ISSN:0975-9646

D. Raghu et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (6) , 2011, 2734-2736

Dynamic Vision-Based Approach in Web Data Extraction

D. Raghu
Department of CSE, Nova College of Engineering and Technology, Jangareddy Gudem, AP

V. Sridhar Reddy
Nova College of Engineering and Technology, West Godavari Dist.,AP

Ch. Raja Jacob
Nova College of Engineering and Technology,West Godavari, AP.

Abstract -The problem of extracting data records on the response pages returned from web databases or search engines. World Wide Web has posed a challenging problem in extracting relevant data. Traditional web crawlers focus only on the surface web while the deep web keeps expanding behind the scene. Deep web pages are created dynamically as a result of queries posed to specific web databases. Extracting structured data from deep Web pages is a challenging problem due to the underlying intricate structures of such pages. The 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 or independent. 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. This paper, a novel and vision-based approach for extracting data from the deep web. Deep splits the process into two phases. The first phase includes Query analysis and Query translation and the second covers vision-based extraction of data from the dynamically created deep web pages. There are several established approaches for the extraction of deep web pages but the proposed method aims at overcoming the inherent limitations, it aims to comparing the data items and presenting them in the proper order.

Keywords : Deep web, Web Database, response page, data record.

I. INTRODUCTION

World Wide Web has close to one million searchable information sources . These searchable information sources include both search engines and Web databases. Web database keeps expanding every day, which drives the focus on researches towards deep web mining. The information in a web database can be fetched only through its web query interface. These Web databases are queried for particular information and the query result is enwrapped to form a dynamic web page called the deep web page. It is almost impossible for the search engines to retrieve this information and hence this is called deep web or hidden web. The result objects obtained from the query submitted, is displayed in the form of data records. the retrieved information (query results) is wrapped on response pages returned by these systems in the form of data records, each of which corresponds to an entity such as a document or a book. Data records are usually displayed visually neatly on Web browsers to ease the consumption of users. When the user submits a query in their search interface, a page is created dynamically which has a list of mobiles that matches the query. This dynamically created page is an example of deep web page. Each mobile detail is displayed in the form of structured data records; each data record contains data items like price, discount, features, color, etc. Data records are structured not only for the ease of humans but also for many applications like deep web crawling were data items need to be extracted from the deep web page. Recently the deep web crawling has gained a lot of attention and many methods have already been proposed for data record extraction from deep web pages. But these proposed methods are structure-based; either based on analyzing HTML codes or the tag types of the web pages. the inherent limitation of They are dependent on the programming language of the web page. Most of these methods are meant for HTML. Even if we assume that only HTML is used to write all the web pages, the previously proposed methods are not fully efficient and fool proof. The evolution of HTML is non-stop and hence the addition of any new tag will require amendment in the previous works in order to adapt to the new version.

II. RELATED WORK

The many approaches have been reported in the literature for extracting information from Web pages. Recently, many automatic approaches [5][6][7][8] have been proposed instead of manual approaches [2] and semi-automatic approaches [3] [4]. For example, [6] and patterns or grammars from multiple pages in HTML DOM trees containing similar data records, and they require an initial set of pages containing similar data records. In [5], a string matching method is proposed, which is based on the observation that all the data records are placed in a specific region and this is reacted in the tag tree by the fact that they share the same path in DOM tree. The method DEPTA[7] used tree alignment instead of tag strings, which exploits nested tree structures to perform more accurate data extraction, so it can be considered as an improvement of MDR[8]. The only works that we are aware of that utilize some visual information to extract data records are [13][14].

However, in these approaches, tag structures are still the primary information utilized while visual information plays a small role. For example, in [13], when the visual information is not used, the recall and precision reduce by only 5%. In contrast, in this paper, our approach performs data record extraction completely based on visual information. Although the works discussed above applied different techniques and theories, they have a common characteristic: they are all implemented based on HTML DOM trees and tags by parsing the HTML documents. In Section 1, we discussed the latent and inevitable limitations of them. Since web pages are used to publish information for humans to browse and read, the desired information we want extracted must be visible, so the visual features of web pages can be very helpful for web information extraction. Currently, some works are proposed to process web pages based on their visual representation. For example, a web page segmentation algorithm VIPs is proposed in [9] which simulates how a user understands web layout structure based on his/her visual perception. Our approach is implemented.

