International Journal Of Advanced Research and Innovations Vol.1, Issue .1
ISSN Online: 2319 – 9253
Print: 2319 – 9245

DEEP WEB DATA EXTRACTION: A VISION-BASED APPROACH

T.Venkata Ramana [1] **Dr.K.Venugopala Rao** [2]

1. Research Scholar, JNTUH, SLC'S IET, Hyderabad.
2. Professor & Head, GNITS, AP. India.

ABSTRACT

Extracting data from the World Wide Web (WWW) has become an important issue in the last few years. Accessing hidden web or invisible web pages is [1]intricate problem by queries submitted to Web databases and the returned data records are [2]enwrapped in dynamically generated Web pages. These web pages are called [3]Deep Web Pages. Many techniques have been explained this problem with explicit limitations because they are Web-page-programming-language dependent. Web pages are always displayed regularly for users to browse in two-dimensional [4]media. This yields different ways for Deep Web Data Extraction to overcome existing techniques by utilizing some interesting common Visual features on the Deep Web Pages. In this paper explore the visual regularity of the data records and data items on deep Web pages and propose a novel vision-based approach to extract structured results from deep Web pages automatically and also propose a new evaluation measure revision to capture the amount of human effort needed to produce perfect extraction. This approach is highly effective for deep Web data extraction.

Key Words: Deep Web Data Extraction, Deep Web Pages, Structured Data, visual features of deep Web pages, wrapper generation, Vision-based Data Extractor.

1. INTRODUCTIION

Extracting data from the World Wide Web (WWW) has become an important issue in the last few years as the number of web pages available on the visible internet has grown to over 20 billion pages with over 1 trillion pages from the invisible web. A web page usually contains various contents such as navigation, decoration, interaction and contact information, which are not related to the topic of the web page.

Furthermore, a web page often contains multiple topics that are not necessarily relevant to each other. Therefore, detecting the content structure of a web page could potentially improve the performance of web information retrieval. Many web applications can utilize the content structures of web pages. For example, some researchers have been trying to use database techniques and build wrappers for web documents. *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.*

[1]intricate -Having many complexly arranged elements; elaborate
[2]enwrapped- Enclose or enfold completely with or as if with a covering and Giving or marked by complete attention to
[3]Deep Web Pages-hidden Web or invisible Web, These special Web pages are generated dynamically and are hard to index by traditional crawler based search engines, such as Google and Yahoo.
[4]medium-A state that is intermediate between extremes; a middle position

In this paper, we call this kind of special Web pages deep Web pages. Each data record on the deep Web pages corresponds to an object. For instance, Fig. 1 shows a typical deep Web page from Yahoo.com. On this page, the books are presented in the form of data records, and each data record contains some data items such as title, author, etc. In order to ease the

ISSN Online: 2319 – 9253
Print: 2319 – 9245

consumption by human users, most Web databases display data records and data items regularly on Web browsers. In this paper, we study the problem of automatically extracting the structured data, including data records and data items, from the deep Web pages

.

Fig. 1. An example deep Web page from Yahoo.com

The organization of the paper is, section 2 describes about Web Data Extraction where we get detailed explanation, section 3 gives related works of Manual Approaches, Semiautomatic Approaches, Automatic Approaches, section 4 describes visual block tree and visual features then section 5 gives clarity about data records extraction, section 5 explains data item extraction process, section 6 goes on visual wrapper generation then section 7 gives experiments of A Vision-Based Approach for Deep Web Data Extraction, finally section 8 and section 9 gives conclusions and future works and references.

