C .Guru Sunanda, Guide: K. Ishthaq Ahamed / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 5, September- October 2012, pp.632-640

Structured Data Extraction from Deepnet

C .Guru Sunanda
(M.Tech) Department of CSE,
G.Pulla Reddy Engg. College,
College, Kurnool. Andhra Pradesh.

Guide: K. Ishthaq Ahamed
Associate Professor
CSE Department, G.Pulla Reddy Engg.
Kurnool. Andhra Pradesh.

Abstract:

Deep Web contents are accessed by queries submitted to Web databases and the returned data records are enwrapped in dynamically generated Web pages (they will be called deep Web pages in this paper). Extracting structured data from deep Web pages is a challenging problem due to the underlying intricate structures of such pages. Until now, a large number of techniques have been proposed to address this problem, but all of them have inherent limitations because they are Web-page-programming-language dependent. As the popular two-dimensional media, the contents on Web pages are always displayed regularly for users to browse.

This motivates us to seek a different way for deep Web data extraction to overcome the limitations of previous works by utilizing some interesting common visual features on the deep Web pages. In this paper, a novel vision-based approach that is Web-page programming language-independent is proposed. This approach primarily utilizes the visual features on the deep Web pages to implement deep Web data extraction, including data record extraction and data item extraction. We also propose a new evaluation measure revision to capture the amount of human effort needed to produce perfect extraction. Our experiments on a large set of Web databases show that the proposed vision-based approach is highly effective for deep Web data extraction.

***Keywords*:** Web mining, Web data extraction, visual features of deep Web pages, wrapper generation, Clustering, Regrouping, Visual matching.

1. INTRODUCTION

Data mining refers to extracting hidden and valuable Knowledge and information from large data bases. It involves method s and algorithms to extract knowledge from different data repositories such as transactional databases, data warehouses text files, and www etc (as sources of data).World Wide Web (WWW) is a vast repository of interlinked hypertext documents known as web pages. A hypertext document consists of both, the contents and the hyperlinks to related documents.

Users access these hypertext documents via software known as web browser. It is used to view the web pages that may contain information in form of text, images, videos and other multimedia. The documents are navigated using hyperlinks, also known as Uniform Resource Locators (URLs).

It is very difficult to search information from such a huge collection of web documents on World Wide Web as the web pages/documents are not organized as books on shelves in a library, nor are web pages completely catalogued at one central location. It is not guaranteed that users will be able to extract information even after knowing where to look for information by knowing its URLs as Web is constantly changing. Therefore, there was a need to develop information extraction tools to search the required information from WWW.

Web Information Extraction: The amount of Web information has been increasing rapidly, especially with the emergence of Web 2.0 environments, where users are encouraged to contribute rich content. Much Web information is presented in the form of a Web record which exists in both detail and list pages

The task of web information extraction (WIE) or information retrieval of records from web pages is usually implemented by programs called wrappers. The process of leaning a wrapper from a group of similar pages is called wrapper induction. Due to its high extraction accuracy, wrapper induction is one of the most popular methods of web information extraction and it is extensively used by many commercial information systems including major search engines. Figure shows the method for extracting information from webpage. Wrapper induction is involving the deduction Rules and is semi automatic. Besides there are automatic Extraction methods are proposed.

