Discovering Informative Content Blocks for Efficient Web Data Extraction

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

Abstract

As web sites are getting more complicated, the construction of web information extraction systems becomes more troublesome and time-consuming. A common theme is the difficulty in locating the segments of a page in which the target information is contained, which we call the informative blocks. So discriminating informative blocks from the noisy blocks and then extracting the informative blocks from web page is an important task. In this paper, we propose a method that utilizes both the visual features and semantic information to extract information block. First, the VIPS (Vision-based Page Segmentation) algorithm is used to partition a web page into semantic blocks with a hierarchy structure. Then spatial features (such as position, size) and content feature (the number of image and links) are extracted to construct feature vector for each block. Secondly based on these feature, the blocks with similar content structures and spatial structures are clustered by means of similarity computation. After clustering blocks with similar structures, determine the cluster with the largest size and nearest distance to the centre of page as informative block.

Keywords:Vision-based Page Segmentation; Information Extraction; Block Clustering

1. Introduction

One of the key issues in web information extraction is to locate and extract target information correctly [3,11]. Most information extraction systems rely on machine learning techniques to build extraction rules. Supervised learning is advantageous in terms of correctness, but typically large volume of data is needed to train the system. Although unsupervised learning is very attractive because it does not require any training data, some results are in accurate [4,9,10]. Wrapper induction relies on semiautomatic supervised learning [4], i.e., it learns extraction rules from labeled web pages and data records and uses them to extract the relevant target information from new web pages with similar patterns as the training data. Current approaches to wrapper induction need to examine the whole pages, which might be problematic if the pages being examined have complex layouts or the induction algorithm is costly.

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 clues about where the content to be extracted resides.

Based on this motivation, this paper proposes a method to identify informative webpage blocks for efficient information extraction. An informative block is defined as a logical part of a web page that contains its core content. Extracting the informative blocks in a web page is not an easy task, chiefly for web pages that are built by using many graphical and visual features for human readability. The key idea of this paper is to achieve this goal and to explore the visual characteristics of the web page, not just relying on the HTML hierarchical structure.

The rest of this paper is organized as follows. Section 2 describes the related work. In section 3, the proposed system is described. Section 4 discusses the page segmentation and block clustering is discussed for section 5. Finally conclusion is presented in section 6.

2. Related Work

Most methods of automatic information extraction are based on tag information analyses, or content and link information which fail to take into account visual structure of the web page. However there have been a number of studies that analyze an

HTML page visually in order to extract target information from the pages.

Deng Cai et al. [2] proposed a Vision-based Page Segmentation Algorithm. VIPS uses visual representation of a web page with the DOM backend. Intuitively, it is clearly a good solution as people also see a web page not as a tree of HTML tags but in visual representation.

There are many methods which are based on Visual Page Segmentation Algorithm. Jinbeom Kang and Joongmin Choi [3] detect the informative blocks in a web page by exploiting the visual page segmentation algorithm to analyze and partition a web page into a set of logical blocks, and then group related blocks with similar structures into block cluster and recognizes the informative block clusters by applying some heuristic rules to the cluster information. However the function to compute the distance between clusters is unspecific and the threshold of identification of informative blocks need human involvement.

Similarly, Cao et al. [4] propose the EIBA (Extract Informative Block Arithmetic) algorithm that first partition a web page into semantic blocks using vision-based page segmentation. The visual and the semantic information got by LSI (Latent Semantic Indexing) are extracted to form the feature-vector of the block. Second they manually annotate informative or uninformative labels to the blocks. The labeled blocks are used as training dataset to train a classification model. Then the informative blocks can be extracted through the model.

Many machine learning and information retrieval technique for processing web pages do not utilize implicit visual information contained in the HTML source. By using the visual information, a new method is proposed to automatically extract informative block from a web page.

3. Proposed System

This section describes the proposed system to discover information block. This system applies the concept of visual feature in web page and semantic information. The system relies on an existing algorithm for vision-based page block segmentation to analyze and partition a web page into a set of visual blocks, and then group related blocks with similar content structures into block clusters by using cosine similarity and recognize informative block according to the size and distance to the centre of the page. The system includes following four steps:

Step 1: Segment the web page by using VIPS algorithm.

