VWDRE – A VISION-BASED APPROACH FOR MINING DATA FROM SEARCH ENGINE RESULT PAGES

M. Sathya
Assistant Professor, Dept of Information Technology,
Veltech Dr.RR & Dr.SR University, Chennai, India

S. Madhan
Assistant Professor, Dept of CSE, University College of Engineering,
Thirukkuvalai, Tamilnadu, India

U. Jothilakshmi
Assistant Professor, Dept of Information Technology,
Veltech Dr.RR & Dr.SR University, Chennai, India

ABSTRACT

The data extraction from the dynamically generated web pages is a challenging factor because the result of the search engines are always different for every query submitted. Many techniques were proposed to address this issue but most of them have the common problem of language-dependency. In order to overcome the limitations of previous works, there are few ways which analyze visual features of the web page. In this paper, we proposed a new vision-based approach which is independent of the code used. It broadly utilizes the visual features on the search engine result pages to locate the data region so as to mine the data records from it. We develop a clustering by similarity algorithm to check the similarity of data records. Also, we propose a technique to generate the wrapper for data record extraction by examining the multiple result pages from the same search engine.

Key words: Vision-Based, Wrapper, Data Extraction, Search Engine Result Pages, DOM tree.


1. INTRODUCTION

The World Wide Web has more and more amount of information with the tremendous growth. Retrieving the required information from the WWW is simple and also quick by entering queries. The Web servers process the queries and return the needed data records
enwrapped in HTML pages in the form of data records. These Web pages are generated dynamically and are it is difficult to index by traditional crawler-based search engines, such as Google and Yahoo. In the search engine result pages, advertisements are also displayed with some pre-defined templates in order to make human browsing easier. In such web pages, the data records are grouped into data regions. In order to make the data records machine processable, the structured data records are needed. From the search engine result pages, such data records are to be extracted. The necessity of automatic wrapper lies in automating the meta search engine and for the evaluation and comparison of shopping lists.

From the study on Web data extraction, we infer that the most of the proposed methods focused on analyzing the HTML source code and or the tag trees of the web pages. Also, the proposed solutions stuck with the following limitations: First, they are programming language dependent. Mostly, HTML-dependent. When new versions of HTML or new tags are introduced, the previous works will have to be amended repeatedly to adapt to new versions or new tags. Second, these methods are not capable of handling the complexity of HTML source code of Web pages. To make the web page more presentable, web designers use scripts such as JavaScript and CSS in the HTML files. These scripts were not considered while extracting data. By observing large number of search engine result pages, the underlying structure of current Web pages is more complicated than ever and is far different from their layouts on Web browsers. Hence it is more difficult for proposed solutions to infer the regularity of the structure of Web pages by only analyzing the tag structures.

To make the users easy to acquire the information retrieved from search engines, a good template is needed for search engine result pages to arrange the data records with visual regularity. Unfortunately, current visual wrappers require many sample pages to accurately predict the data region. The accuracy of the visual wrappers can be improved by improving the similarity check performance and data detection algorithms of a wrapper.

By examining the simple statistical properties of a DOM Tree and referring to conclusions made by others [1], [4], [6], [8], [14], [15],[17], we find that it is possible to filter out irrelevant information in a web page and formulate a similarity check based on the properties of the DOM Tree. By making full use of DOM tree properties and visual features of data records, a wrapper called Visual Wrapper for Data Record Extraction (VWDRE) is developed which facilitates the users to identify the data regions easily and accurately. It reduces the difficulty of analyzing complex Web page source files.

Our approach employs a three-step strategy. First, Visual features of the search engine result pages are considered and DOM tree is constructed. Second, identify data regions from the DOM tree and finally generate the visual wrapper which is a set of visual extraction rules based on search engine result page such that the data record extraction can be efficiently performed by our visual wrappers.