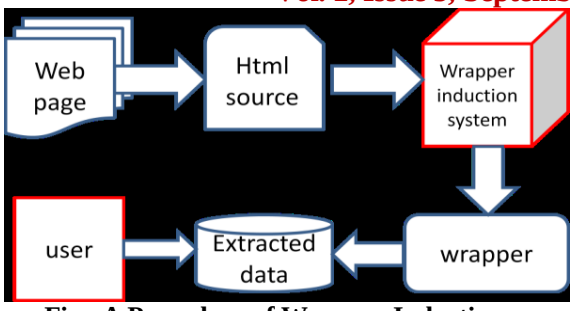


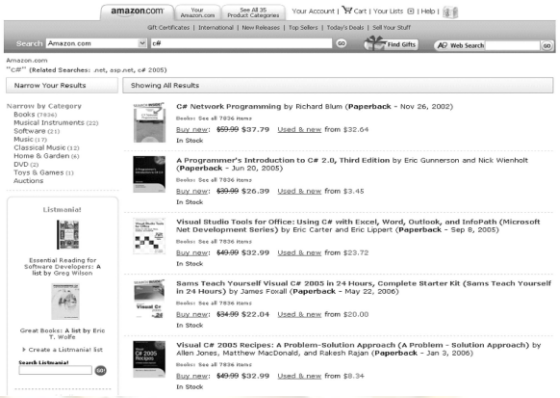
Fig. A Procedure of Wrapper Induction

It is difficult to extract a record from a page, specifically in the case of cross records. For example, Figure 2 is a part of web page taken from www.justdial.com .In this example two records are presented, the names of two TV dealers are presented together in the first part , while their location are shown in the second part ,their phone numbers are shown in third part in each record. Those two records' HTML sources are crossed with each other. Many shopping websites (e.g.,www.amazon.com,www.bestbuy.com, www.diamond.com, www.costco.com, etc.) also list some of their products in a similar way as in Figure. In Most of the web information methods HTML page is converted in to the document object model (DOM) .This is done by parsing the HTML page.

Fig. Web Record Presentations

2. PROPOSED DESIGN

The World Wide Web has more and more online Web databases which can be searched through their Web query interfaces. All the Web databases make up the deep Web (hidden Web or invisible Web). Often the retrieved information (query results) is enwrapped in Web pages in the form of data records. These special Web pages are generated dynamically and are hard to index by traditional crawler based search engines, such as Google and Yahoo. This kind of special Web page is called as deep Web pages. Each data record on the deep Web pages corresponds to an object. In order to ease the consumption by human users, most Web databases display data records and data items regularly on Web browsers. However, to make the data records and data items in them machine processable, which is needed in many applications such as deep Web crawling and metasearching, the structured data need to be extracted from the deep Web pages. The problem of automatically extracting the structured data, including data records and data items, from the deep Web pages can be solved by using Vision Based Approach.



Vision-Based Data Extractor (Vide):

Vision-based Data Extractor (ViDE), to extract structured results from deep Web pages automatically. ViDE is primarily based on the visual features human users can capture on the deep Web pages while also utilizing some simple non-visual information such as data types and frequent symbols to make the solution more robust. ViDE consists of two main components, Vision based Data Record extractor (ViDRE) and Vision-based Data Item extractor (ViDIE). By using visual features for data extraction, ViDE avoids the limitations of those solutions that need to analyze complex Web page source files.

This approach employs a four-step strategy. First, given a sample deep Web page from a Web database, obtain its visual representation and transform it into a Visual Block tree which will be introduced later; second, extract data records from the Visual Block tree; third, partition extracted data records into data items and align the data items of the same semantic together; and fourth, generate visual wrappers (a set of visual extraction rules) for the Web database based on sample deep Web pages such that both data record extraction and data item extraction for new deep Web pages that are from the same Web database can be carried out more efficiently using the visual wrappers.

ViDE is independent of any specific Web page programming language. Although our current implementation uses the VIPS algorithm to obtain a deep Web page's Visual Block tree and VIPS needs to analyze the HTML source code of the page, our solution is independent of any specific method used to obtain the Visual Block tree in the sense that any tool that can segment the Web pages into a tree

structure based on the visual information, not HTML source code, can be used to replace VIPS in the implementation of ViDE.

3. Visual Block Tree and Visual Features:

3.1Visual Information of Web Pages

The information on Web pages consists of both texts and images (static pictures, flash, video, etc.). The visual information of Web pages used in this paper includes mostly information related to Web page layout (location and size) and font.

3.1.1 Web Page Layout

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.

3.1.2 Font

The fonts of the texts on a Web page are also very useful visual information, which are determined by many attributes. Two fonts are considered to be the same only if they have the same value under each attribute.

