Efficient Approach for Webpage Segmentation

1Sekhar Babu Boddu 2Sarada. Vattikuti 3Swathi Mogudumpuram
Asst Professor, Dept Of MCA Student, Dept Of MCA Student, Dept Of MCA
1,2,3 K L University, Guntur, Andhra Pradesh, India.

Abstract

With the enormous growing amount of information available on the Internet, an effective technique for users to apprehend the useful information from the unnecessary information is urgently required. Cleaning Web pages before mining becomes critical for improving performance of information retrieval and information extraction. So, we propose to remove various noisy data patterns in Web pages instead of extracting relevant content from Web pages to get main content information. Cleaning Web pages before mining includes Web page segmentation. Content mining is the scanning and mining of text, pictures and graphs of a Web page to determine the relevance of the content to the search query. Cleaning Web pages before mining includes Web page segmentation. In this, the webpage segmentation is done by using VIPS algorithm (Vision Based Page Segmentation). We see that Page segmentation is effective when applying VIPS algorithm rather than DOM (Document Object Model). We have implemented our method on several commercial Web sites and News Websites to evaluate the performance and improvement of our approach. Experimental results show the effectiveness of the approach. In this paper we show that VIPS is more efficient than DOM.

Keywords

DOM, VIPS, Page Segmentation

1. Introduction

Web page contains an enormous amount of information, which is having blocks, sections etc. These blocks may contain navigations, main content, ads etc. Being able to identify a block within a web page that represents the primary topic of that page may help a search engine decide which words are the most important ones on the page when it tries to associate the page with keywords that someone might search with to find that page. These blocks are very useful at the time of indexing the Webpages. When a person wants to search for a particular item, the keyword, by search engines are matched with the blocks which may specify the main content information. By dividing the page into blocks search engine can easily relate to, what is the topic that is most relevant. By dividing the page the weight or priority for a particular web page link will have more value, it is also helpful for eliminating noisy data. Pages that appear in search results where the query terms searched for are related to the primary content of those pages are likely to provide a much better experience for a searcher than pages appearing in those search results where noise on the page is related to the query searched for by someone. Today the Web has become the largest information source for people. Most information retrieval systems on the Web regard web pages as the smallest and undividable units, but a web page as a whole may not be appropriate to represent a single topic. A web page usually contains various contents such as navigation, decoration, interaction and contact information, which are not related to the topic of the web-page. Furthermore, a web page often contains multiple topics that are not necessarily relevant to one another. Therefore, detecting the semantic content structure of a web page could potentially improve the performance of web information retrieval.

2. DOM (Document Object Model)

The Document Object Model (DOM) is an application programming interface (API) for valid HTML and well-formed XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated. In the DOM specification, the term "document" is used in the broad sense - increasingly, XML is being used as a way of representing many different kinds of information that may be stored in diverse systems, and much of this would traditionally be seen as data rather than as documents. Nevertheless, XML presents this data as documents, and the DOM may be used to manage this data.

With the Document Object Model, programmers can build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML or XML document can be accessed, changed, deleted, or added using the Document Object Model, with a few exceptions - in particular, the DOM interfaces for the XML internal and external subsets have not yet been specified.

As a W3C specification, one important objective for the Document Object Model is to provide a standard programming interface that can be used in a wide variety of environments and applications. The DOM is designed to be used with any programming language. In order to provide a precise, language-independent specification of the DOM interfaces, we have chosen to define the specifications in Object Management Group (OMG) IDL [OMGIDL], as defined in the CORBA 2.3.1 specification [CORBA]. In addition to the OMG IDL specification, we provide bindings for Java [Java] and ECMAScript [ECMAScript] (an industry-
standard scripting language based on JavaScript and Script.

In the DOM, documents have a logical structure which is very much like a tree; to be more precise, which is like a "forest" or "grove", which can contain more than one tree. Each document contains zero or one doctype nodes, one root element node, and zero or more comments or processing instructions; the root element serves as the root of the element tree for the document. However, the DOM does not specify that documents must be implemented as a tree or a grove, nor does it specify how the relationships among objects be implemented.

The DOM is a logical model that may be implemented in any convenient manner. In this specification, we use the term structure model to describe the tree-like representation of a document. We also use the term "tree" when referring to the arrangement of those information items which can be reached by using "tree-walking" methods; (this does not include attributes). One important property of DOM structure models is structural isomorphism; if any two Document Object Model implementations are used to create a representation of the same document, they will create the same structure model, in accordance with the XML Information Set [Infoset].

