



VISION BASED PAGE SEGMENTATION ALGORITHM (VIPS) FOR WEB MINING

Mohan Raj C S

Asst Professor, Dept of MCA, Al-Ameen Institute of Information Sciences, Bangalore, India.

E-mail:mhanrajcs@gmail.com

Abstract:

Now days, Web pages provide a large amount of structured data, which is required by many advanced applications. This data can be searched through their Web query interfaces. The retrieved information is also called 'deep or hidden data'. The deep data is enwrapped in Web pages in the form of data records. These special Web pages are generated dynamically and presented to users in the form of HTML documents along with other content. These web pages can be a virtual gold mine of information for business, if mined effectively. Automation Anywhere intelligently extracts information. Running on SMART Automation Technology, it can automatically login to websites, account for changes in the source website, extract that information and copy it to another application reliably in a format specified by you. The purpose of VIPS is to extract the semantic structure of a web page based on its visual presentation. Such a semantic structure corresponds to a tree structure. In this tree each node corresponds to a block. A value is assigned to each node. This value is called the Degree of Coherence. This value is assigned to indicate the coherent content in the block based on visual perception. The VIPS algorithm first extracts all the suitable blocks from the HTML DOM tree. After that it finds the separators between these blocks. The separators refer to the horizontal or vertical lines in a web page that visually cross with no blocks. The semantics of the web page is constructed on the basis of these blocks.

Keywords: DOM, Web mining, Web data extraction

INTRODUCTION

Web mining aims to discover useful information or knowledge from the Web hyperlink structure, page content, and usage data. Although Web mining uses many data mining techniques, as mentioned above it is not purely an application of traditional data mining due to the heterogeneity and semi-structured or unstructured nature of the Web data. Many new mining tasks and algorithms were invented in the past decade. Based on the primary kinds of data used in the mining process, web mining consists of: web usage mining, web structure mining and web content mining. Mining extraction and integration of useful data, information and knowledge from web page content is called web content mining. Web Content Mining uses the ideas and principles of data mining and knowledge discovery to print more specific data. Web Data Extraction systems [1] are software applications for the purpose of extracting information from Web sources like Web pages. A Web Data Extraction system usually

interacts with a Web source and extracts data stored in it and converts the extracted data in the most convenient structured format and stores it for further usage. World Wide Web (WWW), as the largest database, often contains various data that we would like to consume for our needs. This data can be searched through Web query interfaces. The retrieved information (query results) is also called as deep data or hidden data[2]. This information is, in most cases, mixed together with formatting code and other information like website title, advertisement and navigation links, headers, footers, scripts and comments - which makes the page more human-friendly, but not machine-friendly. This deep data is enwrapped in webpages in the form of data records. These special Web pages, which contains these data records are generated dynamically and presented to users in the form of HTML documents along with other content. These web pages can be a virtual gold mine of information for business, if mined effectively. Web Wrapper [2] is a program that extracts content of HTML pages, and translates it into a relational form.

CHALLENGES IN WEB MINING

The web poses great challenges for resource and knowledge discovery based on the following observations

- **The web is too huge** – the size of the web is very huge and rapidly increasing. This seems that the web is too huge for data warehousing and data mining.
- **Complexity of Web pages** – the web pages do not have unifying structure. They are very complex as compared to traditional text document. There are huge amounts of documents in digital library of web. These libraries are not arranged according to any particular sorted order.
- **Web is dynamic information source** – the information on the web is rapidly updated. The data such as news, stock markets, weather, sports, shopping, etc., are regularly updated [3].
- **Diversity of user communities** – the user community on the web is rapidly expanding. These users have different backgrounds, interests, and usage purposes. There are more than 100 million workstations that are connected to the Internet and still rapidly increasing.
- **Relevancy of Information** – It is considered that a particular person is generally interested in only small portion of the web, while the rest of the portion of the web contains the information that is not relevant to the user and may swamp desired results.

RELATED WORK

A number of approaches have been reported in the literature for extracting information from Web pages. We briefly review earlier 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.

MANUAL APPROACHES

Some of the best known tools that adopt manual approaches are Minerva, TSIMMIS, and Web-OQL [1]. Obviously, they have low efficiency and are not scalable.

