Automatic Extraction of Data Record from Web Page based on Visual Features

Nwe Nwe Hlaing, Thi Thi Soe Nyunt
University of Computer Studies, Yangon
nwe2hlaing@gmail.com,ttsoenyunt@gmail.com

Abstract

The Web is increasingly becoming a very large information source. However, the information is visually structured such that it is easy for humans to recognize data records and presentation patterns, but not for computers. As web sites are getting more complicated, the construction of web information extraction system becomes more troublesome and time-consuming. Hence, tools for the mining of data regions, data records and data items need to be developed in order to provide value added services. Large number of techniques has been proposed to address this problem, but all of them have inherent limitations. In this paper, we propose an approach for automatic data record extraction method from web page, which we call Vision based Extraction of data Record (VER). The approach is based on the observation that visual similarity of the data record in web document. Firstly, we adopt VIPS (Vision-based Page Segmentation) algorithm to partition a web page into semantic blocks. Then, blocks are clustered by proposed block clustering method according to the appearance similarity. Among these clusters, we identify data region and finally extract data record from data region.

1. Introduction

Web information extraction is an important task for information integration. Multiple web pages may present same information using completely different formats or syntaxes, which makes integration of information a challenging task. Structure of current web pages is more complicated than ever and is far different from their layouts on web browsers. Due to the heterogeneity and lack of structure, automated discovery of targeted information becomes a complex task. A typical web page consists of many blocks or areas, e.g., main content areas, navigation areas, advertisements, etc. For a particular application, only part of the information is useful, and the rest are noises. Hence, it is useful to separate these areas automatically. Pages in data-intensive web sites are usually automatically generated from the back-end DBMS using scripts. Hence, the structured data on the web are often very important since they represent their host page's essential information, e.g., details about the list of products and services.

Semi-structured data records contained in the Web pages provide useful information for shopping agents and meta-search engines. Structured data objects are an important type of information on the web. Such objects are often data records retrieved from a backend database and displayed in web pages with some fixed templates. Extracting data from such data records enables one to integrate data from multiple sites to provide value-added services, e.g., comparative shopping, and meta-querying.

Recently, web information extraction has become more challenging due to the complexity and the diversity of web structures and representation. This is an expectable phenomenon since the Internet has been so popular and there are now many types of web contents, including text, videos, images, speeches, or flashes. The HTML structure of a web document has also become more complicated, making it harder to extract the target content by using the DOM (Document Object Model) tree only. Another trend is that web designers are adding more advanced graphical features to the web content to make it more appealing. Therefore it would be helpful for wrapper induction and information extraction if we could provide some visual clues about where the content to be extracted resides. Moreover, semantically similar objects are usually clustered together and resemble each other in the sense of human perception.

Given a web page that contains multiple data records (at least three), we discover data region from the underlying web page, extract the data record from them. In our approach, we do not require any prior knowledge of the target page and its content; or any domain specific assumption. In this paper, we study an automatic wrapper generator for extraction of data record from web page. We present a vision based extraction of data record (VER) method to extract data records from the web pages automatically. We focus on clustering web page segment block to improve information extraction task. The only input to the system is list page of same presentation and the goal of proposed system we will built is to be able to extract data record accurately.

The rest of this paper is organized as follows. An overview on the related work and background is described in Section 2. Section 3 describes the proposed system which presents VIPS and block clustering method for automatic data record extraction. Section 4 discusses data region identification. Finally conclude the paper in Section 5.

2. Related Work and Background

Information extraction from web pages is an active research area. The existing works in web data extraction can be classified according to their automation degree (for a survey, see [3]). There are several approaches [4], [6], [7], [8], [9] for structured data extraction, which is also called wrapper generation. The first approach [8] is to manually write an extraction program for each web site based on observed format patterns of the site. This manual approach is very labor intensive and time consuming. Hence, it does not scale to a large number of sites.

The second approach [9] is wrapper induction or wrapper learning, which is currently the main technique. Wrapper learning works as follows: The user first manually labels a set of training pages. A learning system then generates rules from the training pages. The resulting rules are then applied to extract target items from web pages. These methods either require prior syntactic knowledge or substantial manual efforts.