Step 2: Cluster the block by using method of similarity computation.

Step 3: Merging these block cluster to discover information block.

Step 4: Determine the cluster with the largest size proportion and the nearest distance to the center of the page as information block.

4. Web Page Segmentation

To segment a webpage into semantically coherent units, the visual presentation of the page contains a lot of useful cues. Generally, a webpage designer would organize the content of a webpage to make it easy for reading. Thus, semantically coherent content is usually grouped together and the entire page is divided into regions for different content using explicit or implicit visual separators such as lines, blank areas, images, font sizes, and colors. Our goal is to derive this content structure from the visual presentation of a webpage. The webpage segmentation problem is defined as below:

Definition 4.1 (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. Each node on this vision-tree represents a data region in the webpage, which is called a *block*. The root block represents the whole page. Each inner block is the aggregation of all its child blocks. All leaf blocks are atomic units (*i.e.*, elements) and form a flat segmentation of the webpage. Since vision-tree can effectively keep related content together while separating semantically different blocks from one another. Figure 2 is a vision-tree for the page in Figure 1, where rectangles denote the inner blocks and use ellipses to denote the leaf blocks (or elements). Due to space limitations, the blocks denoted by dotted rectangles are not fully expanded.

Figure 1. A sample web page with two similar data record

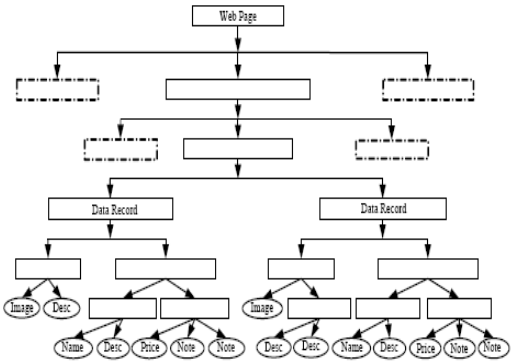

Figure 2. The vision tree of the page in figure 1.

A new method is based on the analysis of both the layouts and the semantic information of the web pages. Before extracting informative 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. Visual cues such as font, color and size, are used to detect blocks. By detecting useful visual cues based on DOM structure, a tree-like vision-based content structure of a web page is obtained. The granularity is controlled by a degree of coherence (DoC) which indicates how coherent each block is. The stopping of the VIPS algorithm is controlled by a predefined DoC (PDoC), which plays a role as a threshold to indicate the finest granularity.

The segmentation only stops when the DoCs of all blocks are no smaller than the PDoC.

5. Block Clustering

Although visually separated blocks provide a semantic partitioning of a page, a block might be too small to be considered as the source for information extraction, especially when the PDoC value of VIPS is set to a large value. In our method, the blocks with similar content structures and spatial structures are clustered by using methods of similarity computation. Since we are dealing with HTML pages and their segmented visual blocks which are represented by feature vectors, we apply equation (1) to measure the similarities among blocks.

VIPS utilizes useful visual cues to obtain a better partition of a page and then spatial features and semantic features are extracted to construct a feature vector for each block. In this paper, we extract 12 features to represent a block and then propose a new method to divide these features: quantifiable features and textual features (or unquantifiable features). The spatial features are called absolute spatial features since they directly use the absolute values of the four features. But using absolute values may make it hard to compare the features between different web pages. For example, a big block in a small page will always be taken as small block when it's compared with the blocks in a big page. In [8], they normalize the absolute features with the width and height of the whole page, and transform them into relative spatial features:

{ *bwidth/ pagewidth, bheight/ pageheight, center_x /pagewidth, center_y /pageheight*}.

As above, we transform the contents features into relative contents features. The relative block features list is given below in Table 1.