The name "Document Object Model" was chosen because it is an "object model" in the traditional object oriented design sense: documents are modeled using objects, and the model encompasses not only the structure of a document, but also the behavior of a document and the objects of which it is composed. In other words, the nodes in the above diagram do not represent a data structure; they represent objects, which have functions and identity. As an object model, the DOM identifies:

- the interfaces and objects used to represent and manipulate a document
- the semantics of these interfaces and objects - including both behavior and attributes
- the relationships and collaborations among these interfaces and objects

The structure of SGML documents has traditionally been represented by an abstract data model, not by an object model. In an abstract data model, the model is centered around the data. In object oriented programming languages, the data itself is encapsulated in objects that hide the data, protecting it from direct external manipulation. The functions associated with these objects determine how the objects may be manipulated, and they are part of the object model.

3. Vision-based Content Structure for Web Pages

In this paper, we propose the vision-based content structure, where every node, called a block, is a basic object or a set of basic objects. It is important to note that, the nodes in the vision-based content structure do not necessarily correspond to the nodes in the DOM tree. Similar to the description of document representation in [Tang et al. 1999], the basic model of vision-based content structure for web pages is described as follows. A web page Ω is represented as a triple Ω = (O, Φ, δ). O = {Ω1, Ω2, ΩN} is a finite set of blocks. All these blocks are not overlapped. Each block can be recursively viewed as a sub-page associated with sub structure induced from the whole page structure. Φ = {φ1, φ

Figure 1: Graphical Representation of DOM
2, \( \Phi Tj \) is a finite set of separators, including horizontal separators and vertical separators. Every separator has a weight indicating its visibility, and all the separators in the same \( \Phi \) have the same weight. Is the expression of every two blocks in \( O \) and can be expressed as: \( \delta = O \times O \rightarrow \Phi \Omega \{ \text{NULL} \} \). For example, suppose \( \Omega \) and \( \Omega \) are two objects in \( O, ( , ) i j \delta \Omega \geq \text{NULL} \) indicates that \( \Omega \) and \( \Omega \) are exactly separated by the separator \( ( , ) i j \delta \Omega \) or we can say the two objects are adjacent to each other. Otherwise, there are other objects between the two blocks \( \Omega \) and \( \Omega \). Since each \( \Omega \) is a sub-web-page of the original page, it has similar content structure as \( \Omega \). Recursively, we have \( t ( t ( , , ) i j \delta \Theta O = O \delta \Theta, t \{ 1, 2, .... \text{Null} \} s \text{st} s \text{st} s \text{st} O = \Theta \text{st} O, t \{ 1, 2, .... \text{st}, \text{st} s \text{st} s \text{st} \} s \text{st} s \text{st} s \text{st} \Theta = \Phi \Theta \Phi \text{st} t ( t ( , , ) i j \delta \Theta O \rightarrow \text{st} O \text{st} NULL \) where \( t s \Theta \) is the \( t \)th object in the sub-web-page level \( s \), \( \text{st} s \text{st} s \text{st} O \) and \( \text{st} s \text{st} s \text{st} \Theta \) are the number of objects in \( t s O \) and number of separators in \( t s \Theta \).

In our algorithm, DoC values are integers ranging from 1 to 10, although alternatively different ranges (e.g., real numbers, etc.) could be used. We can pre-define the Permitted Degree of Coherence (PDoC) to achieve different granularities of content structure for different applications. The smaller the PDoC is, the coarser the content structure would be. For example in Figure 2(a), the visual block VB2_1 may not be further partitioned with an appropriate PDoC. Different application can use VIPS to segment web page to a different granularity with proper PDoC. The vision-based content structure is more likely to provide a semantic partitioning of the page. Every node of the structure is likely to convey certain semantics. For instance, in Figure 2(a) we can see that VB2_1_1 denotes the category links of Yahoo! Shopping auctions, and that VB2_2_1 and VB2_2_2 show details of the two different comics.

The VIPS Algorithm

5 Experiments

In this section, we will first show two example pages, comparing our VIPS result and the DOM tree structure. Then we provide some performance evaluation of our proposed VIPS algorithm based on a large collection of web pages from Yahoo. We also conduct experiments to evaluate how the algorithm can be used to enhance Web information retrieval.