For the wrapper generation, we propose a new algorithm called clustering by similarity algorithm that is to exactly locate data from the DOM tree. The accuracy of vision-based wrappers can be improved by improving the performance of the similarity check and data region detection algorithms of a wrapper. Our statistical study on the properties of the DOM tree and referring to conclusion made by [2], [4], [6], [8], [14], [15], [17], we discover that we can filter out the irrelevant information present in the web page and also formulate a similarity check based on DOM tree properties.
2. RELATED WORK

Many approaches have been analyzed for data extraction from the search engine result pages. Good surveys about the existing work can be found in [10] and [7]. The visual wrappers perform better in the aspect of locating data rather than analyzing the HTML tag structures only. Mostly, the current visual wrappers focus on visual cues whereas which also fails in the aspect of ignoring the repeated data in the search engine result pages.

The early approaches tried to do the data extraction manually in which the programmers analyzed the search engine result pages to locate their desired data or fields. But these ways differ for every search engine and also for every item searched in the search engine. Some popular tools that follow manual approaches are Minerva [21], TSIMMIS [22], and Web-OQL [23]. They are less efficient and are also not scalable.

There are some semiautomatic approaches which fall into two categories sequence-based and tree-based. The sequence-based approaches are WIEN [20], and Stalker [24] which represented documents as sequences of tokens or characters, and generated delimiter-based extraction rules through a set of training examples. The tree-based approaches are W4F [18] and XWrap [19] which parse the document into a DOM tree and then extract the needed data or fields. However, these approaches additionally require manual efforts, for example, labeling few sample pages, which is labor-intensive and also time-consuming.

The efficiency of visual wrappers has to be improved and the manual efforts must be reduced. The existing automatic approaches are Omini [11], RoadRunner [12], IEPAD [9], MDR [13], DEPTA [14], and the method in [25]. Few approaches such as Omini and the method in [25] perform data extraction which included noise data also. The methods proposed in Omini, DEPTA, RoadRunner, IEPAD, MDR, and the method in [25] do not generate wrappers, i.e., they do extraction for every search engine result page every time without using the extraction rules that was derived already.

In addition to these approaches, there are several works (DeLa [26], DEPTA, and the method in [25]) do data extraction by assigning labels to the nodes of DOM tree. DeLa used HTML tag structure for constructing regular expression wrapper and extract data items. Like DeLa, DEPTA also operates on HTML tag tree structures to arrange the data records and then match the records. However, both of these data alignment techniques are mainly based on HTML tree structures but not vision-based. In the method given in [20], a clustering approach is proposed to perform alignment based on five features of data items, including font of text. However, this method is primarily text-based and HTML tag-structure-based, whereas our method is primarily vision-based.

The vision-based data extraction was utilized in the works such as ViNTS [27], ViPERS[17], HCRF[16], and VSDR [28]. ViNTs use the visual features on the query result pages to capture content regularities given as Content Lines, and use the HTML tags to combine them. ViPER also uses visual information on a Web page for data extraction along with the help of a multiple sequence alignment technique. But, in these two approaches, tag structures are still the primary information used, whereas the visual information was not much used. In the method HCRF, both data record extraction and attribute labeling are focused. The method VSDR [28] uses the visual observation that, in general, the data records are found at the center of the search engine result page and this center location is rendered by the web browser. This wrapper computes the visual center of a HTML page and constructs a boundary region by itself to encompass the data area. But, the method VSDR may overestimate the
boundary region size and may include some non-record data. Spacing is a key factor for this method VSDR to locate and extract the data records.

This paper contains several sections. Section 2 describes the current work related to our work. Section 3 gives interesting visual information from the search engine result pages. Section 4 provides the implementation details of our wrapper. Finally Section 5 summarizes our work.

![Figure 1 Extraction process](image)

**3. VISUAL OBSERVATIONS FROM SEARCH ENGINE RESULT PAGES**

On analyzing the papers [6], [8], [13], [14], [15], [17], [18] for information extraction in search engine result pages, several unique features inherent to a data record were pointed out. The information on Web pages consists of both texts and images, may be static pictures, flash, video, etc. The visual information of Web pages includes information mostly related to Web page layout like location, size and font. The visual information of the Web pages can be taken from the programming interface of the Web browsers. Also we made several observations based on the constitution of a data record. After considering all these observations, we come out with a way to formulate the simple statistics of a DOM tree so that to extract data records correctly. The following are the observations given by several authors as presented in their papers:

**Observation 1**

The papers [6], [8], [15], [17], [18] specified about this observation which is related to the position or location of the data records in the search engine result page. It states:

The data records in the search engine result pages are placed horizontally in the center. The size of the data records is always found to be large with respect to the size of the full page.