Table 1 Relative Block Features

Class	Division	Feature name	Definiton	Description
quantifiable features	spatial features	*rbwidth*	*bwidth/ pagewidth*	*pagewidth* is the width of the whole page, *pageheight* is the height of the whole page, *pagesize* is the size of the whole page.
		rbheight	bheight/ pageheight	
		rcenter_x	center_x /pagewidth	
		rcenter_y	center_y/pageheight	
		rsize	size/pagesize	
	content features	*rtextlen*	*textlen/pagetextlen*	*rtextlen* is the length of whole page. *pageimgnum* is the number of images contained in the page. *pagelinknum* is the number of links contained in the page.
		rimgnum	*imgnum/pageimgnum*	
		rimgsize	*imgsize/pagesize*	
		rlinknum	*linknum/pagelinknum*	
		rlinktextlen	*linktextlen/rtextlen*	
textual features	semantic features	*text*		text content of a block
		linktext		Anchor text context of a block

Let B_x and B_y be two visual blocks with its corresponding feature vectors V_x and V_y. The similarity sim_{xy} between B_x and B_y is computed as follows:

$$Sim_{xy}=w1*sim_{1xy}+w2*sim_{2xy} \quad (1)$$

Where sim_{1xy} , sim_{2xy} is the similarity of quantifiable features and textual features between block B_x and B_y respectively and $w1$ is the weight of quantifiable features and $w2$, textual features weight. sim_{1xy}, the similarity of quantifiable features between block B_x and B_y is the similarity between vector V_x and V_y. We compute the similarity between two vectors by cosine formula as given below:

$$cosine(V_x:V_y) = \frac{\sum_{i=1}^{n} V_{xi} \times V_{yi}}{\sqrt{\sum_{i=1}^{n} V_{xi}^2} \sqrt{\sum_{i=1}^{n} V_{yi}^2}} \quad (2)$$

where n is the dimension of vector.

The similarity of text between block Bx and By *(sim2xy)* is computed by equation (2). The textual features, in other words, string values contain the core information that we are interested in and needs to be extracted. Thus it is important for the clustering algorithm to exploit the similarity of string values to be able to group similar text tokens together. VSM (Vector Space Model) is used to construct the similarity measure for the textual features.

VSM represents natural language documents in a formal manner by the use of vectors in a multi-dimensional space. In particular, each document in the vector space model is represented as a vector contains a list of feature terms and its weights; the standard TF-IDF method is used to represent each document into a vector. Given two textual features represented by two vectors TV_x and TV_y, the similarity between TV_x and TV_y is cosine (TV_x: TV_y).

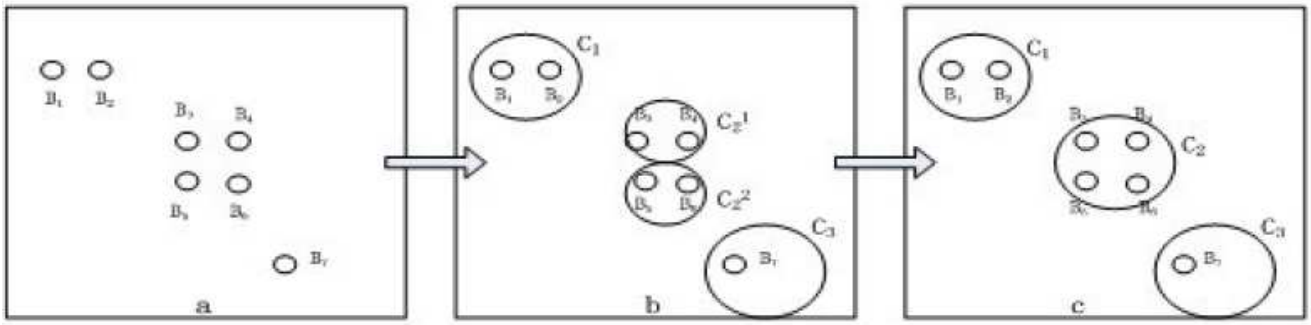

Figure 3. Block Clustering

Our block clustering method consists of two steps: the first one is to build clusters by computing the similarity among blocks, and eventually map each block onto a cluster that contains its nearest block; the second one is to merge the resulting clusters. Fig.3 shows an example of block clustering, assuming that the blocks are positioned in the space according to their relative distances. The similarity sim_{xy} between two blocks B_x and B_y is computed by the equation 1. In our method, the threshold is set to the median value of all the block distances, denoted by Median. Therefore the cluster building procedure is simplified as follows:

Procedure BlockCluster
Put all the blocks B_x into the pool;
FOR(every block B_x in pool){
compute the sim_{xy} with other blocks and find the nearest block B_y;
IF(sim_{xy} < Median){
group B_x and B_y into a new cluster;
delete B_x and B_y from the pool;
}
ELSE{
create a new cluster for B_x ;
delete B_x from the pool;
}
}

The second step is to merge clusters. To determine if two clusters must be merged, we define the cluster distance $csim_{kl}$ between two clusters C_k and C_l as the maximum value of sim_{xy}, for every two blocks B_i C_k and B_j C_l. The cluster merging procedure is defined as given below:

Procedure BlockMerging
FOR(every cluster C_k)
{
compute the $csim_{kl}$ with other clusters;
IF($csim_{kl}$ < Median){
clusters C_k and C_l are merged;
}
}

For every cluster, we can get the spatial features and content features. Take C_2 for example, the *size* of the cluster is the sum of all blocks' size in the cluster. And the *center_x* and *center_y* of C_2 is recalculated by coordinates of all the contained block centers.

After clustering blocks with similar structures, determines informative clusters. Typically, web authors would put the most important information in the center and the banner bar on the header, the navigation bar on the left or the right side and the copyright on the footer. And the area covered by a rectangle that bounds the informative blocks is more than the area covered by rectangles bounding others. Thus, the importance of a block can be reflected by spatial features such as position, size, etc. Based on this observation, we then determine the cluster with the largest size proportion and the nearest distance to the center of the page, which is target informative cluster. The blocks contained in the cluster are informative blocks and the information such as text, links, and images that contained in the blocks is meaningful information which is important to the classification of web pages.

6. Conclusion

This paper proposes a method to discover the informative blocks in a web page for efficient information extraction. This paper is composed of four steps: visual page block segmentation, block clustering, and merging these cluster and discovering informative block. Regarding block segmentation, it relies on the VIPS algorithm to analyze and partition a web page into a set of visually-separated blocks; regarding block clustering, it groups related blocks with content structures and spatial structures into a block cluster by using block cluster algorithm, and merge these block cluster according to the similarity of cluster and regarding discovering informative block, it relies on measuring the size of the block and the distance to the centre of the page.

7. References

[1] 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

[2] Cai, D., Yu, S., Wen, J.-R. and Ma, W.-Y. (2003). Extracting Content Structure for Web Pages based on Visual Representation, *Asia Pacific Web Conference (APWeb 2003), pp. 406417.*

[3] 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.

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

[5] Jinbeom Kang, Joongmin Choi, Detecting Informative Web Page Blocks for Efficient Information Extraction Using Visual Block Segmentation, in the proceeding of 2007 International Symposium on Information Technology Convergence (ISITC 2007), Jeonju, Korea, November, 2007

[6] Kovacevic, M., Diligenti, M., Gori, M. and Milutinovic, V., Recognition of Common Areas in a Web Page Using Visual Information: a possible application in a page classification, in the proceedings of 2002 IEEE International Conference on Data Mining (ICDM' 02), Maebashi City, Japan, December, 2002

[7] Nahm, U.Y., Bilenko, M. and Mooney R.J. Two Approaches to Handling Noisy Variation in Text Mining. ICML-2002 Workshop on Text Learning, 2002

[8] R. Song, H. Liu, J.R. Wen, and W.Y. Ma, Learning block importance models for web pages, 13th International World Wide Web Conference (WWW 2004), New York, USA, May, 2004

[9] 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

[10] Wong, T.-L. and Lam, W. (2007). Adapting web information extraction knowledge via mining site-invariant and site-dependent features. CM Transactions on Internet Technology.

[11] 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.

[12] 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