The third approach [4] is the automatic approach. The structured data objects on a web are normally database records retrieved from underlying web databases and displayed in web pages with some fixed templates. Automatic methods aim to find patterns/grammars from the web pages and then use them to extract data. Examples of automatic systems are IEPAD [4], ROADRUNNER [6], MDR [1], DEPTA [14] and VIPS [11]. Some of these systems make use of the Patricia (PAT) tree for discovering the record boundaries automatically and a pattern-based extraction rule to extract the web data. This method has a poor performance due to the various limitations of the PAT tree.

ROADRUNNER [6] extracts a template by analyzing a pair of web pages of the same class at a time. It uses one page to derive an initial template and then tries to match the second page with the template. Deriving of the initial template has to be again done manually, which is a major limitation of this approach.

Another problem with the existing automatic approaches is their assumption that the relevant information of a data record is contained in a contiguous segment of HTML code, which is not always true. MDR [1] basically exploits the regularities in the HTML tag structure directly. MDR works well only for table and form enwrapped records while our method does not have this limitation. MDR algorithm makes use of the HTML tag tree of the web page to extract data records from the page. However, an incorrect tag tree may be constructed due to the misuse of HTML tags, which in turn makes it impossible to extract data records correctly. DEPTA [14] uses visual information (locations on the screen at which the tags are rendered) to find data records. Rather than analyzing the HTML code, the visual information is utilized to infer the structural relationship among tags and to construct a tag tree. But this method of constructing a tag tree has the limitation that, the tag tree can be built correctly only as long as the browser is able to render the page correctly.

Another similar system is Vints[7]. Vints proposes an algorithm to find SRRs (search result records) from returned pages of the search engines. However, our method focuses on list pages of same presentation template. Although some aspects and pieces of web information extraction may be around in various techniques, the important of this paper focus on the some interesting visual clues of web page and appearance similarity of web pages' data record. Moreover our method exploits continuous and non continuous data record.

3. Proposed System

This section describes the proposed system to extract data record from web page. This paper proposes an automatic extraction of data record by using visual clues, which we called VER (Vision based Extraction of data Record). An overview of proposed system is shown in figure 1. Our approach applies the concept of visual feature in web page and semantic information. The system relies on an existing algorithm for page segmentation to analyze and partition a web page into a set of visual blocks, and then group related blocks by appearance similarity of data record. Finally extract data record from identified data region.

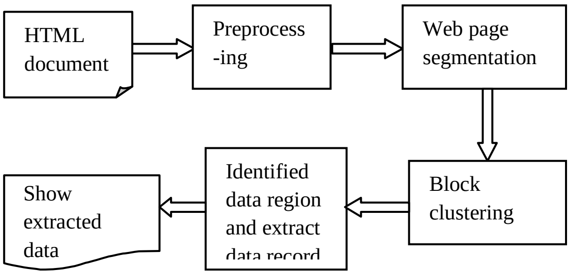


Figure 1. Overview of the System

The overall algorithm of the proposed system is as follows:

```
VER  Algorithm
Begin
1. Accept input web page.
2. Preprocessing the web page to filter the useless nodes.
3. Segment the web page into semantic blocks by using VIPS algorithm.
4. Cluster the block by using visual appearance similarity of web page.
5. Identified the data region and extract the data record from data region.
End.
```

3.1. Filtering Useless Nodes (Preprocessing)

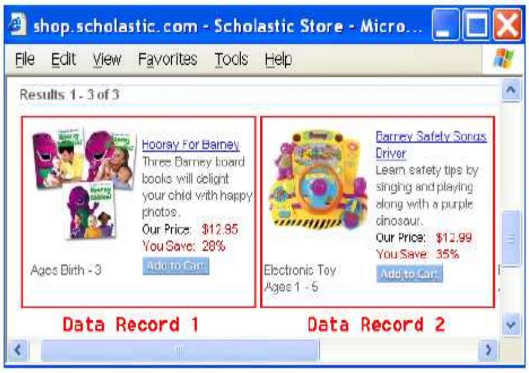
An HTML page contains tags as well as texts. Among all the tags, some of them are not of any interest for the whole wrapper generation process. An unimportant tags are <nobr>, <wbr>, <script>, <style>, <b>, <meta>, <noscript>, <input> etc.

Moreover, comment tags and &nbsp are also considered unnecessary for extraction purposes. First of all, we need to eliminate these nodes to get clean Html page for further processing.

3.2. Web Page Segmentation

3.2.1. Definition (Webpage Segmentation): Given a webpage, webpage segmentation is the task of partitioning the page at the semantic level and constructing a vision-tree for the page. Each node in the vision-tree will correspond to a block of coherent content in the original page.

