DWDE-IR: An Efficient Deep Web Data Extraction for Information Retrieval on Web Mining

Aysha Banu[1], M. Chitra[2]
[1]Research Scholar, Anna University Chennai, ayshahusain11@gmail.com
[2]Professor, Department of Information Technology, Sona College of Technology, Salem,

Abstract— **Deep Web is a widely unexplored data source, is becoming an important research topic. Retrieving structured data from deep web pages is the challenging problem due to their complex structure. In this paper, Information extracts on the Deep Web pages based on the Deep Web Data Extraction technique (DWDR-IR). Search engines usually return a large number of pages in response to the user queries. To help the users to navigate in the result list, ranking methods are activated on the search results. In this paper, a page ranking mechanism called Coherence Ratio based Page (CRP) ranking algorithm is used. To retrieve the information accurately, an approach called WordNet is used. WordNet checks the similarity of data records and find the correct data region with higher precision using the semantic properties of data records. This concept is very important to display the valuable results occur on the top of the result list on the basis of browsing behavior of the user, it reduces the search space and provides high accuracy. This approach handles the visual features on the deep web data extraction, including data item extraction, data record extraction and visual wrapper generation. The proposed work removes all noise such as header, footer, irrelevant advertisement and irrelevant content using NoiSe Filter (NSFilter) algorithm. The proposed method retrieves perfect extraction of relevant results from the deep web pages. DWDE-IR results higher precision, recall and filter accuracy than the existing method ViDE.**

Index Terms— **Data item extraction, Data record extraction, Deep web data extraction, Ranking algorithm, Visual wrapper generation, WordNet.**

I. Introduction

Deep web is the web that is dynamically generated from the data sources such as file systems or databases. On surface web, data are available through URLs; whereas in deep web data are guarded by a search interface. Crawling deep web is the process of collecting hidden data by issuing queries through various search interfaces. It includes web services, HTML forms and programmable web APIs crawls deep web data sources is essential for several reasons, such as indexing deep web data sources or backing up data.

Most of the information available on the internet cannot be directly accessed via the static link; Users must type some keywords before getting the information hidden in the web database. Deep web contains 400-500 times more information and 15% larger visit capacity than that of surface web. In addition quality of data is also relatively higher. All the web databases make up the deep web; it may be hidden web or invisible web). Generally the query which is entered by the user is enclosed in web pages in the form of data records. These web pages are difficult to index by the traditional crawler based search engines like Google and Yahoo. Data records on a query result page display uniformity in their content, appearance and structure. They show structural and visual similarities. The displayed data in the primary database, and the data items for each record are retrieved from the database in response to the user's query, the same template is used to present the record.

The lexical database for the English is WordNet. It is used to check and find the meaning of words in their contents using the semantic relations of the words. Usually WordNet is used for information retrieval and this approach is also applicable to data extraction on deep web pages. Most of the existing work deals with the extraction of data records are based on the theme of identifying repeated patterns. They identify repeated patterns in the HTML source code of multiple training pages from a data source in order to infer a common structure or template and use it to extract structured data from new pages from the same source. As most of the web pages are written in HTML, it is Web page programming language dependent i.e. HTML dependent. Moreover HTML is no longer the exclusive web page programming language; the other languages have been introduced such as XHTML and XML. Simultaneously to attract human users' consumption of knowledge retrieved from search engines; best template designers of deep web pages always arrange the data records and the data items with visual uniformity to meet the reading habits of human beings. The Document Object Model (DOM) is a cross-platform and language independent convention for representing and interacting with objects in HTML, XHTML and XML documents.

In this paper, an efficient Deep Web Data Extraction for Information Retrieval technique (DWDE-IR) is proposed. Here Visual Block Tree is used to extract the semantic structure of a web page based on the visual presentation. Visual features are used to identify the special information present on the deep web pages. Then data record

doi:10.4304/jetwi.6.1.133-141

extraction phase includes the Filtering, Clustering and Regrouping process. Then Data Item Extraction phase consists of Record Segmentation and Item Alignment steps. Finally Visual Wrapper Generation includes Data Record wrapper and Data Item Wrapper process. The proposed work removes all the noise blocks present on the entire web page and also it reduces the time with high accuracy.

The remaining part of the paper is organized as follows: Section II involves the works related to the Deep Web Data Extraction. Section III involves the existing method ViDE description and problems of ViDE. Section IV involves the description of the proposed method DWDE-IR. Section V includes the performance analysis and explanation of results. The paper is concluded in Section VI.

II. Related Work

Wei et al proposed a vision-based approach that was Web-page-programming-language-independent. This method utilizes the visual features on the deep Web pages to implement deep Web data extraction including data record extraction and data item extraction. The evaluation measure was used to capture the amount of human effort needed to produce the extraction [1]. *Hong* proposed a method WordNet to check the similarity of data records and detect the correct data region with higher precision using the semantic properties of these data records. The proposed method extracts three types of data records namely single-section data records, multiple-section data records and loosely structured data records [2]. *Lin et al* proposed an algorithm for discerning the common path based on hierarchical DOM. Based on the common path and predefined regular expression the target data of the Deep Web was extracted [3]. *Han et al* proposed a Deep Web data extraction and service system based on the principle of cloud technology. A multi-node parallel computing system structure was adopted and a task scheduling algorithm was designed in the data extraction process. In this paper, the lask load was balanced among the nodes to accomplish the data extraction rapidly [4].