2. WEB DATA EXTRACTION

A web data extraction system is a software system that automatically and repeatedly extracts data from web pages with changing content and delivers the extracted data to a database or some other application. The task of web data extraction performed by such a system is usually diVisual Wrapper Generationd into five different functions: (1) web interaction, which comprises mainly the navigation to usually pre-determined target web pages containing the desired information; (2) support for wrapper generation and execution, where a wrapper is a program that identifies the desired data on target pages, extracts the data and transforms it into a structured format; (3) scheduling, which allows repeated application of previously generated wrappers to their respective target pages; (4) data transformation, which includes filtering, transforming, refining, and integrating data extracted from one or more sources and structuring the result according to a desired output format (usually XML or relational tables); and (5) delivering the resulting structured data to external applications such as database management systems, data warehouses, business software systems, content management systems, decision support systems, RSS publishers, email servers, or SMS servers. Alternatively, the output can be used to generate new web services out of existing and continually changing web sources.

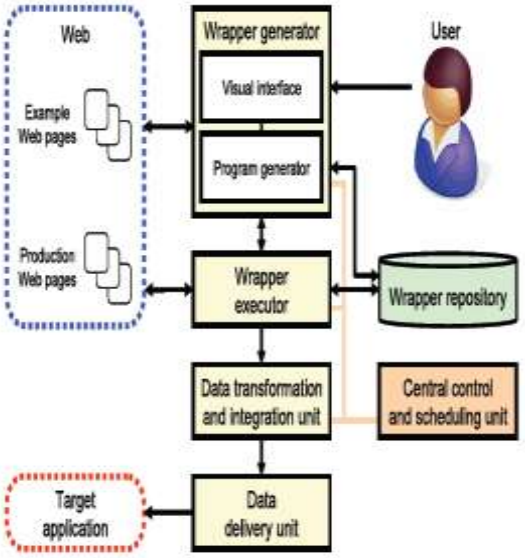

Fig. 1: Architecture of a typical web data extraction system

Our 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

ISSN Online: 2319 – 9253
Print: 2319 – 9245

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.

In this paper, we also propose a new measure, revision, to evaluate the performance of Web data extraction tools. It is the percentage of the Web databases whose data records or data items cannot be perfectly extracted (i.e., at least one of the precision and recall is not 100 percent). For these Web databases, manual revision of the extraction rules is needed to achieve perfect extraction.

In summary, this paper has the following contributions: 1) A novel technique is proposed to perform data extraction from deep Web pages using primarily visual features. We open a promising research direction where the visual features are utilized to extract deep Web data automatically. 2) A new performance measure, revision, is proposed to evaluate Web data extraction tools. This measure reflects how likely a tool will fail to generate a perfect wrapper for a site. 3) A large data set consisting of 1,000 Web databases across 42 domains is used in our experimental study. In contrast, the data sets used in previous works seldom had more than 100 Web databases. Our experimental results indicate that our approach is very effective.

3. RELATED WORK

A number of approaches have been reported in the literature for extracting information from Web pages. Good surveys about previous works on Web data extraction can be found in [16] and [5]. In this section, we briefly review previous works based on the degree of automation in Web data extraction, and compare our approach with fully automated solutions since our approach belongs to this category.

3.1 Manual Approaches

The earliest approaches are the manual approaches in which languages were designed to assist programmer in constructing wrappers to identify and extract all the desired data items/fields. Some of the best known tools that adopt manual approaches are Minerva [7], TSIMMIS [11], and Web-OQL [1]. Obviously, they have low efficiency and are not scalable.

3.2 Semiautomatic Approaches

Semiautomatic techniques can be classified into sequence based and tree-based. The former, such as WIEN [15], Soft-Mealy [12], and Stalker [22], represents documents as sequences of tokens or characters, and generates delimiter based extraction rules through a set of training examples. The latter, such as W4F [24] and XWrap [19], parses the document into a hierarchical tree (DOM tree), based on which they perform the extraction process. These approaches require manual efforts, for example, labeling some sample pages, which is labor-intensive and time-consuming.

