer Science, 235, 2009, pp. 3-18.

[48] M. Najork & J. L. Wiener, "Breadth-first crawling yields high-quality pages," In Proceedings of the 10th international conference on World Wide Web, ACM, April 2001, pp. 114-118.