Srikantaiah et al proposed a similarity based web data extraction and integration system (WDES and WDICS) to extract search result pages from the web and integrate its contents to enable the user to perform intended analysis. The system provides for local replication of search result pages. This system results better precision and recall than Data Extraction based Partial Tree Alignment (DEPTA) [5]. *Li et al* proposed a Web data scheme and a domain data model. It also puts forward the web table positioning and web table records extracting based on web data schema and an integration algorithm based on the main data model [6]. *Huang et al* proposed an approach to minimize the communication cost to select the appropriate query. A set covering model was used to indicate the web database. An incremental harvest model was learnt by the machine learning method to select the appropriate query automatically [7]. *Hong et al* proposed an approach for information extraction based on fast heuristics techniques. Filtering rules was used to detect and filter out irrelevant data records. A tree matchnig algorithm was also used to increase the speed of data extraction. A data alignment algorithm was proposed to align iterative and disjunctive data items [8].

Kayed et al proposed an approach to measure the relevancy of retrieved web sites to user query concepts and rank them accordingly. a relevancy measure called ontology concepts was proposed. It hepls to re-rank the retrieved documents according to their relevancy to the search query [9]. WSD have been implemented in Lucene using query expansion with thesaurus and relevance feedback. The extended Lesk algorithm was re-implemented to disambiguate the query using WordNet. Expansion terms were limited up to 20 words chosen from expansion term candidates from disambiguated query's senses information, co-occurrence terms and most frequent terms using Kullback-Leibler Distance. The process gets iterated to find the best number of expansion iteration. This method provides better understanding of WSD in information retrieval system performance.

Du and Hai proposed an extension similarity and an intension similarity that analyzed a user's browsing patterns and their hyperlinks. Also the information content similarity between two nouns were compared automatically by examining their ISA and Part-Of hierarchy and using a user's web log. A method for computing the semantic similarity between two concepts in two different lattices and finding the semantic ranking of web pages is proposed. This method proved that the semantic ranking of web pages is useful and efficient for making a web crawlers choice of a web page [10]. *Palekar, et al* utilized the visual features of the deep Web pages to implement deep Web data extraction including data record extraction and data item extraction [11]. *Das and Kumar* designed a technique called Hidden Web Query Technique (HEET) for modelling and query the hidden web. Also a new approach to modelling of consecutive forms and concept of comprehensive form have been presented [12]. *Kayed and Chia Hui* proposed an unsupervised page-level data extraction approach to deduce the schema and templates for each individual deep websites, which contains either singleton or multiple data records in one web page. FiVaTech handles tree matching, mining techniques and tree alignment to achieve the challenging task. [13].

Sreekrishna et al proposed a semantic ontology based deep web data classification method (SODWEB). This method is used to classify the data in the deep web automatically. The URL is choosen for the process of analyzing the semantic association among the concepts. Then the URL of deep web search source was mapped to the category hierarchy obtained [14]. *Patel et al* proposed an effective Deep web data integration approach based on Schema and Attributes Extraction of Query Inetrfaces. This approach avoided the incorrect subsets while grouping attributes and is highly effective on schema extraction of source query interfaces on the invisible web [15]. *Anderson and Hong* proposed a novel approach for extracting the data records from deep web pages [16]. Based on structural regularity, visual and content

similarity between data records displayed on the query result page. This approach used to identify each data record individually while ignoring noise items such as navigation bars and advertisements. This approach resulted higher accuracy.

Miao et al proposed a method for record extraction that captures a list of objects in a more robust way based on a holistic analysis of a web page. It focused on the repetition of tag path appears on the DOM tree of the web document. This paper introduced a similarity measure that captures the visual signal. Tag paths are clustered based on the similarity measure and sets of tag path that form the structure od data records and extracted [17]. *Mukherjee, et al* classified users based on their internet usage patterns and for each class, maintain a cache of web documents. Searching the contents is based on term set analysis and direction cosine distance approach. Finally a more actualize and relevant result set were generated for the query [18]. *Bronzi, et al* proposed an innovative approach that aims at pushing further the level of automation of existing wrapper generation systems by leveraging the redundancy of data on the web. The result of this approach shown a relevant improvement in the precision of the extracted data without a significant loss in the result [19]. *Li and Xie* proposed a keyword-based user interface system EasyUI for achieving web-scale data integration and easy to use for ordinary users. Also the following challenging areas were addressed: indexing schemata terms, data values and domain features of the Deep Web, processing user input, mapping the user query to the domain that the user query most likely corresponds to, and translating the user query into candidate structured queries for the domain. [20].

III. Existing Method- ViDE

A. Description of ViDE

ViDE is the Vision Based Approach for Deep Web Data Extraction, that is Web-page-programming-language-independent methodology [1]. This approach uses the Visual Block Tree to extract the semantic structure of a web page based on the visual presentation. Visual features are used to identify the special information present on the deep web pages. It includes the Data Record Extraction process such as Filtering, Clustering and Regrouping. Filtering removes the top and bottom records of the web page. Clustering the blocks is based on block similarity. The regrouping process uses the regrouping algorithm to regain the blocks. Data Item Extraction phase includes Record Segmentation and Item Alignment process. Record segmentation uses the heuristic rules to segment the records. Data Item alignment focuses on the problem of how to align the data items of the same semantic together and also keep the order of the data items in each data record. Finally, the Visual Wrapper Generation process includes data record wrapper and the data item wrapper. The data record wrapper locates the data region in the Visual Block tree and then extracts the data records from the child blocks of the data region. The data item wrapper takes the attributes $\{a_1, a_2,\ldots,a_n\}$ which are obtained from the sample page and a sequence of data items $\{item_1, item_2, \ldots, item_m\}$ obtained from a new data record, the wrapper process the data items in order to decide which attribute the current data item can be matched. This process is much faster than the wrapper generation process.