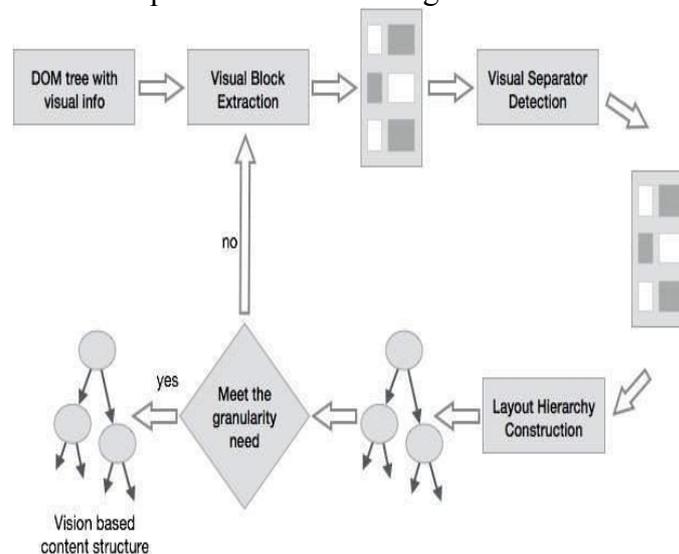
AUTOMATIC APPROACHES

In order to improve the efficiency and reduce manual efforts, most recent researches focus on automatic approaches instead of manual ones. Some representative automatic approaches are Omini [2], Roadrunner, IEPAD, MDR, DEPTA.

Vision-based page segmentation (VIPS)

- a) The purpose of VIPS is to extract the semantic structure of a web page based on its visual presentation.
- b) Such a semantic structure corresponds to a tree structure. In this tree each node corresponds to a block.
- c) A value is assigned to each node. This value is called the Degree of Coherence. This value is assigned to indicate the coherent content in the block based on visual perception.
- d) The VIPS algorithm first extracts all the suitable blocks from the HTML DOM tree. After that it finds the separators between these blocks.
- e) The separators refer to the horizontal or vertical lines in a web page that visually cross with no blocks.
- f) The semantics of the web page is constructed on the basis of these blocks.

The following figure shows the procedure of VIPS algorithm –



In this paper we are going to discuss about the VIPS algorithm. Vision based Page Segmentation algorithm (VIPS) extracts the blocks structure by using some visual cues and tag properties of the nodes. Unlike DOM based page segmentation, a visual block can contain DOM nodes from different branches in the DOM structure with different granularities. Structure tags such as <TABLE> and <P> can be divided appropriately with the help of visual information and wrong presentation of DOM structure can be reorganized to a proper form. Therefore, VIPS can achieve a better content structure for the original web page[4][5].

This VIPS algorithm is an automatic top-down; tag tree independent approach to detect web content structure. Basically, the vision-based content structure is obtained by using DOM structure. In this algorithm we follow three steps first one is block extraction, separator detection and content structure construction. These trees as a whole regarded as a round. The algorithm is top-down. The web page is firstly segmented into several big blocks and the hierarchical structure of this level is recorded. For each block, the segmentation process is carried out recursively until we get sufficient small blocks. The visual information of web pages, which has been introduced above, can be obtained through the programming interface provided by web browsers. In this paper, we employ the VIPS algorithm to transform a deep

web page into a visual block tree .A visual block tree is actually a segmentation of a web page. The root block represents the whole page, and each block in the tree corresponds to a rectangular region on the web pages. The leaf blocks are the blocks that cannot be segmented further, and they represent the minimum semantic units, such as continuous texts or images. These visual block tree is constructed by using DOM tree. DOM tree means document object model. Therefore these are all about the design part of the visual block tree and after that we will extract images, links and data.

VISUAL BLOCK EXTRACTION

In this step, we aim at finding all appropriate visual blocks contained in the current subpage. In general, every node in the DOM tree can represent a visual block. However, some “huge” nodes such as <TABLE> and <P> are used only for organization purpose and are not appropriate to represent a single visual block. In these cases, the current node should be further divided and replaced by its children. Due to the flexibility of HTML grammar, many web pages do not fully obey the W3C HTML specification, so the DOM tree cannot always reflect the true relationship of the different DOM node. For each extracted node that represents a visual block, its DoC value is set according to its intra visual difference. This process is iterated until all appropriate nodes are found to represent the visual blocks in the current subpage.

```
Algorithm DivideDomtree(pNode, nLevel)
{
  IF (Dividable(pNode, nLevel) == TRUE)
  {
    FOR EACH child OF pNode
    {
      DivideDomtree(child, nLevel);
    }
  } ELSE {
    Put the sub-tree (pNode) into the
    pool as a block;
  }
}
```

The visual block extraction algorithm

We judge if a DOM node can be divided based on following considerations:

- a) The properties of the DOM node itself. For example, the HTML tag of this node, the background color of this node, the size and shape of this block corresponding to this DOM node.
- b) The properties of the children of the DOM node. For example, the HTML tags of children nodes, background color of children and size of the children. The number of different kinds of children is also a consideration.