Font Attributes and Examples

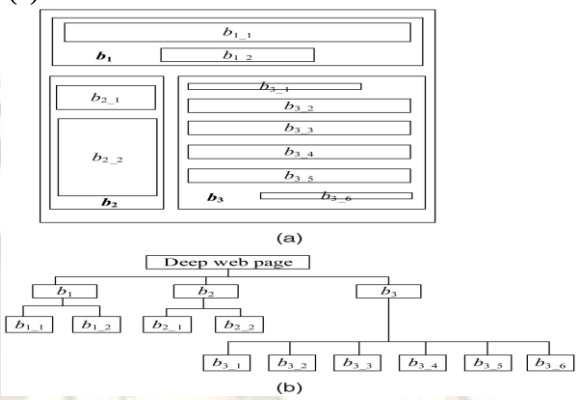
Font factor	Example	Font factor	Example
Size	A (10pt)	underline	A
face	A(Sans Serif)	italic	*A*
color	A (red)	weight	**A**
strikethrough	~~A~~	frame	A

3.2 Deep Web Page Representation

The visual information of Web pages, which has been introduced above, can be obtained through the programming interface provided by Web browsers (i.e., IE). In this paper, we employ the VIPS algorithm to transform a deep Web page into a Visual Block tree and extract the visual information. A Visual Block tree is actually a segmentation of a Web page. The root block represents the whole page, and each block in the tree corresponds to a rectangular region on the Web page. The leaf blocks are the blocks that cannot be segmented further, and they represent the minimum semantic units, such as continuous texts or images.

An actual Visual Block tree of a deep Web page may contain hundreds even thousands of blocks. Visual Block tree has three interesting properties. First, block a contains block b if a is an ancestor of b. Second, a and b do not overlap if they do not satisfy property one. Third, the blocks with the same parent are arranged in the tree according to the order of the corresponding nodes appearing on the page.

(a) The presentation structure and
(b) its Visual Block tree.



Visual Features of Deep Web Pages

Visual features are important for identifying special information on Web pages. Deep Web pages are special Web pages that contain data records retrieved from Web databases, and we hypothesize that there are some distinct visual features for data records and data items.

Position features (PFs). These features indicate the location of the data region on a deep Web page.

PF1: Data regions are always centered horizontally.

PF2: The size of the data region is usually large relative to the area size of the whole page.

Since the data records are the contents in focus on deep Web pages, Web page designers always have the region containing the data records centrally and conspicuously placed on pages to capture the user's attention. The two interesting facts. First, data regions are always located in the middle section horizontally on deep Web pages. Second, the size of a data region is usually large when there are enough data records in the data region. The actual size of a data region may change greatly because it is not only influenced by the number of data records retrieved, but also by what information is included in each data record. Therefore, this approach uses the ratio of the size of the data region to the size of whole deep Web page instead of the actual size.

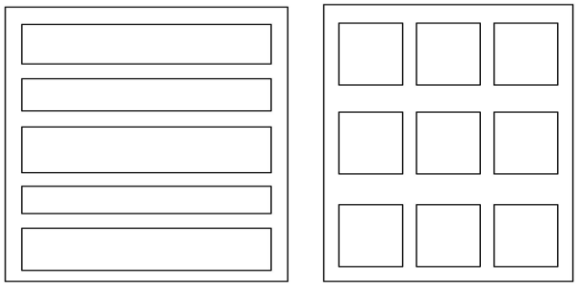
Layout features (LFs). These features indicate how the data records in the data region are typically arranged.

LF1: The data records are usually aligned flush left in the data region.

LF2: All data records are adjoining.

LF3: Adjoining data records do not overlap, and the space between any two adjoining records is the same.

Layout models of data records on deep Web pages.



Data records are usually presented in one of the two layout models. In Model 1, the data records are arranged in a single column evenly, though they may be different in width and height. LF1 implies that the data records have the same distance to the left boundary of the data region. In Model 2, data records are arranged in multiple columns, and the data records in the same column have the same distance to the left boundary of the data region.

Appearance features (AFs). These features capture the visual features within data records.

AF1: Data records are very similar in their appearances, and the similarity includes the sizes of the images they contain and the fonts they use.