B. Problems in ViDE

Extracting structured data from deep web pages is a challenging problem due to their complex structure. Until now, there are a large number of techniques have been proposed to overcome this problem. But all of them have some limitations because they are Web-page-programming-language dependent. ViDE system can only process deep web pages containing only one data region not in multidata-region. It removes the noise blocks on the top and bottom of the web pages and does not guarantee the removal of all the noise blocks. Record segmentation process is based on Visual Block tree. ViDE system takes a long time and low efficiency in extraction.

IV. DWDE-IR: Deep Web Data Extraction for Information Retrieval

A. Basic Concepts about Web Page and Layout

The web pages consist of different kind of information in the form of texts, images, flash, video etc. The Web Page Layout is represented as a coordinate system. The origin locates at the top left corner of the web page. The x-axis is represented as left - right horizontal position and the y-axis are represented as top - bottom vertical position. Suppose each text or image is incorporated in a minimum bounding rectangle with their sides are parallel to the axes, then a text or image has an exact coordinate *(x , y)* on the web page. Here *x* refers horizontal distance from the origin to the left side of its corresponding rectangle and *y* refers vertical distance from the origin to the upper side of its corresponding rectangle. The x and y coordinates and sizes of texts or images on the Web page from the Web Page Layout.

B. Visual Block Tree and Visual Features

Vision-based Page Segmentation Algorithm (VIPS) [1] aims to extract the semantic structure of a web page based on its visual presentation. Such semantic structure is usually called as a tree structure. Each node in the tree correlates to a block. Each node will be assigned a value called the Degree of Coherence. It is used to indicate how coherent of the content in the block based on visual perception. The algorithm extracts all the suitable blocks from the HTML DOM tree and then it finds the separators between these blocks. Separators denote the horizontal or vertical lines in a web page that visually cross with no blocks. Based on these separators the semantic tree of the web page is constructed so that the web page can be represented as a set of blocks. Noisy information such as navigation, advertisement and other decoration can be easily removed because they are often placed in certain positions on a page. Contents with different topics are categorized as separate blocks.

1) Visual Features

Usually web pages are used to broadcast information to users like other kinds of media such as newspaper and TV. Visual features are very important to identifying special information on web pages. Deep web pages contain data records which are retrieved from the web databases.

The visual features are recognized based on its four features:

1. Position features (PF)
2. Layout features (LF)
3. Appearance features (AF)
4. Content features (CF)

Position features demonstrate the location of the data region on deep web page**.** Data regions are always placed centered horizontally. Layout features show that how the data records in the data region is commonly arranged. Appearance feature catches the visual features within data records. Content features trace the regularity of the contents in data records.

Fig.1. (a) Example for Deep Web Page

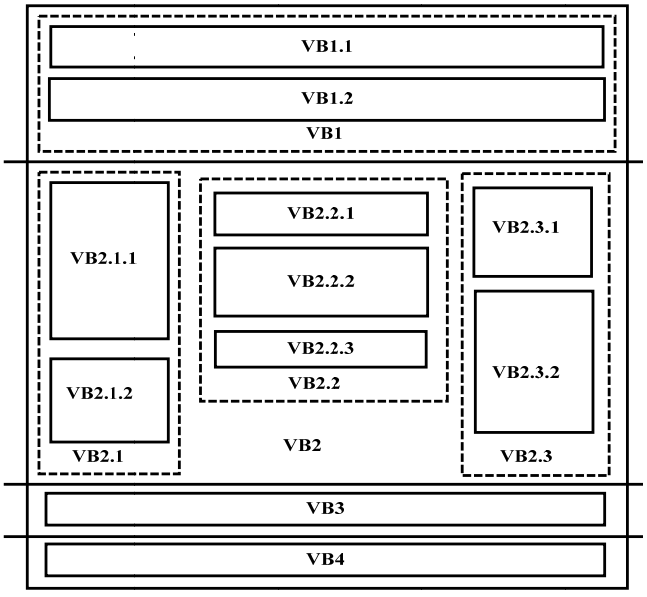

Fig.1. (b) Visual Presentation Structure

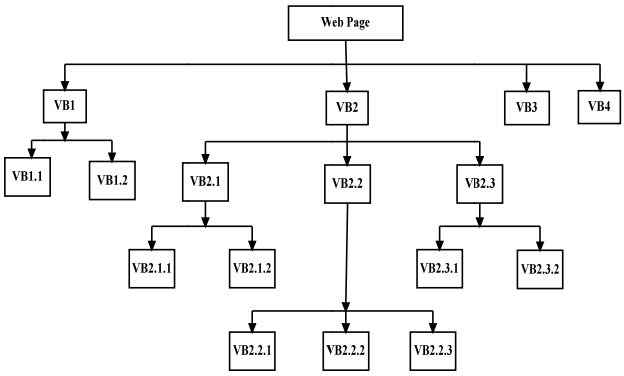

Fig.1. (c) Visual Block Tree

C. Data Record Extraction

Data record extraction aims to locate the boundary of data records and extract them from the deep web pages. So the following three techniques are incorporated:

1) Filtering – NSFilter Algorithm

Using NSFilter algorithm, all the noise blocks located on the web page are removed. Noise blocks are present on anywhere on the web page body. Existing system removes only the noise blocks present on the top and bottom of the webpage. To overcome this limitation proposed system introduced the NSFilter algorithm.