5.1 Simple examples of web page segmentation

In this sub-section, the segmentation results of two web page are presented to give people an intuition how our VIPS algorithm works. In the meantime, we show the DOM tree of these pages. We can clearly find that we cannot get the right structure of the blocks only based on naïve DOM tree. Moreover, it is hard for us to decide which node conveys semantic meanings and where we should stop in different applications. Figure 3(a) shows our VIPS result on a sample page 1. The left part shows the page with different regions (different visual blocks in VIPS algorithms) marked with rectangles. The block marked with red rectangle is the block VB1-2-2-1. The upper right part shows the vision-based content structure of this page, while the lower one shows some statistics of selected visual block. From the right VIPS tree, we can know the hierarchical structure of different blocks. In different applications, we can control the partition granularity by setting PDoC, as shown in the northeast corner. The DoC value of each block is shown behind the node name (in parenthesis) in the right VIPS tree. Although from page

Figure 2: Example of vision-based content structure for a web page of Yahoo!
layout we see that VB1-2-1, VB1-2-2-1 and VB1-2-2-1 are parallel, in our heuristic rules the separator between VB1-2-1 and VB1-2-2-1 will get higher weight than the separator between VB1-2-2-1 and VB1-2-2-1 because of the different background colors. So VB1-2-2-1 and VB1-2-2-2 will be merged to VB1-2-2, paralleled to VB1-2-1.

For comparison, we show the DOM tree and its corresponding blocks in Figure 3(b). We can see that the area marked with red line is a <TR> node (with three <TD> children). From visual perspective, these <TD> nodes should not be grouped together, but we cannot get this information from DOM tree structure, while in VIPS this problem can be solved with the spatial and visual information of these blocks. We got the right content structure using our VIPS algorithm. Take another example web page1. We show the DOM tree structure and VIPS segmentation result in Figure 4.

In DOM tree structure, the images and the texts are belonging to different <TR> nodes. It is hard to decide the exact description text of the image. VIPS result clearly reveals the semantic relationship of the images and their surrounding texts. We can use these surrounding texts to represent the images and used this text representation in a web image search system.

From these examples, we can see that our VIPS algorithm can successfully identify the relationships among different blocks in the web page, while DOM structure fails. Moreover, VIPS algorithm assigns a DoC value to each node in vision-based content structure, which is critical in determining where to stop in different applications.

5.2 Performance of VIPS algorithm

To evaluate the performance of VIPS algorithm, we select 600 web pages from popular sites listed in 14 main categories of Yahoo! directory (http://www.yahoo.com). The VIPS algorithm is run on all the pages and the results are assessed by several individuals. Five volunteers are asked to judge the results. Table 1 shows the result:

<table>
<thead>
<tr>
<th>Human judgment</th>
<th>user1</th>
<th>user2</th>
<th>user3</th>
<th>user4</th>
<th>user5</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfect</td>
<td>299</td>
<td>307</td>
<td>374</td>
<td>310</td>
<td>377</td>
<td>1667</td>
</tr>
<tr>
<td>Satisfactory</td>
<td>231</td>
<td>266</td>
<td>193</td>
<td>237</td>
<td>1124</td>
<td></td>
</tr>
<tr>
<td>Fair</td>
<td>64</td>
<td>23</td>
<td>29</td>
<td>48</td>
<td>20</td>
<td>184</td>
</tr>
<tr>
<td>Bad</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 1: Evaluation of VIPS Algorithm

As can be seen, 1667+1124=2791 (93%) pages have their semantic content structures correctly detected. For those “fair” pages, the hierarchies are incorrectly constructed because they contain insufficient visual cues to separate blocks while people get the structure by understanding the semantic meaning of the blocks. For the “bad” cases, one major reason is that the browser (i.e. Internet Explorer in our experiments) provides wrong position information so that our algorithm cannot get the correct content structure. Another reason is that, for several pages the images (e.g., a very thin image representing a line) are used to divide different content blocks. Our algorithm currently cannot handle this situation.

5.3 Experiments on web information retrieval

Query expansion is an efficient way to improve the performance of information retrieval [Buckley et al. 1992] [Efthimiadis 1996]. The quality of expansion terms is heavily affected by the top-ranked documents. Noises and multitopics 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. We choose Okapi [Robertson 1997] as the retrieval system and WT10g [Bailey 2001] in TREC-9 and TREC 2001 in Web Tracks as the data set. 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 [Robertson et al. 1999] as the weight function and set $k_1 = 1.2$, $k_2 = 1000$, $b = 0.75$, 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:
• 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 2

6. Conclusions

In this paper a new approach for extracting web content structure based on visual representation was proposed. The produced web content structure is very helpful for applications such as web adaptation, information retrieval and information extraction. By identifying the logic relationship of web content based on visual layout information. In this paper, we proposed the vision-based content structure, where every node, called a block, is a basic object or a set of basic objects. It is important to note that, the nodes in the vision-based content structure do not necessarily correspond to the nodes in the DOM tree.