Features of Visual Deep Web Pages

Web pages are used to publish information to users, similar to other kinds of media. 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. As a result, 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. Our observation based on a large number of deep Web pages is consistent with this hypothesis. We describe the main visual features in this section and show the statistics about the accuracy of these features at the end. 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. By investigating a large number of deep Web pages, we found 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, our approach uses the ratio of the size of the data region to the size of whole deep Web page instead of the actual size. In our experiments in the threshold of the ratio is set at 0.4, that is, if the ratio of the horizontally centered region is greater than or equal to 0.4, then the region is recognized as the data region.

III. DYNAMIC DEEP DATA RECORD EXTRACTOR

Dynamic Deep has two components, the Deep data record extractor and the Deep data item extractor . Since dynamic deep makes use of the visual features, it overcomes the limitations of existing works. Dynamic deep employs four steps for data extraction from deep web page. First, it takes sample deep web page from a particular web database and obtains its visual representation. Later this is converted to a Visual Block Tree. Second, data records are extracted from visual block tree. Third, this extracted data records are further partitioned into data items and the similar data items (semantically related) are clustered together. Fourth, the most important step is the generation of visual wrappers, which will extract the data items and data records from the other deep web pages which are dynamically created by the same web database.

IV. DATA EXTRACTION

To extract a data record from a deep web page, first, the boundary of the data records must be discovered. The record extraction process must follow two rules: 1) for every data region that is considered, all the data records in it must be extracted 2) every valid data item in an extracted record must be included and not single incorrect data item should be included. To extract data records from the deep web page, first the data region is located then the data records from the region are extracted. Data region is a rectangular region that includes data records of the web page. Data region corresponds to a block in the visual block tree. To identify the data region, rules R3 and R4 are employed. Rule R4 can be implemented as a formula that is given by- [6] *(areablock/areapage)* > *Tregion,* where *Tregion* threshold which is trained from sample deep web pages. In case there is more than one block in the visual block tree that satisfies these two rules, the one with smallest area is chosen. The data regions in the visual block tree can be found efficiently and accurately by this method. Data records are the child blocks of data regions, so it is enough that we concentrate only on the child blocks of data region. To accurately extract data records from a data region two facts must be considered. One, noise blocks and two, the number of blocks that makes one data region. Noise blocks are blocks that do not belong to any data record but they are contained in data region. Noise blocks can be statistical information (e.g. 1000 matching results for "user's search *keyword"*) or they can be annotation (e.g. 1, 2, 3, (prev), (next)). The second fact that must be considered is that the number of blocks that makes a data record is not fixed. Many blocks in the visual block tree might correspond to a single data record. For example, in block B2 and B3 belong to same data record while B4, B5 and B6 correspond to another data record.

Filtering the Noise

Noise blocks do not appear in between the data records, they appear either at the top or bottom of the data region (inference of rule R5). According to R5, data records are usually aligned flush left in the data region, so all the blocks that are not aligned flush left are considered as noise and are filtered. But this step does not ensure the filtering of all the

noise blocks. For example, in Fig3 blocks B1 and B9 are noise blocks but in this step only B9 will be removed as B1 is aligned flush left.

Visual -based clustering

The rest of the blocks are considered useful blocks and are clustered based on their appearance. Items in data records can be primarily classified into two: text and image. Images of two data records can be considered similar if they are of the same size and text similarity is based on same font attributes. Text can be further divided into plain text and link text.

V. PERFORMANCE EVOLUTION

Two measures, precision and recall, are widely used to measure the performance of data record extraction algorithms in published literatures. Precision is the percentage of correctly extracted records among all extracted records and recall is the percentage of correctly extracted records among all records that exist on response pages. In our experiments, a data record is correctly extracted only if anything in it is not missed and anything not in it is not included. Besides precision and recall, there is an important measure neglected by other researchers. It is the number of websites with perfect precision and recall, i.e., both precision and recall are 100% at the same time. This measure has a great meaning for web data extraction in real applications. We give a simple example to explain this. Suppose there are three approaches (A1, A2 and A3) which can extract data records from response pages, and they use the same data set (5 web sites, 10 data records in each web site). A1 extracts 9 records for each site and they are all correct. So the average precision and recall of A1 are 100% and 90%, respectively. A2 extracts 11 records for each site and 10 are correct. So the average precision and recall of A2 are 90.9% and 100%, respectively. A3 extracts 10 records for 4 of the 5 sites and they are all correct. For the 5th site, A3 extracts no records. So the average precision and recall of A3 are both 80%. Based on average precision and recall, A1 and A2 are better than A3. But in real applications A3 may be the best choice. The reason is that in order to make precision and recall 100%, A1 and A2 have to be manually tuned/adjusted for each web site, while A3 only needs to be manually tuned for one web site. In other words, A3 needs the minimum manual intervention.