---

for each LF_i in *LF*
  If LF_i = *center*
    Add into *FB;* *//removes header, footer, right left noises*
end for

for each *FB* in new
  if *FB = Error*
  FB = FB - FB$_i$ *// removes advertisements and unwanted contents*
end for

Here *FB* is the filtered block list. Error contains the unwanted information blocks which are to be removed. Data regions are always placed centred horizontally. NSFilter removes all the noises such as header, footer, advertisements and other unwanted information's present on the webpage.

2) Clustering- Aggregate Clustering Approach

The filtered blocks are clustered using an Aggregate cluster approach based on the block and value similarity. In the existing system, similarity can be calculated using how many blocks present in the same type of the web page. It depends based upon the block count, not a value presented on the web page. To overcome this problem proposed system clusters the block based on both block similarity and value similarity. The equation for computing the appearance similarity between two blocks b_1 and b_2 is given below:

$$Sim(b_{s1}, b_{s2}) = w_i * simImage(b_{s1}, b_{s2}) + w_{pt} * simPText(b_{s1}, b_{s2}) + w_{lt} * simLText(b_{s1}, b_{s2}) \quad (1)$$

where *simImage*(b_{s1}, b_{s2}), *simPText* (b_{s1}, b_{s2}), and *simLText* (b_{s1}, b_{s2}) are the similarities established on image size, plain text font and link text font. w_i , w_{pt} , and w_{lt} are the weights of these similarities.

The equation for computing the appearance similarity between two values v_1 and v_2 of two blocks (b_{s1}, b_{s2}) is given below:

$$Sim\ (v_1, v_2) = w_t * semsim\ (v_1, v_2) \quad (2)$$

where *simImage* (v_1, v_2), *simPText* (v_1, v_2) and *simLText* (v_1, v_2) are the similarities based on Image tag, Plain text content and Link text content. w_i , w_{pt} , and w_{lt} are the weights of these similarities.

$$SimImage(b_{s1}, b_{s2}) = \frac{\min\{Ta_i(b_{s1}), Ta_i(b_{s2})\}}{\max\{Ta_i(b_{s1}), Ta_i(b_{s2})\}} \quad (3)$$

$$w_i = \frac{Ta_i(b_{s1}) + Ta_i(b_{s2})}{Ta_b(b_{s1}) + Ta_b(bs_2)} \quad (4)$$

$$SimPText(b_{s1}, b_{s2}) = \frac{\min\{Tf_{pt}(b_{s1}), Tf_{pt}(bs_2)\}}{\max\{Tf_{pt}(b_{s1}), Tf_{pt}(b_{s2})\}} \quad (5)$$

$$w_{pt} = \frac{Ta_{pt}(b_{s1}) + Ta_{pt}(b_{s2})}{Ta_b(b_{s1}) + Ta_b(b_{s2})} \quad (6)$$

$$SimLText(b_{s1}, b_{s2}) = \frac{\min\{Tf_{lt}(b_{s1}), Tf_{lt}(b_{s2})\}}{\max\{Tf_{lt}(b_{s1}), Tf_{lt}(b_{s2})\}} \quad (7)$$

$$w_{lt} = \frac{Ta_{lt}(b_{s1}) + Ta_{lt}(b_{s2})}{Ta_b(b_{s1}) + Ta_b(b_{s2})} \quad (8)$$

$$w_t = \frac{Tot\ text\ v_1 + Tot\ text\ v_2}{Tot\ b_{s1} + Tot\ b_{s2}} \quad (9)$$

Here $Ta_i(b)$ represents the total area of images available in block b. $Ta_b(b)$ denotes the total area of block b. $Tf_{pt}(b)$ denotes the total number of fonts of the plain text available in block b. $Ta_{pt}(b)$ is the total area of plain text in block b. $Tf_{lt}(b)$ denotes the total number of fonts of the link text available in block b. $Ta_{lt}(b)$ is the total area of link text in block b.

Clustering the blocks based on blocks and value similarities:

$$Sim(b_1, b_2) = sim\ (bs_1, bs_2) + sim\ (v_1, v_2) \quad (10)$$

Where $Sim(b_1, b_2)$ denotes the similarity measure between the blocks and value. The nearest value among the blocks are clustered based on the eqn(10)

3) Regrouping- MERGE Method

The clustered blocks are needed to regroup, such that the blocks belonging to the same data record form a group. Existing system uses the manual process of regrouping according to their position. The result of the existing system is not so accurate for regrouping. So in this work MERGE method is used to merge the blocks based on their tags, value and data block.

Algorithm: Regrouping- MERGE METHOD

Input: $k_1, k_2 \ldots k_m$: group of clusters generated by blocks clustering from a given deep web page *A*.
Output: $d_1, d_2 \ldots d_m$: each of them corresponds to data record on *A*.
Begin
//Step1: Sort the given blocks in k_i according to their position
(based on top to bottom; then left to right)
1: **for** each cluster k_i do
2: **for** any two blocks $b_{i,j}$ and $b_{i,k}$ in k_i
//$1 \le j \le k \le |k_i|$
3: **if** $b_{i,j}$ and $b_{i,k}$ are in different lines on *A*, and $b_{i,k}$ is above $b_{i,j}$
4: $b_{i,j} \leftrightarrow b_{i,k}$; //exchange their orders in k_i
5: **else if** $b_{i,j}$ and $b_{i,k}$ are in the same line on *A*, and $b_{i,k}$ is in front of $b_{i,j}$
6: $b_{i,j} \leftrightarrow b_{i,k}$;
7: **else**
8: if sim ($b_{i,j}$, $b_{i,k}$) > threshold
9: merge ($b_{i,j}$, $b_{i,k}$);
10: **end** until no exchange occurs;
11: form the minimum-bounding rectangle Rct_i for k_i;