AF2: The data items of the same semantic in different data records have similar presentations with respect to position, size (image data item), and font (text data item).

AF3: The neighboring text data items of different semantics often (not always) use distinguishable fonts.

AF1 describes the visual similarity at the data record level. Generally, there are three types of data contents in data records, i.e., images, plain texts (the texts without hyperlinks), and link texts (the texts with hyperlinks).

The Statistics on the Visual Features

	Images (pixel)	plain texts		link texts	
		Total font number	Shared font number	Total font number	Shared font number
record1	115*115	5	5	2	2
record2	115*115	5	5	2	2
record3	115*110	5	5	2	2
record4	115*115	5	5	2	2
record5	115*115	5	5	2	2

AF2 and AF3 describe the visual similarity at the data item level. The text data items of the same semantic always use the same font, and the image data items of the same semantic are often similar in size. The positions of data items in their respective data records can be classified into two kinds: absolute position and relative position. The former means that the positions of the data items of certain semantic are fixed in the line they belong to, while the latter refers to the position of a data item relative to the data item ahead of it. Furthermore, the items of the same semantic from different data records share the same kind of position.

AF3 indicates that the neighboring text data items of different semantics often use distinguishable fonts. However, AF3 is not a robust feature because some neighboring data items may use the same font. Neighboring data items with the same font are treated as a composite data item.

Composite data items have very simple string patterns and the real data items in them can often be separated by a limited number of symbols, such as ":", "/" etc. In addition, the composite data items of the same semantics share the same string pattern. Hence, it's easy to break composite data items into real data items using some predefined separating symbols

Content feature (CF). These features hint the regularity of the contents in data records.

CF1: The first data item in each data record is always of a mandatory type.

CF2: The presentation of data items in data records follows a fixed order.

CF3: There are often some fixed static texts in data records, which are not from the underlying Web database.

The data records correspond to the entities in real world, and they consist of data items with different semantics that describe the attribute values of the entities. The data items can be classified into two kinds: mandatory and optional. Mandatory data items appear in all data records. In contrast, optional items may be missing in some data records.

This deep Web data extraction solution is developed mainly based on the above four types of visual features. PF is used to locate the region containing all the data records on a deep Web page; LF and AF are combined together to extract the data records and data items.

3.4 Special Supplementary Information

Several types of simple nonvisual information are also used in our approach in this paper. They are same text, frequent symbol, and data type.

Non visual Information Used

Special complementary information	Remarks
Same text	Given two texts, we can determine whether or not they are the same.
Frequent symbol	Given the deep web pages of a web database, if some symbols/words (e.g., ISBN, $) appear in all the data items of an attribute, they are called frequent symbols.
Data type	They are predefined, including image, text, number, date, price, email, etc

4. Data Records Extraction:

Data record extraction aims to discover the boundary of data records and extract them from the deep Web pages. An ideal record extractor should achieve the following:

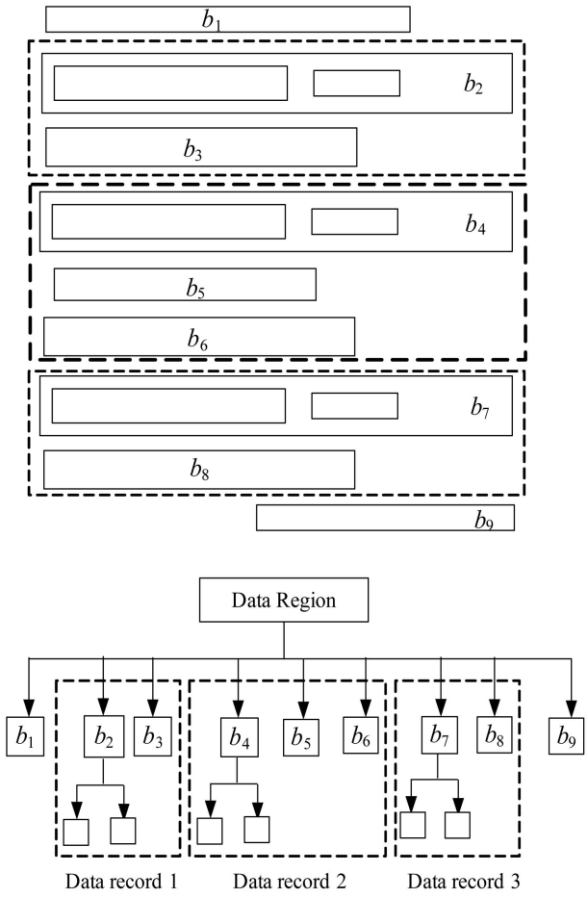
1) all data records in the data region are extracted and