3.3 Automatic Approaches

In order to improve the efficiency and reduce manual efforts, most recent researches focus on automatic approaches instead of manual or semiautomatic ones. Some representative automatic approaches are Omini [2], RoadRunner [8], IEPAD [6], MDR [17], DEPTA [29], and the method in [9]. Some of these approaches perform only data record extraction but not data item extraction, such as Omini and the method in [9]. RoadRunner, IEPAD, MDR, DEPTA, Omini, and the method in [9] do not generate wrappers, i.e., they identify patterns and perform extraction for each Web page directly without using previously derived extraction rules. The techniques of these works have been discussed and compared in [5], and we do not discuss them any further here. Note that all of them mainly depend on analyzing the source code of Web pages.

ISSN Online: 2319 – 9253
Print: 2319 – 9245

4. VISUAL BLOCK TREE AND VISUAL FEATURES

Before the main techniques of our approach are presented, we describe the basic concepts and visual features that our approach needs.

4.1 Visual Information of Web Pages

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

4.1.1 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 top down. 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.

4.1.2 Font

The fonts of the texts on a Web page are also very useful visual information, which are determined by many attributes as shown in Table 1. Two fonts are considered to be the same only if they have the same value under each attribute.

4.2 Deep Web Page Representation

The visual information of Web pages, which has been introduced above, can be obtained through the programming interface proVisual Wrapper Generationd by Web browsers (i.e., IE). In this paper, we employ the VIPS algorithm [4] 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.

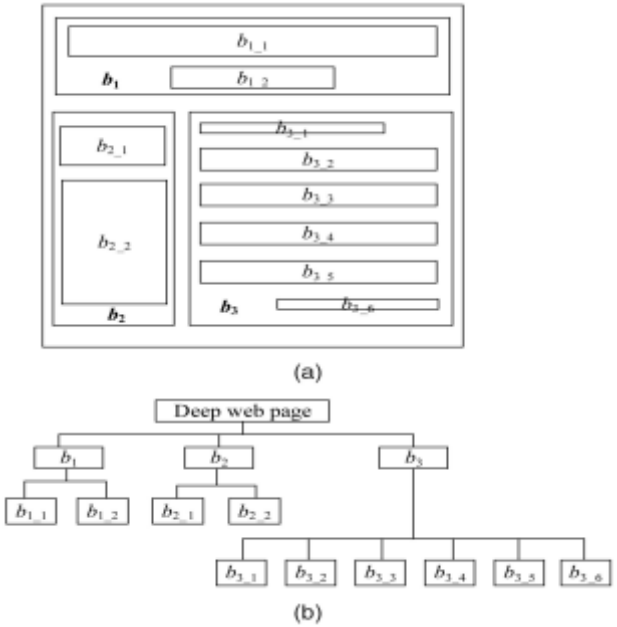

Fig.3. (a) The presentation structure and (b) its Visual Block tree.

4.3 Visual Features of Deep Web Pages

Web pages are used to publish information to users, similar to other kinds of media, such as newspaper and TV. The designers often associate different types of information with distinct visual characteristics (such as font, position, etc.) to make the information on Web pages easy to understand.

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.

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.

ISSN Online: 2319 – 9253
Print: 2319 – 9245

Data records are usually presented in one of the two layout models shown in Fig. 3. 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.

4.4 Special Supplementary Information

Several types of simple non visual information are also used in our approach in this paper. They are same text, frequent symbol, and data type, as explained in Table. Obviously, the above information is very useful to determine whether the data items in different data records from the same Web database belong to the same semantic. The above information can be captured easily from the Web pages using some simple heuristic rules without the need to analyze the HTML source code or the tag trees of the Web pages