//Step2: Initialize g groups and g is the number of data records on *A*
12: $k_{max=} \{k_i \mid |k_i| = \max\{|k_1|, |k_2| \ldots |k_m|\}\}$; //$g = |k_{max}|$
13: for each block $b_{max,i}$ in k_{max}
14: initialize group V_i;
15: put $b_{max,i}$ into V_i;

//Step 3: Insert the blocks into right groups and each group corresponds to a data record
16: **for** each cluster k_i
17: **if** Rct_i overlaps with Rct_{max} on *A*
18: **if** Rct_i is ahead of (behind) Rct_{max}
19: **for** each block $b_{i,j}$ in k_i
20: find the nearest block $b_{max,k}$ in k_{max} that is behind $b_{i,j}$ on the web page;
21: place $b_{i,j}$ into group V_k;
End

D. Data Item Extraction

In this paper, the problem of segmenting the data records into a sequence of data items and aligning the data

items of the same semantics together are focused. To address this issue, two techniques are used.

1) Data Record Segmentation- MSEGMENT Algorithm

The regrouped records are taken as inputs to segment the data records. The records within a data area are identified as sub-trees. The sub-trees rooted through children of the data area root. The segmentation process depends on record separators i.e sub-trees interleaved with data records. The root node of a record separator is a child of the data area root and it does not contain any text or URL. Since each of the record contains only one instance of matching node (MN), the segmentation process heuristically identifies the areas belonging to a single record.

Steps for MSegment Algorithm

Input: Regrouped records
Output: Segmented records

```
1   Consider CR
2   Expand CR
3   Consider S of CR
4   If S ϵ RS
5       Consider CR as PR
6   Else
7       Compute dist between two CR
8       Consider all 2 x (dist-1) to left and right ST
9       Repeat step 8 to all CR and compute similarity
    between the identified expansion
10      If several expansion have same similarity
11          Choose the highest similarity among R
```

Here CR denotes the candidate record, RS denotes record segmentation, S is the siblings of the CR, PR denotes proper record and R denotes records.

In the first step of the algorithm, each sub tree in the DOM containing a single MN and rooted at a direct child of the data area root is considered as a CR. A successive step tries to expand the CR to adjacent sub trees in the DOM. Therefore consider the S of CR. If they are RS, consider each CR as a PR; otherwise compute the distance between CR as the number of S between their root nodes. Then consider all the 2 x (dist-1) expansions of a record to left and right adjacent sub trees. Apply the same expansion to all the CR and compute the similarity. Choose the highest structural similarity among R.

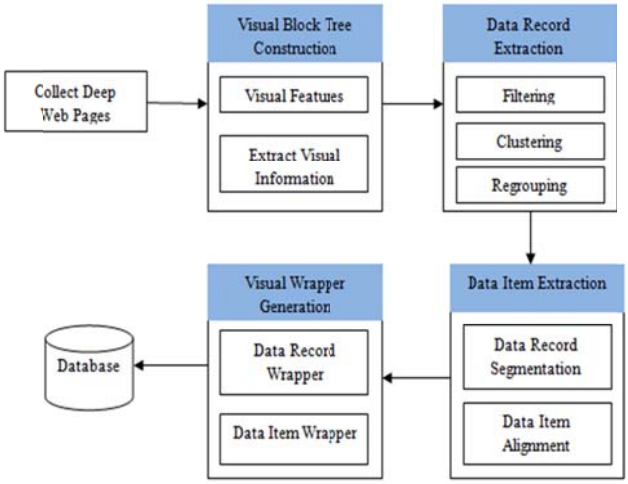

Fig.2. Extracting information from the web database

2) Data Item Alignment- Nested Structure

The data records are represented as $\{r_1, r_2 \ldots r_n\}$ and each data record r_i is denoted as a sequence of data items $\{item_1, item_2 \ldots item_m\}$. The entire data item has a unique position in its corresponding sequence according to the semantic order.

Data item alignment focuses on the problem of how to align the data items of the same semantic together and also keep the order of the data items in each data record. Proposed work uses the nested structure for data item alignment.

Given a column c_1 which contains n data values, the intra-column similarity $similarity_{intra}$ be the average data value similarity within each column in c_1

$$Similarity_{intra} = 2\,\frac{\sum_{q=1}^{m-1}\sum_{p=q+1}^{m} L_{pq}}{m(m-1)} \qquad (11)$$

Here L_{pq} is the data value similarity between pth and qth data values of c_1.

For two columns c_1 and c_2 which has f and g data values respectively, the inter-column similarity $similarity_{inter}$ is defined to be the average data value similarity of every pair of data values in c_1 and c_2.

$$Similarity_{inter} = \frac{\sum_{q=1}^{g}\sum_{p=1}^{f} L_{pq}}{fg} \qquad (12)$$

Here L_{pq} is the data value similarity between pth data values of c_1 and qth data values of c_2 using the data value similarity. After $similarity_{intra}$ and $similarity_{inter}$ are calculated for identified columns c_p , if $similarity_{inter}$ / $similarity_{intra} > S_{nest}$ is a threshold that is set to 0.5, c_p is assumed to be a nested column set, which means that the data values in it are generated from a nested structure.

E. Visual Wrapper Generation-MViDE

Wrapper Generation includes data record wrapper and the data item wrapper. They are the programs which execute the data record extraction and data item extraction with a set of parameter extracted from sample pages. Each Web database a normal deep Web page containing the

maximum number of data records are used to generate the wrappers.