2) for each extracted data record, no data item is missed and no incorrect data item is included.

Instead of extracting data records from the deep Web page directly, we first locate the data region, and then, extract data records from the data region. PF1 and PF2 indicate that the data records are the primary content on the deep Web pages and the data region is centrally located on these pages. The data region corresponds to a block in the Visual Block tree. We locate the data region by finding the block that satisfies the two position features. Each feature can be considered as a rule or a requirement. The first rule can be applied directly, while the second rule can be represented by (areab/areapage) > Tregion, where areab is the area of block b, areapage is the area of the whole deep Web page, and Tregion is a threshold. The threshold is trained from sample deep Web pages. If more than one block satisfies both rules, we select the block with the smallest area. Though very simple, this method can find the data region in the Visual Block tree accurately and efficiently.

Each data record corresponds to one or more sub trees in the Visual Block tree, which are just the child blocks of the data region. So, we only need to focus on the child blocks of the data region. In order to extract data records from the data region accurately, two facts must be considered. First, there may be blocks that do not belong to any data record, such as the statistical information (e.g. about 2,038 matching results for java) and annotation about data records (e.g., 1, 2, 3, 4, 5 (Next)). These blocks are called noise blocks.

Fig.. A general case of data region.



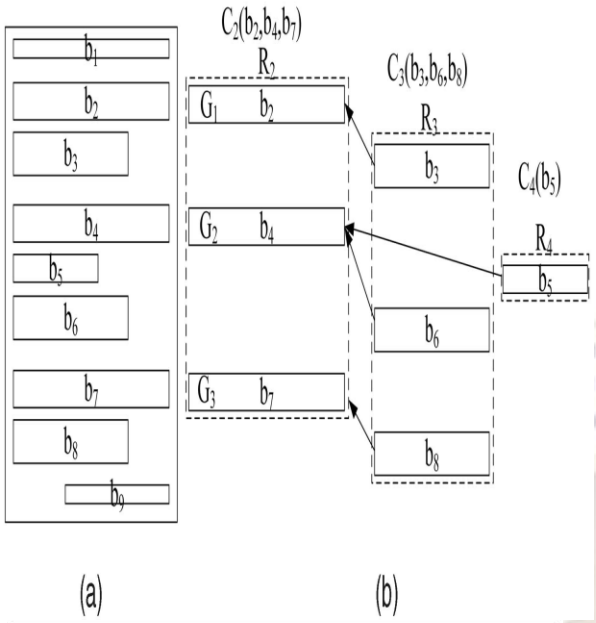
Noise blocks may appear in the data region because they are often close to the data records. According to LF2, noise blocks cannot appear between data records. They always appear at the top or the bottom of the data region. Second, one data record may correspond to one or more blocks in the Visual Block tree, and the total number of blocks in which one data record contains is not fixed.

In Fig, block b1 (statistical information) and b9 (annotation) are noise blocks; there are three data records (b2 and b3 form data record 1; b4, b5, and b6 form data record 2; b7 and b8 form data record 3), and the dashed boxes are the boundaries of data records. Data record extraction is to discover the boundary of data records based on the LF and AF features. That is, we attempt to determine which blocks belong to the same data record.

We achieve this in the following three phases:

1. Phase 1: Filter out some noise blocks.
2. Phase 2: Cluster the remaining blocks by computing their appearance similarity.
3. Phase 3: Discover data record boundary by regrouping blocks.

Fig.. An illustration of data record extraction.