IMPLEMENTATION

In this section we are going to implement the DOM tree in order to find out the visual block tree **Fig 1.**(a) The presentation structure, (b) its visual block tree.

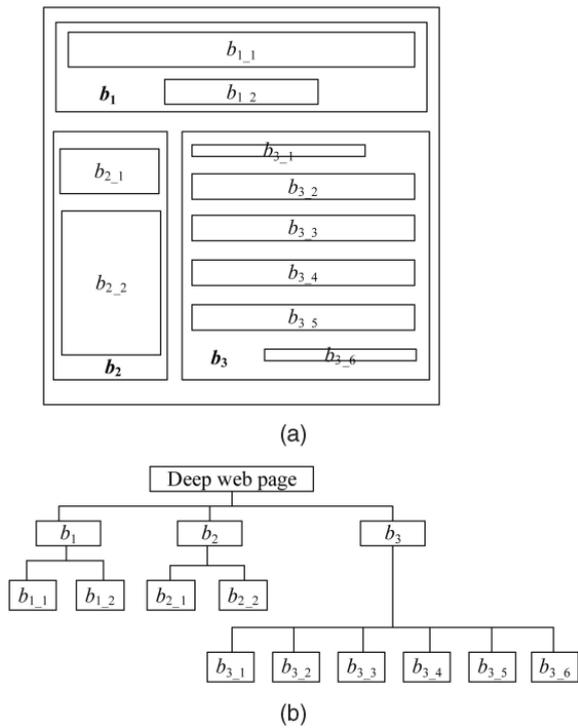


Fig 1 DOM structure

DOM TREE

In VIPS algorithm we will use DOM trees to find out the visual block tree. The Document Object Model (DOM) is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents. Aspects of the DOM (such as its "Elements") may be addressed and manipulated within the syntax of the programming language in use. The public interface of a DOM is specified in its application programming interface (API). The DOM is a programming API for documents. It is based on an object structure that closely resembles the structure of the document it models. For instance, consider this table, taken from an HTML document. In this we will take a sample html code and converted into a DOM [6][7]

```

<TABLE>
<TBODY>
<TR>
<TD>Shady Grove</TD>
<TD>Aeolian</TD>
</TR>
<TR>
<TD>Over the River, Charlie</TD>
<TD>Dorian</TD>
</TR>
</TBODY>
</TABLE>

```

A graphical representation of the DOM tree of the above html code is given as below.

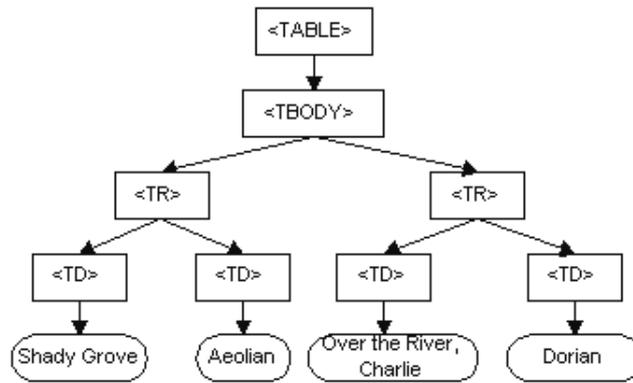


Fig2: Graphical representation of DOM

The DOM structure refers to a tree like structure where the HTML tag in the page corresponds to a node in the DOM tree. We can segment the web page by using predefined tags in HTML. The DOM is separated into 3 different parts levels:

- Core DOM - standard model for any structured document
- XML DOM - standard model for XML documents
- HTML DOM - standard model for HTML documents

The html DOM is a standard object model for any structured document, a standard interface for programming interface html, platform and language independent. The XML DOM views an XML document as a tree-structure. The tree structure is called a node-tree.