1) Data Record Wrapper

For each of the record, visual data record wrapper find the first block of each record and the last block of the last data record (b_{LDR}). To achieve this objective, the visual information of the first block of each data record extracted from the sample page is saved and the distance (t) between two data records are also saved. For the child blocks of the data region in a new page, find the first block of each data record by the visual similarity with the saved visual information. Then b_{LDR} on the new page needs to be placed. The vertical distance between any two neighboring blocks in one data record is always smaller than t and the vertical distance between b_{LDR} and its next block is not smaller than t. Hence the first block is recognized whose distance with its next block is larger than t as b_{LDR}.

2) Data Item Wrapper

The data alignment algorithm groups data items from different data records into columns or attributes such that data items under the same column have the same semantics. Given a sequence of attributes $\{t_1,t_2,..,t_n\}$ obtained from the sample web page and a sequence of data items $\{d_1,d_2,\ldots,d_m\}$ which is obtained from a new data record. The data items are wrapped in order to decide which of the attribute matches the current data item. For d_i and t_j, if they are same as *T,B* and *A* then their match is recognized. The wrapper then checks whether d_{i+1} and t_{j+1} if not, it checks d_i and t_{j+1}. Repeat the process until all the data items are matched to their corresponding attributes.

T is the font used by the data items, *B* denotes Boolean, and *A* is the image, text, number, date etc.

F. Query Processing

The data record list and item are extracted from the database. Then the extracted result of the database is compared with the user query. The co-occurrence ratio is calculated for the user query and the extracted data record. Co-occurrence ratio1 is calculated using query and the key term. Co-occurrence ratio2 is calculated using query and the extracted record list. When the information is queried through a search engine, the result will be presented as a list, and the result which best satisfies the user demand will appear at the top of the list. The web pages are displayed based on the Page Ranking Algorithm.

Co-occurrence Ratio1

$$R1 = \sum_{i=1}^{m} Sim(q, t_i) \quad (13)$$

where *Sim* is the similarity measure between query q and key term t_i.

Co-occurrence Ratio2

$$R2 = \sum_{i=1}^{m} SEMSim(q, t_i) \quad (14)$$

where *SEMsim* is the semantic similarity between query q and t_i is the extracted record list based on WordNet [2].

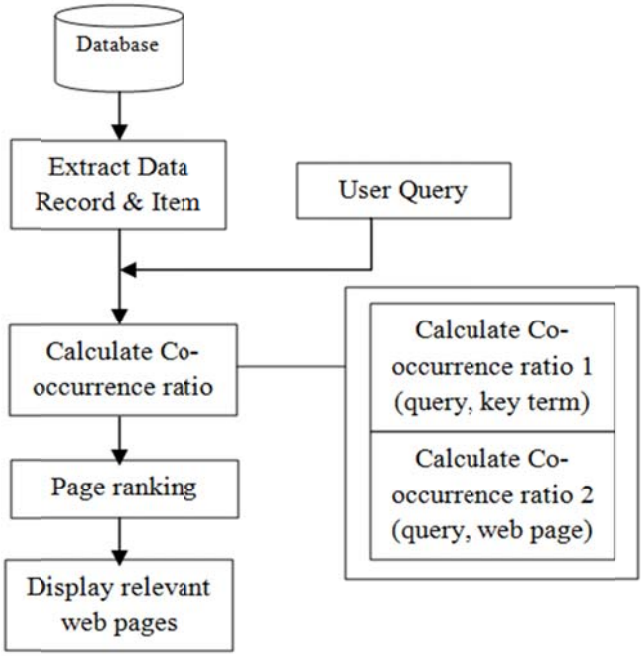

Fig.3. Flow for generating accurate results

Algorithm-CRP Ranking

Input: Query, Extracted record list q;
Output: Score List
Convert q into t1, t2… t_m;
for (j=1; j<=n; j++)
  Calculate R1 using eqn (1)
  Calculate R2 using eqn (2)
C[j] = k_1 * R1 + k_2 * R2;
end for

Select the top k results from the result set, select (R, K);
Here k_1 and k_2 are the weighting coefficient, $k_1 + k_2=1$.

Based on this algorithm the similarity measures are calculated and the web pages are displayed based on the best score among the matched web pages.

V. PERFORMANCE ANALYSIS

In this paper the performance quality is measured using Precision, Recall, Filter Accuracy and search time between the proposed system DWDE-IR and ViDE. The implementation is carried out by taking the two datasets like GDS and SDS.

A. Precision

Precision is the fraction of retrieved instances that are relevant. Precision is calculated correctly based on the extracted data items and records with the total number of extracted data records and items.

$$\textit{Precision for data record extraction} = \frac{Rec_c}{Rec_e} \quad (15)$$

where Rec_c is the total number of correctly extracted data records and Rec_e is the total number of data records extracted.

$$\textit{Precision for data item extraction} = \frac{Item_c}{Item_e} \quad (16)$$

where $Item_c$ is the total number of correctly extracted data items and $Item_e$ is the total number of data items extracted.

Fig.4. and Fig5 are the precision output for the existing technique ViDE and the proposed method DWDE-IR for record and item extraction.