7. References:
Adelberg, B., NoDoSE: A tool for semi-
automatically extracting structured and semi-
structured data from text documents, In Proceedings
of ACM SIGMOD Conference on Management of

Ashish, N. and Knoblock, C. A., Semi-Automatic
Wrapper Generation for Internet Information
Sources, In Proceedings of the Conference on
Cooperative Information Systems, 1997, pp. 160-
169.

Ashish, N. and Knoblock, C. A., Wrapper
Generation for Semi-structured Internet Sources,

Bailey, P., Craswell, N., and Hawking, D.,
Engineering a multi-purpose test collection for Web
retrieval experiments, Information Processing and

Bar-Yossef, Z. and Rajagopalan, S., Template
Detection via Data Mining and its Applications,
In Proceedings of the 11th International World Wide

Bernard, M. L., Retrieval algorithms for topic distillation in a hyperlinked
environment, In Proceedings of the 21st ACM
International Conference on Research and Development in Information Retrieval (SIGIR98),

Buckley, C., Salton, G., and Allan, J., Automatic
Retrieval with Locality Information Using Smart, In the First Text Retrieval Conference (TREC-1),
National Institute of Standards and Technology,
Gaithersburg, MD, 1992, pp. 59-72.

Butler, D., Liu, L., and Pu, C., A Fully Automated
Object Extraction System for the World Wide Web,
In International Conference on Distributed

Buyukkokten, O., Garcia-Molina, H., and Paepcke,
A., Accordion Summarization for End-
GameBrowsing on PDAs and Cellular Phones, In Proceedings of the Conference on Human Factors in

Chakrabarti, S., Integrating the Document Object
Model with hyperlinks for enhanced topic
Distillation and information extraction, in the 10th

Chakrabarti, S., Joshi, M., and Tawde, V., Enhanced
topic distillation using text, mark-up tags, and
hyperlinks, In Proceedings of the 24th annual
international ACM SIGIR conference on Research
and development in information retrieval , ACM

Chakrabarti, S., Punera, K., and Subramanyam, M.,
Accelerated focused crawling through online
relevance feedback, In Proceedings of the eleventh international conference on World Wide Web

Q., Function-Based Object Model Towards Website
Adaptation, In Proceedings of the 10th International

Diao, Y., Lu, H., Chen, S., and Tian, Z., Toward
Learning Based Web Query Processing, In Proceeding of International Conference on Very
Large Databases, 2000, pp. 317-328.

Efthimiadis, N. E., Query Expansion, In Annual
Review of Information Systems and Technology,

Embley, D. W., Jiang, Y., and Ng, Y.-K., Record-
boundary discovery in Web documents, In Proceeding of the 1999 ACM SIGMOD
international conference on Management of data,

Hammer J., Garcia-Molina, H., Cho, J., Aranha, R.,
and Crespo, A., Extracting Semi-structured
Information from the Web, In Proceedings of the
Workshop on Management for Semi-structured
Data, 1997, pp. 18-25.

Kaasinen, E., Aaltonen, M., Kolari, J., Melakoski,
S., and Laakko, T., Two Approaches to bringing

Kleinberg, J., Authoritative sources in a hyperlinked
environment, In Proceedings of the 9th ACM-SIAM
Symposium on Discrete Algorithms, Baltimore,

Lin, S.-H. And Ho, J.-M., Discovering Informative
Content Blocks from Web Documents, In proceedings of ACM SIGKDD'02, 2002.

Page, L., Brin, S., Motwani, R. and Winograd, T.
The Page Rank citation ranking: Bringing order to
the web, Technical report, Stanford University,

Rahman, A., Alam, H., and Hartono, R., Content
Extraction from HTML Documents, In proceedings of the First International Workshop on Web

Robertson, S. E., Overview of the okapi projects,
3-7.

Robertson, S. E. and Walker, S., Okapi/Keenbow at
TREC-8, In the Eighth Text Retrieval Conference (TREC 8), 1999, pp. 151-162.

Tang, Y. Y., Cheriet, M., Liu, J., Said, J. N., and
Suen, C. Y., Document Analysis and

www.ijert.org
Recognition by Computers, Handbook of Pattern Recognition and Computer Vision, edited by C.