**Observation 2**

The papers [6], [8], [13], [14], [15] pointed out this observation which is related to the layout of the search engine result page. It states:
In general, the data records are aligned in the left portion of the data region. The data records are adjoined with uniform distance in between.

**Observation 3**

The papers [6], [8], [13], [14], [15] specified about this observation which describe about appearance of the data records. It states:

Data Records usually fall under a specific regular expression rule to represent their individual data; and so they have nearly similar tree structure.

**Our Observation**

Along with visual information, we notice that similar text, frequent symbols and same set of data types are used in the search engine result pages.

We carefully examined the above four observations and find that these criteria could be formulated using visual information and statistical frequency measures of the DOM tree structure of data records. Three steps of filtering rules are proposed in this paper, considering the above observations. These observations will be taken as part of the requirements for VWDRE wrapper to extract data records from web page.

4. **VWDRE WRAPPER**

VWDRE consists of two main components. The first component involves parsing the HTML page and organizing it into Document Object Model (DOM) tree representation. In the second component, VWDRE extracts data records using visual observations and DOM properties of the data records. In this stage (Figure 2) VWDRE goes through three stages for the extraction of data records. After all the successful filtering processes, we will end up with only one data region in which only the correct data records left.

![Figure 2 Presentation structure](image)

4.1. **DOM Tree Construction**

Initially, the HTML web page of the search engine result pages is parsed and aligned them into a DOM tree. To simplify our wrapper generation, it is assumed that the search engine result page under extraction must contain at least 3 repetitive patterns. This assumption is based on the observations stated above that the majority of response pages contain more than three repetitive patterns. Those repetitive patterns can referred as potential data records.
4.2. Data Records Extraction

This process focuses to exactly locate the boundary of the data records and extract them from the search engine result pages. An ideal data record extractor should do the following: 1) the HTML tags in the web page are filtered, 2) similar data records are identified and grouped and 3) all the data records in the data region are extracted. Also we should ensure that we don’t include any false data record and similarly we should not miss any valid data record. The process is as follows:

Phase 1: HTML tags filtering
Phase 2: Clustering the data records by similarity
Phase 3: Discover individual data records

Phase 1: HTML tags filtering

In this process, at first, we should identify the data region by filtering out the HTML tags and then we do the extraction process. Observation 1 indicates that the data records are always the primary content of the search engine result page and also data region is in the center of the web page. The data region is put as a block in the DOM tree. Using the observation1 we locate the data region in the center. Each observation can be considered as a rule or a requirement.

The observation1 can be expressed as (area of node/area of whole page) > T where T is a Threshold. The Threshold can be predicted by training several search engine result pages. If we find that more than one node satisfies this, we can find the data region in the DOM tree accurately and also efficiently. The data records which correspond to more one than sub tree in DOM tree are considered as the child nodes of the data regions as Fig. 2 shows.

Phase 2: Clustering by similarity

Using observation 2, we group the data records by their similarity. In phase 1, we located the data region, using which we are going to do the clustering of records. Observation 2 states that the data records are aligned in the left portion of the data region. And also it adds that the data records are adjoined with uniform distance in between. Noise data may appear in the data region as they are placed close to the data records. But the noise data cannot appear in between the data records because the data records are uniformly separated. There are few chances that the data record may belong to more than one nodes in DOM tree and so the total nodes containing individual data record may vary.
In Fig. 2, node n1 (statistical information) and b9 are noise data; there are three data records (n2 and n3 form data record 1; n4, n5, and n6 form data record 2; n7 and n8 form data record 3), and the dashed boxes are the boundaries of data records.

Observation 3 states that, the data records usually fall under a specific regular expression rule to represent their individual data. There are usually three types of information occur mostly in the search engine result pages, i.e., images, plain text, and link text. The similar blocks are identified on the basis of size of the image and the font used. If the data is an image, we consider size for finding similarity. For plain text and link text, we focus on font style. Intuitively, if two blocks are found to be similar on image size and font, they should be more similar in appearance.