The XML DOM contains methods (functions) to traverse XML trees, access, insert, and delete nodes. However, before an XML document can be accessed and manipulated, it must be loaded into an XML DOM object. An XML parser reads XML, and converts it into an XML DOM object that can be accessed with JavaScript. Most browsers have a built-in XML parser. For security reasons, modern browsers do not allow access across domains. This means, that both the web page and the XML file it tries to load, must be located on the same server[8].

A web browser typically reads and renders HTML documents. This happens in two phases: the *parsing phase* and the *rendering phase*. During the parsing phase, the browser reads the markup in the document, breaks it down into components, and builds a document object model (DOM) tree. By using this VIPS algorithm we will separate the links, images and the data very easily and then we will extract that links, images and the data very easily.

EXPERIMENTS ON WEB INFORMATION RETRIEVAL

Query expansion is an efficient way to improve the performance of information retrieval [10]. The quality of expansion terms is heavily affected by the top-ranked documents. Noises and multi-topics are the two major negative factors for expansion term selection in the web context. Since our VIPS algorithm can group semantically related content into a block, the term correlations within a segment will be much higher than those in other parts of a web page. With improved term correlations, high-quality expansion terms can be extracted from segments and used to improve information retrieval performance[10].

We choose Okapi [11] as the retrieval system and WT10g [9] in TREC-9 and TREC 2001 Web Tracks as the dataset. WT10g contains 1.69 million pages and amounts to about 10G. The 50 queries from TREC 2001 Web Track are used as the query set and only the TOPIC field for retrieval, and use Okapi's BM2500 [12] as the weight function and set $k_1= 1.2$, $k_3= 1000$, $b= 0.75$, and $avdl= 61200$. The baseline is 16.55% in our experiments. An initial list of ranked

web pages is obtained by using any traditional information retrieval methods. Then we apply different page segmentation algorithms (including our VIPS algorithm with PDoC(6) and a naïve DOM-based approach) to the top 80 pages and get the set of candidate segments. The most relevant (e.g. top 20) segments from this candidate set are used to select expansion terms. These selected terms are used to construct a new expanded query to retrieve the final results. We compared our method with the traditional pseudo-relevance feedback algorithm using whole document and a naïve segmentation method based on DOM tree, which are briefly described below[9]:

- Our Vision-based approach (denoted as VIPS): The PDoC is set to 6. To reduce the effect of tiny blocks, blocks less than 10 words are removed. The top 80 pages returned by the Initial retrieval phase are segmented to form the candidate segment set.
- Simple DOM-based approach (denoted as DOMPS): We iterate the DOM tree for some structural tags such as TITLE, P, TABLE, UL and H1~H6. If there are no more structural tags within the current structural tag, a block is constructed and identified by this tag. Free text between two tags is also treated as a special block. Similar to VIPS, tiny blocks less than 10 words are also removed, and the candidate segments are chosen from the top 80 pages returned by the initial retrieval phase.
- Traditional full document approach (denoted as FULLDOC): The traditional pseudo-relevance feedback based on the whole web page is implemented for a comparison purpose. The experimental result is shown in Table 1 and Fig 3.

Table 1. Performance comparison of query expansion using different page segmentation methods

Number of Segments	Baseline (%)	FULLDOC (%)	DOMPS (%)	VIPS (%)
3	16.55	17.56	17.94	18.01
		(+6.10)	(+8.40)	(+8.82)
5		17.46	18.15	19.39
		(+5.50)	(+9.67)	(+17.16)
10		19.10	18.05	19.92
		(+15.41)	(+9.06)	(+20.36)
20		17.89	19.24	20.98
		(+8.10)	(+16.25)	(+26.77)
30		17.40	19.32	19.68
		(+5.14)	(+16.74)	(+18.91)
40	15.50	19.57	17.24	
	(-6.34)	(+18.25)	(+4.17)	
50	13.82	19.67	16.63	
	(-16.50)	(+18.85)	(+0.48)	
60	14.40	18.58	16.37	
	(-12.99)	(+12.27)	(-1.09)	