formulas	remarks
$simIMG(b_1,b_2)=\frac{Min\{sa_i(b_1),sa_i(b_2)\}}{Max\{sa_i(b_1),sa_i(b_2)\}}$	$sa_i(b)$ is the total area of images in block b. $sa_b(b)$ is the total area of block b. $fn_{pt}(b)$ is the total number of fonts of the plain texts in block b. $sa_{pt}(b)$ is the total area of the plain texts in block b. $fn_{li}(b)$ is the total number of fonts of the link texts in block b. $sa_{li}(b)$ is the total area of the link texts in block b.
$w_i=\frac{sa_i(b_1)+sa_i(b_2)}{sa_b(b_1)+sa_b(b_2)}$	
$simPT(b_1,b_2)=\frac{Min\{fn_{pt}(b_1),fn_{pt}(b_2)\}}{Max\{fn_{pt}(b_1),fn_{pt}(b_2)\}}$	
$w_{pt}=\frac{sa_{pt}(b_1)+sa_{pt}(b_2)}{sa_b(b_1)+sa_b(b_2)}$	
$simLT(b_1,b_2)=\frac{Min\{fn_{li}(b_1),fn_{li}(b_2)\}}{Max\{fn_{li}(b_1),fn_{li}(b_2)\}}$	
$w_{li}=\frac{sa_{li}(b_1)+sa_{li}(b_2)}{sa_b(b_1)+sa_b(b_2)}$	

6. DATA ITEM EXTRACTION

A data record can be regarded as the description of its corresponding object, which consists o 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. When introducing CF, we mentioned that 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. Below, we focus on the problems of segmenting the data records into a sequence of data items and aligning the data items of the same semantics together. Note 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.

7. VISUAL WRAPPER GENERATION

Visual Wrapper Generation 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.

ISSN Online: 2319 – 9253
Print: 2319 – 9245

In contrast, we mainly use the visual information to generate our wrappers.

8. EXPERIMENTS

We have implemented an operational deep Web data extraction system for Visual Wrapper Generation based on the techniques we proposed. Our experiments are done on a Pentium 4 1.9 GH, 512 MB PC. In this section, we first describe the data sets used in our experiments, and then, introduce the performance measures used. At last, we evaluate both ViDRE and ViDIE. We also choose MDR [17] and DEPTA [29] to compare with ViDRE and ViDIE, respectively. MDR and DEPTA are the recent works on Web data record extraction and data item extraction, and they are both HTML-based approaches.

8.1 Data Sets

Most performance studies of previous works used small data sets, which are inadequate in assuring the impartiality of the experimental results. In our work, we use a large data set to carry out the experiments.

GDS. This data set is collected from CompletePlanet(www.completeplanet.com), which is currently the largest deep Web repository with more than 70,000 entries of Web databases. These Web databases are classified into 42 categories covering most domains in the real world. GDS contains 1,000 available Web databases. For each Web database, we submit five queries and gather five deep Web pages with each containing at least three data records.

8.2 Performance Measures

All previous works use precision and recall to evaluate their experimental results (some also include F-measure, which is the weighted harmonic mean of precision and recall). These measures are also used in our evaluation. In this paper, we propose a new metric, revision, to measure the performance of an automated extraction algorithm. It is defined to be the percentage of the Web databases whose data records or data items are not perfectly extracted, i.e., either precision or recall is not 100 percent. This measure indicates the percentage of Web databases the automated solution fails to achieve perfect extraction, and manual revision of the solution is needed to fix this. An example is used to illustrate the significance of this measure.

8.3 Experimental Results on ViDRE

In this part, we evaluate ViDRE and also compare it with MDR. MDR has a similarity threshold, which is set at the default value (60 percent) in our test, based on the suggestion of the authors of MDR. Similarity threshold, which is set at 0.8. In this experiment, the input to ViDRE and MDR contains the deep Web pages and the output contains data records extracted. For ViDRE, one sample result page containing the most data records is used to generate the data record wrapper for each Web database. Table 8 shows the experimental results on both GDS and SDS. Based on our experiment, it takes approximately 1 second to generate the data record wrapper for each page and less than half second to use the wrapper for data record extraction.

	dataset	*precision*	*recall*	*revision*
ViDRE	GDS	98.7%	97.2%	12.4%
	SDS	98.5%	97.8%	10.9%
MDR	GDS	85.3%	53.2%	55.2%
	SDS	78.7%	47.3%	63.8%