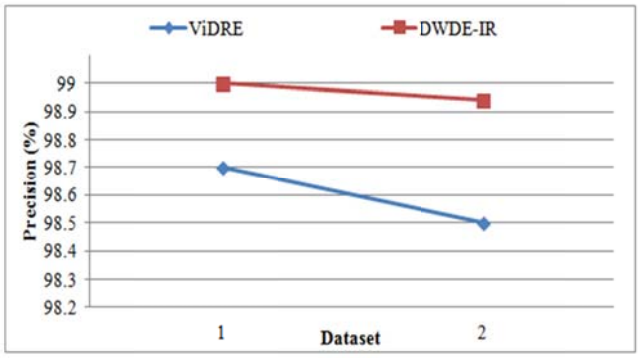

Fig.4. Precision between ViDE and DWDE-IR for record extraction

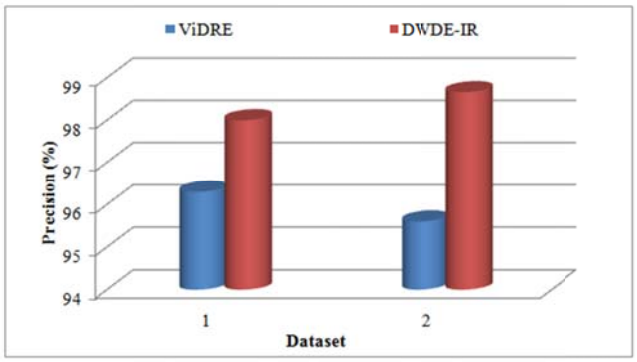

Fig.5. Precision between ViDE and DWDE-IR for item extraction

B. Recall

Recall is the fraction of relevant instances that are retrieved. Recall is calculated correctly based on the extracted data items and records with the total number of data records and items.

$$\textit{Recall for data record extraction} = \frac{Rec_c}{Rec_d} \quad (17)$$

where Rec_c is the total number of correctly extracted data records and Rec_d is the total number of data records.

$$\textit{Recall for data item extraction} = \frac{Item_c}{Item_d} \quad (18)$$

where $Item_c$ is the total number of correctly extracted data items and $Item_d$ is the total number of data items.

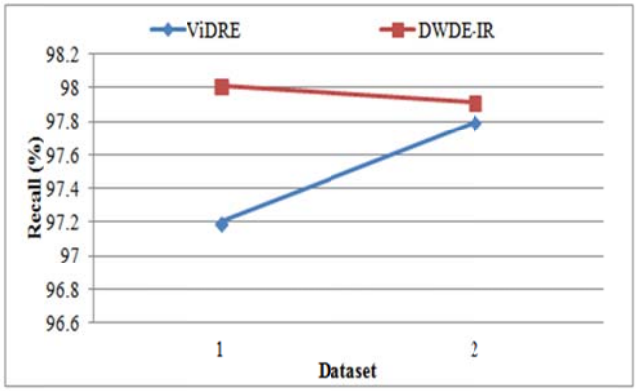

Fig.6. Recall between ViDE and DWDE-IR for record extraction

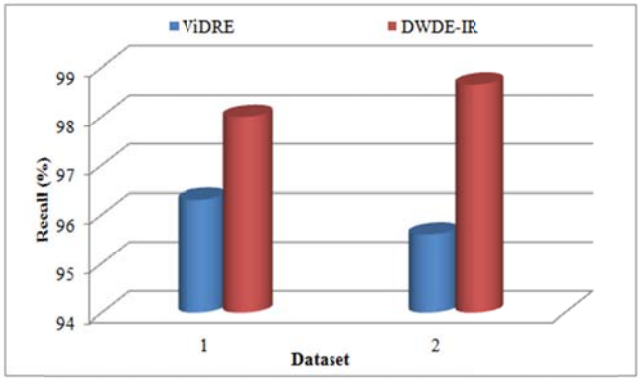

Fig.7. Recall between ViDE and DWDE-IR for item extraction

Fig.6. and Fig.7 shows the comparison of recall calculated for record and item extraction between ViDE and DWDE-IR. The result shows that the proposed system DWDE-IR generates better relevant instances than the existing system ViDE.

C. Filter Accuracy

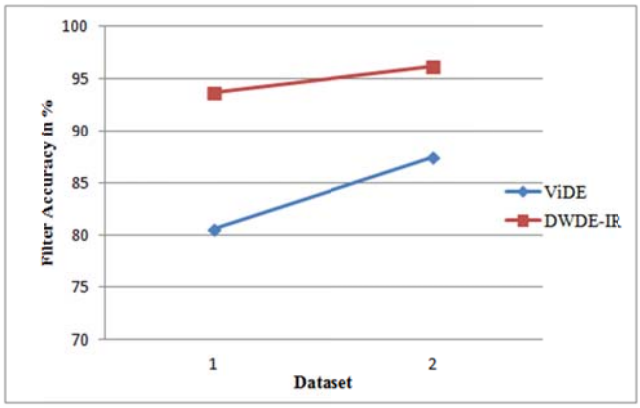

Fig.8. Filter Accuracy between ViDE and DWDE-IR

DWDE-IR generates more accurate result than the existing ViDE approach. DWDE-IR removes the noise present on anywhere on the webpage; whereas ViDE removes only the noise present on the top and bottom of the webpage.

D. Search Time

The searching time for the proposed DWDE-IR consumes less time when compared with the existing ViDE. It is shown in Fig.9.