Based on the definition, the output of webpage segmentation is the vision-tree of a webpage. Since vision-tree can effectively keep related content together while separating semantically different blocks from one another. Figure 3 is a vision-tree for the page in Figure 2, where rectangles denote the inner blocks and use ellipses to denote the leaf blocks (or elements).



Results 1 - 3 of 3

Hooray For Barney
Three Barney board books will delight your chid with happy photos.
Our Price: $12.95
You Save: 28%
Ages Birth - 3

Barney Safety Songs Driver
Learn safety tips by singing and playing along with a purple dinosaur.
Our Price: $12.99
You Save: 35%
Electronic Toy
Ages 1 - 5

Data Record 1

Data Record 2

Figure 2. A sample web page with two similar data record

For web data extraction, the first thing is to find a good representation format for web pages. Good representation can make the extraction task easier and improve extraction accuracy. In most previous work, tag-tree, which is a natural representation of the tag structure, is commonly used to represent a webpage. However, as Cai et al. (2004) pointed out, tag-trees tend to reveal presentation structure rather than content structure, and are often not accurate enough to discriminate different semantic portions in a webpage. A new method is based on the analysis of both the layouts and the semantic information of the web pages. Before extracting data region blocks, it is necessary to identify blocks occurring in a web page. VIPS (Vision-based Page Segmentation) algorithm excels in both an appropriate partition granularity and coherent semantic aggregation. VIPS can efficiently keep related content together while separating semantically different blocks from each other. Therefore, firstly we use VIPS algorithm for our data record extraction process.

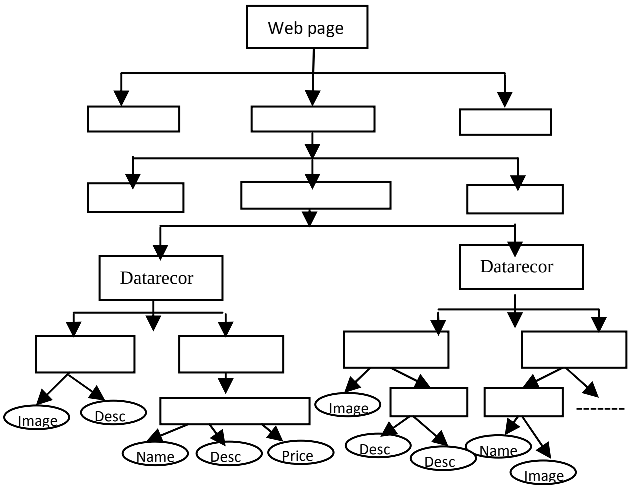


Figure 3. The Vision tree of the web page

3.2.2. Vision based Page Segmentation Algorithm

In VIPS (Vision-based Page Segmentation) algorithm which makes use of DOM structure as well as visual cues, to obtain the vision-based content structure for a web page. This algorithm consists of following 3 steps:

Step 1: Visual Block Extraction

In this phase, firstly find all appropriate visual blocks contained in the current sub tree. Normally, every node inside a current node can represent a visual block, like all the children of BODY. However, some “huge” nodes such as TABLE and P may act only for organization purpose and are not appropriate to represent a single visual block. Therefore, in these cases, this DOM node should be further divide the current node and replace it by its children. This process is iterated until all appropriate nodes are found to represent the visual blocks in the web page. At the end of this step, for each node that represents a visual block, its DoC value is set according to its intra visual difference. Some heuristic rules are used to determine whether a DOM node should be replaced as the following:

Tag cue: Tags such as HR are often used to separate different topics from visual perspective. Therefore a DOM node should be divided if it contains these tags.

Color cue: divide a DOM node if its background color is different from one of its children’s.

Text cue: If most of the children of a DOM nodes are Text nodes (i.e., no tags surround them), do not divide it. Size cue: prefer to divide a DOM node if the standard deviation of size of its children is larger than a threshold.

Step 2: Visual Separator Detection

When all blocks are extracted, they are put into a pool for separator detection. The algorithm defines visual separators as horizontal or vertical lines in a web page that visually cross with no blocks in the pool and then set appropriate weight to each separator according to the following patterns and select those with highest weight as the actual separators.

Distance pattern: The more the distance between blocks on different side of the separator, the higher the weight.