Comparison Results between ViDRE and MDR

	dataset	*precision*	*recall*	*revision*
ViDIE	GDS	96.3%	97.2%	14.1%
	SDS	95.6%	98.4%	11.6%
DEPTA	GDS	75.3%	71.6%	32.8%
	SDS	66.1%	54.1%	37.6%

Comparison Results between ViDIE and DEPTA

8.4 Experimental Results on ViDIE

In this part, we evaluate ViDIE and compare it with DEPTA. DEPTA can be considered as the follow-up work for MDR, and its authors also called it MDRII. Only correct data records from

ISSN Online: 2319 – 9253
Print: 2319 – 9245

ViDRE are used to evaluate ViDIE and DEPTA. For ViDIE, two sample result pages are used to generate the data item wrapper for each Web database. Table 9 shows the experimental results of ViDIE and DEPTA on both GDS and SDS. Our experiments indicate that it takes between 0.5 and 1.5 seconds to generate the data item wrapper for each page and less than half second to use the wrapper for data item extraction.

9. CONCLUSIONS AND FUTURE WORKS

With the flourish of the deep Web, users have a great opportunity to benefit from such abundant information in it. 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. In this paper, we focused on the structured Web data extraction problem, including data record extraction and data item extraction.

First, we surveyed previous works on Web data extraction and investigated their inherent limitations. Meanwhile, we found that the visual information of Web pages can help us implement Web data extraction. Based on our observations of a large number of deep Web pages, we identified a set of interesting common visual features that are useful for deep Web data extraction. Based on these visual features, we proposed a novel vision-based approach to extract structured data from deep Web pages, which can avoid the limitations of previous works. The main trait of this vision-based approach is that it primarily utilizes the visual features of deep Web pages.

Our 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 proVisual Wrapper Generation strong eVisual Wrapper Generationnce that rich visual features on deep Web pages can be used as the basis to design highly effective data extraction algorithms.

However, there are still some remaining issues and we plan to address them in the future.First, Visual Wrapper Generation can only process deep Web pages containing one data region, while there is significant number of multidata-region deep Web pages. Though Zhao et al. [31] have attempted to address this problem, their solution is HTML-dependent and its performance has a large room for improvement. We intend to propose a vision-based approach to tackle this problem. Second, the efficiency of Visual Wrapper Generation can be improved. In the current Visual Wrapper Generation, the visual information of Web pages is obtained by calling the programming APIs of IE, which is a time-consuming process. To address this problem, we intend to develop a set of new APIs to obtain the visual information directly from the Web pages.

REFERENCES

[1] V. Anupam, J. Freire, B. Kumar and D. Lieuwen. Automating web navigation with the WebVCR. *Computer Networks*, 33(1-6): 503-517, 2000.

[2] R. Baumgartner, S. Flesca and G. Gottlob. Visual web information extraction with Lixto. *VLDB*, 2001.

[3] V. Crescenzi, G. Mecca and P. Merialdo. RoadRunner: Towards automatic data extraction from large Web sites. *VLDB*, 2001.

[4] O. Etzioni, M.J. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D.S. Weld, and Y. Yates. Web-scale information extraction in KnowItAll: (preliminary results), *WWW*, 2004.

[5] A.H.F. Laender, B.A. Ribeiro-Neto and A.S. da Silva. DEByE – Data extraction by example. *Data and Knowledge Engineering*, 40(2): 121-154, 2000.

ISSN Online: 2319 – 9253
Print: 2319 – 9245

[6] Sahuguet and F. Azavant. Building intelligent web applications using lightweight wrappers. *Data and Knowledge Engineering*, 36:(3) 283-316, 2001.

[7] S. Kuhlins and R. Tredwell. Toolkits for Generating Wrappers: A Survey of Software Toolkits for Automated Data Extraction from Websites. NODe 2002,LNCS:2591, 2003.

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

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.