Algorithm block regrouping

Input: $C_1,C_2,\ldots,C_m$: a group of clusters generated by blocks clustering from a given sample deep web page P

Output: $G_1,G_2,\ldots,G_m$: each of them corresponds to a data record on P

```
Begin
//Step 1. sort the blocks in Ci according to their positions in the page
(from top to bottom and then from left to right)
1 for each cluster Ci do
2    for any two blocks bi,j and bi,k in Ci   //1≤j<k≤|Ci|
3     if  bi,j and bi,k are in different lines on P, and bi,k is above bi,j
4          bi,j↔bi,k;      //exchange their orders in Ci;
5     else if bi,j and bi,k are in the same line on P, and bi,k is in front of bi,j
6          bi,j↔bi,k;
7 end until no exchange occurs;
8 form the minimum-bounding rectangle Reci for Ci;
//Step 2. initialize n groups, and n is the number of data records on P
9  Cmax={Ci | |Ci|=max{|C1|,|C2|,……, |Cm|}};  // n=| Cmax|
10 for each block bmax,i in Cmax
11    Initialize group Gi;
12    put bmax,i into Gi;
//Step 3. put the blocks into the right groups, and each group
corresponds to a data record
13 for each cluster Ci
14    if Reci overlaps with Recmax on P
15       if Reci is ahead of (behind) Recmax
16          for each block bi,j in Ci
17             find the nearest block bmax,k in Cmax that is behind (ahead
                  of) bi,j on the web page;
18             place bi,j into group Gk;
End
```

5. Data Item Extraction:

A data record can be regarded as the description of its corresponding object, which consists of a group of data items and some static template texts. In real applications, these extracted structured data records are stored (often in relational tables) at data item level and the data items of the same semantic must be placed under the same column.

There are three types of data items in data records: mandatory data items, optional data items, and static data items. We extract all three types of data items. Note that static data items are often annotations to data and are useful for future applications, such as Web data annotation. And even that data item extraction is different from data record extraction; the former focuses on the leaf nodes of the Visual Block tree, while the latter focuses on the child blocks of the data region in the Visual Block tree.

5.1 Data Record Segmentation

AF3 indicates that composite data items cannot be segmented any more in the Visual Block tree. So, given a data record, we can collect its leaf nodes in the Visual Block tree in left to right order to carry out data record segmentation. Each composite data item also corresponds to a leaf node. We can treat it as a regular data item initially, and then, segment it into the real data items with the heuristic rules mentioned in AF3 after the initial data item alignment.

5.2 Data Item Alignment

CF1 indicates that we cannot align data items directly due to the existence of optional data items. It is natural for data records to miss some data items in some domains. Every data record has been turned into a sequence of data items through data record segmentation. 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. In the following, we first define visual matching of data items, and then, propose an algorithm for data item alignment.