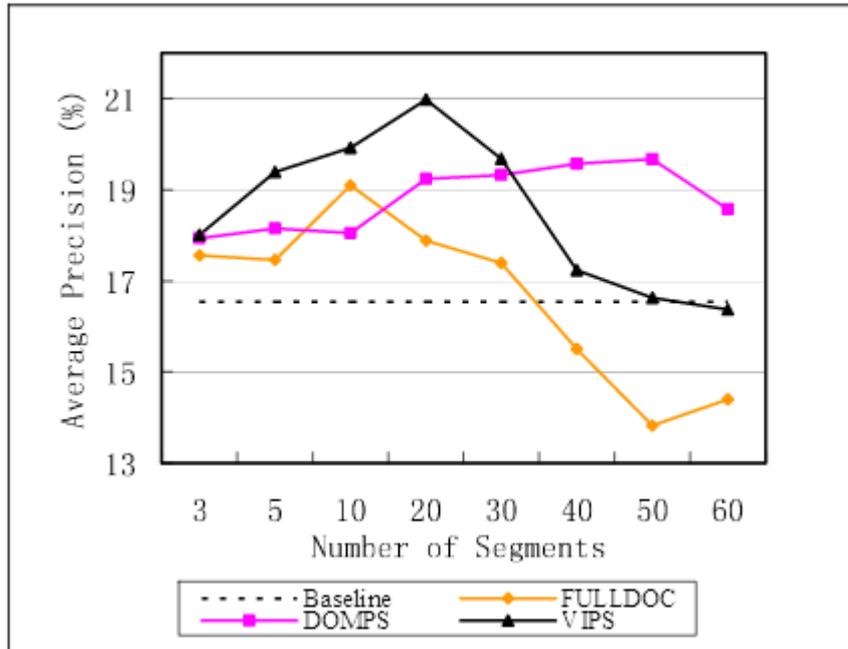


Fig3. Performance comparison of pseudo-relevance feedback based on three different ways of selecting query expansion terms.

As can be seen, the average retrieval precision can be improved after partitioning pages into blocks, no matter which segmentation algorithm is used. In the case of FULLDOC, the aximal average precision is 19.10% when the top 10 documents are used to expand the query. DOMPS obtains 19.67% when the top 50 blocks are used, a little better than FULLDOC. VIPS gets the best result 20.98% when the top 20 blocks are used and achieves 26.77% improvement.

CONCLUSION

In this paper we have proposed the concept of extracting data easily from the web page using VIPS algorithm. Earlier they had used web page programming language dependent that is very difficult to analyze the data because of complicated html and xml structures. So we will extract the data easily by using this VIPS algorithm.

REFERENCES

- [1] Ashish, N. and Knoblock. C. A., ‘Semi-Automatic Wrapper Generation for Internet Information Sources’. *Proceedings of the Conference on Cooperative Information Systems*, 1997, pp. 160-169.
- [2] Bar-Yossef, Z. and Rajagopalan, S.. ‘Template Detection via Data Mining and its Applications.’ *Proceedings of the 11th International World Wide Web Conference (WWW2002)*, 2002.

- [3] Adelberg, B. 'NoDoSE: A Tool for Semi-Automatically Extracting Structured and Semi-Structured Data from Text Documents.' *Proceedings of ACM SIGMOD Conference on Management of Data*, 1998, pp. 283-294.
- [4] G.O. Arocena and A.O. Mendelzon. "WebOQL: Restructuring Documents, Databases, and Webs." *Proc. Int'l Conf. Data Eng.(ICDE)*, pp. 24-33, 1998.
- [5] www.w3schools.com
- [6] <http://arxiv.org/pdf/1207.0246.pdf> (Accessed: May, 2013).
- [7] C. H. Chang, M. Kayed, M. R. Girgis, and K. Shaalan, "A Survey of Web Information Extraction Systems." *IEEE Transactions on Knowledge and Data Engineering*, 2006, pp. 1411-1428.
- [8] Hua Wang, Yang Zhang, "Web Data Extraction Based on Simple Tree Matching," *IEEE*, 2010, pp. 15-18.
- [9]. Bailey, P., Craswell, N. and Hawking, D. *Engineering a Multi-Purpose Test Collection for Web Retrieval Experiments, Information Processing and Management*, 2001.
- [10]. Buckley, C., Salton, G., and Allan, J., "Automatic Retrieval with Locality Information Using Smart". *The First Text Retrieval Conference (TREC-1)*, National Institute of Standards and Technology, Gaithersburg, MD, 1992, pp. 59-72.
- [11]. Robertson, S. E. "Overview of the Okapi Projects." *Journal of Documentation*, Vol. 53, No. 1, 1997, pp. 3-7.