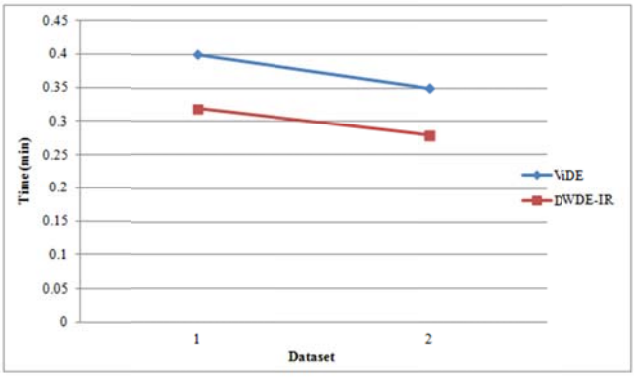

Fig.9. Search time between ViDE and DWDE-IR

VI. Conclusion And Future Work

The problem of Deep Web Data extraction has received a lot of problems in recent years and most of the proposed solutions are based on HTML source code. In this paper, an efficient Deep Web Data Extraction for Information Retrieval technique (DWDE-IR) is proposed. The proposed system is html independent and can even process deep web pages containing multiple data region. DWDE-IR removes all the noise blocks such as header, footer, irrelevant advertisement and irrelevant content from the webpage. It uses the page ranking algorithm to display the best suited result at the top of the result list. The processing time is quite less and also produces high accuracy results. The proposed system results higher precision, recall and filter accuracy than the existing approaches.

In future, the different types of wrapper generation method are applied to test better data record wrapping and data item wrapping. Then best clustering technique is incorporated to attain an accurate information retrieval.

References

[1] L. Wei, M. Xiaofeng, and M. Weiyi, "ViDE: A Vision-Based Approach for Deep Web Data Extraction," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 22, pp. 447-460, 2010.

[2] J. L. Hong, "Data extraction for deep web using wordnet," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 41, pp. 854-868, 2011.

[3] T. Lin, B. H. Qiang, S. Long, and H. Qian, "Deep Web Data Extraction Based on Regular Expression," *Advanced Materials Research,* vol. 718, pp. 2242-2247, 2013.

[4] Z. Y. Han, F. Y. Wang, P. Sun, and Z. Y. Li, "A Deep Web Data Extraction and Application System Based on Cloud Technology," *Advanced Materials Research,* vol. 756, pp. 2583-2587, 2013.

[5] K. Srikantaiah, M. Suraj, K. Venugopal, S. Iyengar, and L. Patnaik, "Similarity Based Web Data Extraction and Integration System for Web Content Mining," in *Advances in Communication, Network, and Computing*, ed: Springer, 2012, pp. 269-274.

[6] G. Li, Z. Y. Han, Z. X. Chen, Z. Y. Li, and P. Sun, "Web Data Extraction and Integration in Domain," *Advanced Materials Research,* vol. 756, pp. 1585-1589, 2013.

[7] Q. Huang, Q. Li, H. Li, and Z. Yan, "An Approach to Incremental Deep Web Crawling Based on Incremental Harvest Model," *Procedia Engineering,* vol. 29, pp. 1081-1087, // 2012.

[8] J. L. Hong, E.-G. Siew, and S. Egerton, "Information extraction for search engines using fast heuristic techniques," *Data & Knowledge Engineering,* vol. 69, pp. 169-196, 2// 2010.

[9] A. Kayed, E. El-Qawasmeh, and Z. Qawaqneh, "Ranking web sites using domain ontology concepts," *Information & Management,* vol. 47, pp. 350-355, 12// 2010.

[10] Y. Du and Y. Hai, "Semantic ranking of web pages based on formal concept analysis," *Journal of Systems and Software,* 2012.

[11] V. Palekar, M. Ali, and R. Meghe, "Deep Web Data Extraction using Web-Programming-Language-Independent Approach," *Journal of Data Mining and Knowledge Discovery,* vol. 3, 2012.

[12] N. N. Das and E. Kumar, "Hidden Web Query Technique for Extracting the Data From Deep Web Data Base," in *Proceedings of the World Congress on Engineering and Computer Science*, 2012.

[13] M. Kayed and C. Chia Hui, "FiVaTech: Page-Level Web Data Extraction from Template Pages," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 22, pp. 249-263, 2010.

[14] M. Sreekrishna, P. Sundaramoorthy, and T. Rajendran, "A Novel Approach for Semantic Based Deep Web Classification," *International Journal of Advanced and Innovative Research (IJAIR),* vol. Vol.2 pp. 394-402, 2013.

[15] G. Patel, A. S. Rajawat, and S. Vyas, "Deep web Data Integration Approach Based on Schema and Attributes Extraction of Query Interfaces," *International Journal of Managment, IT and Engineering,* vol. 2, pp. 272-284, 2012.

[16] N. Anderson and J. Hong, "Visually extracting data records from the deep web," in *Proceedings of the 22nd international conference on World Wide Web companion*, 2013, pp. 1233-1238.

[17] G. Miao, J. Tatemura, W.-P. Hsiung, A. Sawires, and L. E. Moser, "Extracting data records from the web using tag path clustering," in *Proceedings of the 18th international conference on World wide web*, 2009, pp. 981-990.

[18] I. Mukherjee, V. Bhattacharya, S. Banerjee, P. Gupta, and P. Mahanti, "Efficient web information retrieval based on usage mining," in *Recent Advances in Information Technology (RAIT), 2012 1st International Conference on*, 2012, pp. 591-595.

[19] M. Bronzi, V. Crescenzi, P. Merialdo, and P. Papotti, "Wrapper generation for overlapping web sources," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, 2011, pp. 32-35.

[20] Y. Li and C. Xie, "A easy user interface of IR system over large scale deep web," in *Mechatronic Science, Electric Engineering and Computer (MEC), 2011 International Conference on*, 2011, pp. 250-253.