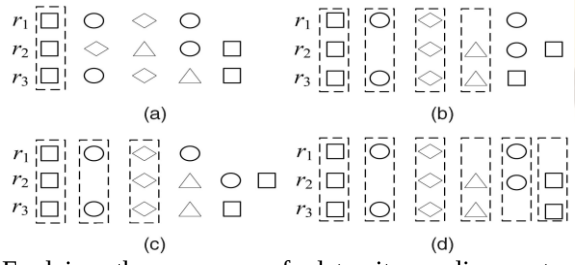
Algorithm data item alignment

Input: a set of extracted data records $\{r_i | 1 \le i \le n\}$
Output: a set of data records $\{r_i | 1 \le i \le n\}$ with all the data items aligned
Begin
1 currentItemSet=ϕ;
2 currentCluster=ϕ;
//put the first unaligned data item of each r_i into currentItemSet:
// $Item_i^{U(i)}$ refers to the first unaligned item of the ith data record
3 currentItemSet $\leftrightarrow$ $Item_i^{U(i)}$ $(1 \le i \le n)$;
4 while currentItemSet$\neq\phi$
5 use the data item matching algorithm to group the data items in currentItemSet into k clusters $\{C_i | 1 \le i \le k\}$ $(k \le n)$;
6 for each cluster C_i
7 for each r_j that does not have a data item in C_i
8 if $Item_j^{U(j)+k}$ is matched with data items in C_i
9 Log position k;
10 else
11 Log position 0;
12 P_i = max value of these logged positions for C_i;
/*Till now, each cluster C_i has a position P_i */
13 if any P_L==0
14 currentCluster=C_L;
15 else
16 currentCluster=C_L whose P_L is max $\{P_1, P_2, \ldots, P_K\}$;
17 for each r_j whose $Item_j^{U(j)}$ is in currentCluster C_L
18 remove $Item_j^{U(j)}$ from currentItemSet;
19 if $Item_j^{U(j)+1}$ exists in r_j
20 put $Item_j^{U(j)+1}$ into currentItemSet;
21 for each r_j that has no item in currentCluster C_L
22 insert a blank item ahead of $Item_j^{U(j)}$ in r_j;
23 U(j)++;
End

5.2.1 Visual Matching of Data Items

AF2 indicates that if two data items from different data records belong to the same semantic, they must have consistent font and position, including both absolute position and relative position

Example for data item alignment.



Explains the process of data item alignment. Suppose there are three data records fr1; r2; r3g and each row is a data record. We use simple geometric shapes (rectangle, circle, triangle, etc.) to denote the data items. The data items represented by the same shape are visually matched data items. We also use item $^{j}{}_{i}$ to denote the jth data item of the ith data record. Initially (Fig.a), all current unaligned data items {item11; item12; item13} of the input data records are placed into one cluster, i.e., they are aligned as the first column. Next (Fig. 10b), the current unaligned data items item21; item22; item23 are matched into two clusters C1 = {item21; item23} and C2 = {item22}. Thus, we need to further decide which cluster should form the next column. The data items in C1 can match item4 2, and the position value 2 is logged (lines 6-12), which means that item42 is the third of the unaligned data items of r2. The data items in C2 can match item31 and item33, and the position value 1 is logged. Because 1 is smaller than 2, the data items in C1 should be ahead of the data items in C2 and form the next column by inserting the blank item into other records at the current positions. The remaining data items can be aligned in the same way (Figs. 10c and 10d).

6. Visual Wrapper Generation:

ViDE has two components: ViDRE and ViDIE. There are two problems with them. First, the complex extraction processes are too slow in supporting real-time applications. Second, the extraction processes would fail if there is only one data record on the page. Since all deep Web pages from the same Web database share the same visual template, once the data records and data items on a deep Web page have been extracted, we can use these extracted data records and data items to generate the extraction wrapper for the Web database so that new deep Web pages from the same Web database can be processed using the wrappers quickly without reapplying the entire extraction process. Our wrappers include data record wrapper and data item wrapper. They are the programs that do data record extraction and data item extraction with a set of parameter obtained from sample pages. For each Web database, we use a normal deep Web page containing the maximum number of data records to generate the wrappers. The wrappers of previous works mainly depend on the structures or the locations of the data records and data items in the tag tree, such as tag path. In contrast, we mainly use the visual information to generate our wrappers.

6.1 Vision-Based Data Record Wrapper

Given a deep Web page, vision-based data record wrapper first locates the data region in the Visual Block tree, and then, extracts the data records from the child blocks of the data region.

Data region location. After the data region R on a sample deep Web page P from site S is located by ViDRE, we save five parameters values (x; y; w; h; l), where (x; y) form the coordinate of R on P, w and h are the width and height of R, and l is the level of R in the Visual Block tree. Given a new

deep Web page P_ from S, we first check the blocks at level l in the Visual Block tree for P_. The data region on P_ should be the block with the largest area overlap with R on P_. The overlap area can be computed using the coordinates and width/height information.

Data record extraction. For each record, our visual data record wrapper aims to find the first block of each record and the last block of the last data record (denoted as blast). To achieve this goal, we save the visual format (the same as the information used in (1)) of the first block of each data record extracted from the sample page and the distance (denoted as d) between two data records. For the child blocks of the data region in a new page, we find the first block of each data record by the visual similarity with the saved visual information. Next, blast on the new page needs to be located. Based on our observation, in order to help the users differentiate data records easily, the vertical distance between any two neighboring blocks in one data record is always smaller than d and the vertical distance between blast and its next block is not smaller than d. Therefore, we recognize the first block whose distance with its next block is larger than d as blast.