Tag pattern: If a visual separator is at the same position as some tags such as HR, its weight is made higher.

Font pattern: If differences of font properties such as font size and font weight are more clearly on two sides of the separator, the weight will be higher.

Color pattern: If background colors are different on two sides of the separator, the weight will be higher.

Step 3: Content Structure Construction

When separators are detected and separators’ weights are set, the content structure can be constructed accordingly. The construction process starts from the separators with the lowest weight and the blocks beside these separators are merged to form new virtual blocks. This process iterates till separators with maximum weights are met. The DoC of each new block is also set via similar methods. After that, each leaf node is checked whether it meets the granularity requirement. For every node that fails, go to the Visual Block Extraction phase again to further construct the sub content structure within that node. If all the nodes meet the requirement, the iterative process is then stopped and the vision-based content structure for the whole page is obtained. The common requirement for DoC is that DoC > PDoC, if PDoC is pre-defined.

3.3. Block Clustering

3.3.1. Visual Features of 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. As a result, visual features are important for identifying special information on web pages. Our data record extraction based on the following two observations.

Observation 1: Position Features (PFs).
These features indicate the location of the data region on a 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 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 web pages, first, data regions are always located in the centre section horizontally on 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.

Observation 2: 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).

Our web data extraction solution is developed mainly based on the above two observation. PFs is used to locate the region containing all the data records on a web page; AFs is used for clustering the block to discover the data records.

3.3.2. Block Clustering with Apperance Similarity

The blocks in the data region are clustered based on their appearance similarity. Since there are three kinds of information in data records, i.e., images, plain text and link text, the appearance similarity of blocks is computed from the three aspects. For images, we care about the size; for plain text and link text, we care about the shared fonts. Intuitively, if two blocks are more similar on image size, font, they should be more similar in appearance. The appearance similarity formula between two blocks b_1 and b_2 is given below:

$$Sim(b_1,b_2)=W_i*sim_{Img}(b_1,b_2)+ W_{pt}*sim_{PT}(b_1,b_2) +W_{lt} * sim_{LT}(b_1,b_2) \quad (1)$$

Where $sim_{Img}(b_1,b_2)$, $sim_{PT}(b_1,b_2)$, and $sim_{LT}(b_1,b_2)$ are the similarity based on image size , plain text font, and link text font, respectively. W_i, W_{pt}, and W_{lt} are the weights of these similarities respectively. Table 1 gives the formulas to compute the component similarities and the weights in different cases.

Table 1. The formulas of block appearance similarity and the weights in different cases

Formulas	Descriptions

$simImg(b1,b2) = \frac{Min\{sa_i(b_1),sa_i}{Max\{sa_i(b_1),sa_i}$	$sa_i(b)$is 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. $fn_{lt}(b)$ is the total number of fonts of the link texts in block b. $sa_{lt}(b)$ is the total area of the link text in block b.
$W_i = \frac{sa_i(b_1)+sa_i(b_2)}{sa_b(b_1)+sa_b(b_2)}$	
$simPT(b1,b2) = \frac{Min\{fn_{pt}(b_1),fn_p}{Max\{fn_{pt}(b_1),fn_p}$	
$W_{pt} = \frac{sa_{pt}(b_1)+sa_{pt}(b_2)}{sa_b(b_1)+sa_b(b_2)}$	
$simLT(b1,b2) = \frac{Min\{fn_{lt}(b_1),fn_{lt}}{Max\{fn_{lt}(b_1),fn_{lt}}$	
$W_{lt} = \frac{sa_{lt}(b_1)+sa_{lt}(b_2)}{sa_b(b_1)+sa_b(b_2)}$	

Our block clustering method consists of two steps: The first one is to build clusters by computing the similarity among blocks. The similarity $sim(b_1,b_2)$ between two blocks bi and bj is computed by the equation (1).The second one is to merge the resulting clusters. The threshold is trained from sample page. So the cluster building procedure is simplified as follows:

Procedure BlockClustering

```
Put all the blocks $b_i$ into the pool;
FOR(every block $b_i$ in pool){
compute the appearance similarity $sim(b_i,b_j)$ between two block
IF($sim(b_i,b_j)$ >threshold){
group $b_i$ and $b_j$ into a new cluster;
delete $b_i$ and $b_j$ from the pool; }
ELSE{
create a new cluster for $b_i$;
delete $b_i$ from the pool; }
}
```

The second step is to merge clusters. To determine if two clusters must be merged, we define the cluster similarity $simC_{kl}$ between two clusters C_k and C_l as the maximum value of $sim(b_i,b_j)$, for every two blocks $b_i \in C_k$ and $b_j \in C_l$.