A simple clustering by similarity algorithm is used to group the data records. First, begin with an arbitrary order of the given nodes or blocks, take first node from the list and cluster further. Next, for each of the remaining nodes, say b, compute its similarity with each existing cluster. This algorithm is explained in Fig 3. The algorithm consists of three steps. In step 1, nodes in each group is rearranged on the basis of their appearance order in the search engine result page i.e., from left to right and also the minimum bounding rectangle is formed. Step 2 initializes m number of groups and m is take as number of data records in the data region P. In step 3, the corresponding cluster of every node is decided.
Phase 3: Discover individual data records

The clusters identified in the previous step do not connect to the exact data record. In this phase, the nodes are rearranged such that, the nodes correspond to the same data record aim at forming the same cluster. Our observation states that, similar type of data records use similar text, frequent symbols and same set of data types. Therefore, each data record is taken out and the particular cluster it belongs to is then identified. And finally, we can obtain the similar type of data records which is in the search engine result page. For example, we know that cost of the product is ahead of product name in each record and they belong to different nodes. Hence, to find the order of different nodes in the search engine result page, a minimum bounding rectangle is built for each cluster on the page. And finally, we can obtain the similar type of data records which is in the search engine result page.

4.3. VWDRE Generation

All the search engine result pages from the same database share the same visual template. Hence, once the data records have been extracted, we can follow the same extraction rules to generate the wrapper for extraction. It will be easier to extract the data records from such search engine result pages as there is no need to apply the entire extraction rules again. Generally, the wrappers are the programs for data extraction with a set of rules obtained from sample web pages. For different search engines and for different set of searches, we use the sample search engine result pages containing the maximum number of data records in order to generate the wrappers. The previous works mainly used the HTML tags and structures of the web page for generating the wrapper. In contrast, in this paper we gave importance to the visual information to generate wrapper for data records extraction.

Our Visual Wrapper for Data Record Extraction locates the data region in the DOM tree and then extracts the data records from the child nodes of the DOM tree. The data region R is located by VWDRE wrapper using the parameters such as x, y coordinates of the data region, width and height of the web page and level of the data region R in the DOM tree. The overlapping area must also be resolved with the coordinates and width/height information. For every page, our visual wrapper tends to find the first data record and last data record so as to refine further. The visual information such as the location of very first data record and the last data record in the web page and the distance between the two data records must be saved for wrapper generation. Using our observations, the users can easily differentiate the data records and obtain the vertical distance between any two neighboring nodes in one data record.

The wrapper’s efficiency is measured based on three factors, the number of actual data records to be extracted, the number of extracted data records from the test web pages, and the number of correct data records extracted from such test pages. Based on these three values, precision and recall are calculated according to the formulae:

\[
\text{Recall} = \frac{\text{Correct}}{\text{Actual}} \times 100
\]

\[
\text{Precision} = \frac{\text{Correct}}{\text{Extracted}} \times 100
\]

Our wrapper took around 370 milliseconds on average to generate a result for a search engine result page when we computed with data set. The algorithms proposed in ViNT and DEPTA took about 400 milliseconds the process of date extraction. As our wrapper uses visual information, it will be able to distinguish correct data records from the incorrect ones. This approach works well on the complicated web pages containing several data regions and so the VWDRE is able to extract the data records efficiently.
5. CONCLUSION AND FUTURE WORK

Generally, the needed information is embedded in the search engine result pages in the form of data records provided by web databases when we place queries. While studying large number of web pages, we observed some common visual features that are helpful in data extraction. And we proposed a vision-based approach to extract structured data from search engine result pages, which can overcome the limitations of previous works.

Our approach consists of three primary steps: DOM tree building, data record extraction and VWDRE visual wrapper generation. The data record extraction process is carried out on the basis of visual features on the web page. The Visual wrapper VWDRE generation is to improve the efficiency of the data record extraction. The experimental results show that usage of visual information in the extraction process is very effective in data extraction algorithms.

However, there are still some remaining issues and we plan to address them in the future. The further refinement on the data records has to be done so as to segregate the individual data items. The efficiency of VWDRE has to be improved by using new APIs to obtain the visual information directly from the web pages.

REFERENCES

VWDRE – A Vision-Based Approach for Mining Data from Search Engine Result Pages