Vision-Based Data Item Wrapper

The basic idea of our vision-based data item wrapper is described as follows: Given a sequence of attributes {a1; a2; . . . ; an} obtained from the sample page and a sequence of data items {item1; item2; . . . ; item} obtained from a new data record, the wrapper processes the data items in order to decide which attribute the current data item can be matched to. For itemi and aj, if they are the same on f, l, and d, their match is recognized. The wrapper then judges whether itemi+1 and aj+1 are matched next, and if not, it judges itemi and aj+1. Repeat this process until all data items are matched to their right attribute

Explanation for (f; l; d)

Parameter	Value	Remarks
f	font	the font used by the data items of this attribute
l	Boolean	*True* denotes that the data items of this attribute are link texts
d	image, text, number, date, email, etc	the data type of this attribute

7. Conclusion:

In general, the desired information is embedded in the deep Web pages in the form of data records returned by Web databases when they respond to users' queries. Therefore, it is an important task to extract the structured data from the deep Web pages for later processing.

The main trait of this vision-based approach is that it primarily utilizes the visual features of deep Web pages. This approach consists of four primary steps: Visual Block tree building, data record extraction, data item extraction, and visual wrapper generation. Visual Block tree building is to build the Visual Block tree for a given sample deep page using the VIPS algorithm. With the Visual Block tree, data record extraction and data item extraction are carried out based on our proposed visual features. Visual wrapper generation is to generate the wrappers that can improve the efficiency of both data record extraction and data item extraction. Highly accurate experimental results provide strong evidence that rich visual features on deep Web pages can be used as the basis to design highly effective data extraction algorithms.

References:

[1] G.O. Arocena and A.O. Mendelzon, *"WebOQL: Restructuring Documents, Databases, and Webs,"* Proc. Int'l Conf. Data Eng. (ICDE), pp. 24-33, 1998.

[2] 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), pp. 361-370, 2001.

[3] D. Cai, X. He, J.-R. Wen, and W.-Y. Ma, "*Block-Level Link Analysis,"* Proc. SIGIR, pp. 440-447, 2004.

[4] D. Cai, S. Yu, J. Wen, and W. Ma, "*Extracting Content Structure for Web Pages Based on Visual Representation,"* Proc. Asia Pacific Web Conf. (APWeb), pp. 406-417, 2003.

[5] C.-H. Chang, M. Kayed, M.R. Girgis, and K.F. Shaalan, "*A Survey of Web Information Extraction Systems,"* IEEE Trans. Knowledge and Data Eng., vol. 18, no. 10, pp. 1411-1428, Oct. 2006.

[6] C.-H. Chang, C.-N. Hsu, and S.-C. Lui, *"Automatic Information Extraction from Semi-Structured Web Pages by Pattern Discovery,"* Decision Support Systems, vol. 35, no. 1, pp. 129-147, 2003.

[7] V. Crescenzi and G. Mecca, *"Grammars Have Exceptions,"* Information Systems, vol. 23, no. 8, pp. 539-565, 1998.

Author Biographies

C.Guru Sunanda received her B.Tech., degree in Computer science and information technology from G.Pulla Reddy Engineering College, Kurnool in 2010. She is pursuing her M.Tech in Computer scince and engineering in G.Pulla Reddy Engineering College, Kurnool. Her area of interest is in the field of data mining

Mr.K.Ishthaq Ahamed received his B.Tech., degree in Mechanical Engineering from G.Pulla Reddy Engineering College, Kurnool India, in 2000, M.tech in CSE from INDIAN SCHOOL OF MINES, DHANBAD in 2002, and currently pursuing Ph.D. in MANET. He is currently working as Associate professor in G.Pulla Reddy Engineering College, Kurnool. .His research interests include Computer Networks.