VI. CONCLUSION

Dynamic Deep web has abundant information in it. To tap these resources, we need an efficient method to get the desired information which is embedded in the deep web pages. The structured data that is extracted can be used for processing in web based applications in real time. The paper effectively extracts the dynamic deep web data records and data items using visual features. In this paper we create a database of deep web pages of different domains, which will have to be updated frequently. This process of update will require an effective algorithm to maintain the efficiency of the system. The future works can be done in integrating this feature in this proposed method.

REFERENCES

[1] B.Liu,R.L.Grossman and Y.Zhai "Mining Data Record in Web Pages" *SIGKDD .03*, August 24-27, 2003, Washington, DC, USA

[2] D.Cai , S.Yu, J.Wen and W.Ma,"Extracting Content Structure for Web Pages Based on Visual Representation"

[3] Wei Liu and X. Meng "ViDE: A Vision-Based Approach for Deep Web Data Extraction" IEEE Transactions On Knowledge And Data Engineering, Vol. 22, No. 3, March 2010

[4] Jayant Madhavan, David Ko, Łucja Kot, Vignesh Ganapathy, Alex Rasmussen and Alon Halevy. "Google's DeepWeb Crawl". PVLDB '08, August 23-28, 2008, Auckland, New Zealand

[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.

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

[9] D.W. Embley, Y.S. Jiang, and Y.-K. Ng, "Record-Boundary Discovery in Web Documents," Proc. ACM SIGMOD, pp. 467- 478, 1999.

[10] N. Kushmerick, "Wrapper Induction: Efficiency and Expressiveness," Artificial Intelligence, vol. 118, nos. 1/2, pp. 15-68, 2000.

[11] A. Laender, B. Ribeiro-Neto, A. da Silva, and J. Teixeira, "A Brief Survey of Web Data Extraction Tools," SIGMOD Record, vol. 31, no. 2, pp. 84-93, 2002.

[12] B. Liu, R.L. Grossman, and Y. Zhai, "Mining Data Records in Web Pages," Proc. Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 601-606, 2003.

[13] W. Liu, X. Meng, and W. Meng, "Vision-Based Web Data Records Extraction," Proc. Int'l Workshop Web and Databases (WebDB '06)pp. 20-25, June 2006.

[14] L. Liu, C. Pu, and W. Han, "XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources," Proc. Int'l Conf. Data Eng. (ICDE), pp. 611-621, 2000.

[15] Y. Lu, H. He, H. Zhao, W. Meng, and C.T. Yu, "Annotating Structured Data of the Deep Web," Proc. Int'l Conf. Data Eng. (ICDE), pp. 376-385, 2007.

[16] J. Madhavan, S.R. Jeffery, S. Cohen, X.L. Dong, D. Ko, C. Yu, and A. Halevy, "Web-Scale Data Integration: You Can Only Afford to Pay As You Go," Proc. Conf. Innovative Data Systems Research (CIDR), pp. 342-350, 2007.

[17] I. Muslea, S. Minton, and C.A. Knoblock, "Hierarchical Wrapper Induction for Semi-Structured Information Sources," Autonomous Agents and Multi-Agent Systems, vol. 4, nos. 1/2, pp. 93-114, 2001.

[18] Z. Nie, J.-R. Wen, and W.-Y. Ma, "Object-Level Vertical Search," Proc. Conf. Innovative Data Systems Research (CIDR), pp. 235-246, 2007.[19] A. Sahuguet and F. Azavant, "Building Intelligent Web Applications Using Lightweight . 36, no. 3, pp. 283-316, 2001.