Procedure BlockMerging

```
FOR(every cluster $C_k$)
{
compute the $simC_{kl}$ with other clusters;
IF($simC_{kl}$ >threshold){
clusters $C_k$ and $C_l$ are merged; }
}
```

4. Identifying Data Region

PF1 and PF2 indicate that the data records are the primary content on the web pages and the data region is centrally located on these pages. The data region corresponds to a block in the Visual Block tree (in this paper we only consider web pages that have only a single data region). We locate the data region by finding the block that satisfies the two PF features. Each feature can be considered a rule or a requirement. The first rule can be applied directly, while the second rule can be represented by $area_b/area_{page}$ >

threshold, where $area_b$ is the area of block b, $area_{page}$ is the area of the web page, and the threshold is used to judge whether b is sufficiently large relative to $area_{page}$. The threshold is trained from sample pages collected from different real web sites. For the blocks that satisfy both rules, we select the block as data region. Though very simple, this method can find the data region in the Visual Block tree accurately and efficiently.

5. Conclusion

The target pages from which the system wants to extract data object patterns are usually dynamic pages that are generated from a database. Therefore, the effort of information extraction can be viewed as a process of extracting the original database information that is transformed into structure data record. In this paper, we propose VER (Vision based Extraction of data Record) to solve the problem of record-level data extraction. First, we adopt VIPS algorithm to partition a web page into block. Then proposed block clustering method groups related blocks with their appearance similarity and define data region by observation 1 (PFs features). Finally, we extract each data records from the data region. Our approach is still implemented and we expect that the system will extract data record accurately and efficient for data integration.

References

[1] B. Liu, R. Grossman and Y. Zhai. Mining Data Records in Web Pages. ACM SIGKDD Conference, 2003.

[2] B. Liu and Y. Zhai. NET – A System for Extracting Web Data from Flat and Nested Data Records. WISE Conference,
2005.

[3] Cai, D., Yu, S., Wen, J.-R. and Ma, W.-Y., VIPS: a vision-based page segmentation algorithm, Microsoft Technical Report. MSR-TR-2003-79, 2003

[4] Chang, C-H., Lui, S-L. IEPAD: Information Extraction Based on Pattern Discovery. WWW-01, 2001.

[5] Chang, C.-H., Kayed, M., Girgis, M., and Shaalan, K. (2006). A survey of web information extraction systems. IEEE Transactions on Knowledge and Data Engineering, 18(10):1411–1428.

[6] Crescenzi, V. and Mecca, G. Automatic information extraction from large websites. Journal of the ACM, 2004, 51(5):731–779.

[7] H. Zhao, W. Meng, Z. Wu, V. Raghavan, C. Yu. Fully Automatic Wrapper Generation for Search Engines. WWW Conference, 2005.

[8] J. Hammer, H. Garcia Molina, J. Cho, and A. Crespo. Extracting semi-structured information from the web. In Proc.of the Workshop on the Management of Semi-structured Data, 1997.

[9] Kushmerick, N. Wrapper Induction: Efficiency and Expressiveness. Artificial Intelligence, 118:15-68, 2000. Clustering-based Approach to Integrating Source Query.

[10] M. Kayed, C.-H. Chang, FiVaTech: Page-Level Web Data Extraction from Template Pages, IEEE TKDE, vol. 22, no. 2, pp. 249-263, Feb. 2010.

[11] Shian-Hua Lin, Jan-Ming Ho, Discovering Informative Content Blocks from Web Documents, IEEE Transactions on Knowledge and Data Engineering, page 41-45, Jan, 2004

[12] Yang, Y. and Zhang, H. HTML page analysis based on visual cues. In Proceedings of the 6th International Conference on Document Analysis and Recognition, 2001, pages 859–864.

[13] YuJuan Cao, ZhenDong Niu, LiuLing Dai,YuMing Zhao, Extraction of Informative Blocks from web pages, in the Proceedings of International Conference on Advanced Language Processing and Web Information Technology, 2008

[14] Y. Zhai, B. Liu. Web Data Extraction Based on Partial Tree Alignment. WWW Conference, 2005