www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242

Volume 4 Issue 6 June 2015, Page No. 12970-12973

Vision Assisted Web structure and Data Extraction Methods

Kumud Jaglan[1], Dr. Kulvinder Singh[2]

[1,2]University Institute of Engineering & Technology, Kurukshetra University, Kurukshetra, India

[1]kumudjaglan@yahoo.com, [2]kshanda@rediffmail.com

Abstract:

Web page segmentation has been done to address the problems in different fields including mobile web, archiving, phishing, etc. In this paper, different algorithms are summarized that web page segmentation addresses in different fields .Web page segmentation has myriad applications like information retrieval, page type classification etc. . This paper presents a survey of web page segmentation algorithms including DOM Tree, VIPS and SD Tree algorithms. VIPS approach is independent of underlying HTML representation and works well even when layout structure is different from the HTML structure. As there is difficulty in finding the meaningful blocks existing approaches presented can extract informative parts from web pages by creating meaningful blocks and segmenting noisy WebPages.

Keywords: ***Web page classification, Web page segmentation, Web Page, Web information extraction***

1. Introduction

Web documents can be viewed as complex objects which often contain multiple entities each of which can represent a standalone unit. However, most information processing applications developed for the web, consider web pages as the smallest undividable units. This fact is best illustrated by web information retrieval engines whose results are presented in the form of links to web documents rather than to the exact regions within these documents that directly match a user's query. A better retrieval performance can be achieved by considering the page not as an undividable unit but as having an underlying semantic structure with topically coherent segments. In WebPages detecting the semantic content structure will help in improving the web information retrieval. Web information extraction is an important task for information integration. Multiple web pages may present same information using completely different formats, 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 invention of targeted data becomes a convoluted task. A web page consists of countless blocks or spans, e.g., main content spans, exploration spans, advertisements, etc. For a particular application, mere portion of the data is functional, and the rest are noises. Hence, it is advantageous to distinct these spans automatically. 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 type of data records enable one to integrate data from multiple sites.

Web pages with some fixed templates. Extracting data from such data records enables one to integrate data from multiple sites.

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 various types of web contents. 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. This paper presents different methods based on semantic structure and evaluating type of web page as these provides a robust solution for information extraction and noise removal.

2. Related work

Many web applications uses web page semantic structure and contents. Hattori *et al.* [1] provided a segmentation method by calculating the distance between content elements of HTML tags structure. Different web page segmentation methods were introduced based on visual and non visual characteristics. In web information accessing, some researchers use database technique for building wrappers for the information extraction. For building wrappers, web documents are divided into parts. Diao *et al*. [2] provided segments of web page based on query processing using different HTML tags. Lin et al. [3] used only table tag for content blocks. Cai *et al*. [4] given an algorithm for extracting web based semantic structure. These semantic structures are in hierarchal nature which corresponds to a block. This algorithm made use of page layout feature. Alcic *et al.* [5] provided distance measures for content units on the basis of web page properties and DOM structure of web page. Nguyen *et al.* [6] proposed a method for segmenting a Web page into its semantic parts. There method segmented the page into blocks and then classifies the blocks. Bhardwaj, A. *et al.* [7], drafted that quick progress of the internet and web publishing methods craft countless data origins published as HTML pages on Web. Though, there was lot of redundant and irrelevant data additionally on web pages. Such data makes varied web excavating tasks such as web page scuttling, web page association, link established ranking, case distillation complex. This paper debated assorted ways for removing informative content from web pages and a new way for content extraction from web pages and density of links. Srivastava, S. *et al.*[8], presented that in the globe of data knowledge the adjustments happens rapidly. As the new technologies always adjusts the globe of data representation, the result was to find out relevant pieces of data cluttering alongside distracted features(like advertisements, links, scrollers etc.) in the finished web page. Data or functional content

extraction from the web pages (structured or semi structured) becomes a critical subject for web users and web miners. So the data extraction from the web page carries a huge importance. A mystifying mystery for data extraction is to depict the noisy area and its removal. They examine the DOM tree segmentation alongside class attribute established approach. The class attribute can be utilized alongside all HTML agents inside the `BODY' serving of the document. It was utilized to craft disparate classes of an agent, whereas every single class can have its own properties. To assess the arrangement presentation countless examinations completed on disparate business, news, and entertainment websites. Chee Sheen Chan *et al.* [9], proposed an ontology based web page segmentation algorithm for extracting web images with its associated contextual information according to its semantic characteristics like picture annotation, clustering of pictures, inference of picture semantic content and picture indexing.

Sanoja, A. *et al.* [10], depicted a web page segmentation framework. It wass a hybrid way inspired by automated document processing methods and visual-based content segmentation techniques. A web page was associated alongside three structures: the DOM tree, the content construction and the logical structure. The DOM tree embodies the HTML agents of a page, the content construction organizes page objects according to content's groups and geometry and in the end the logical construction is the consequence of mapping content construction on the basis of the human-perceptible meaning that conforms the blocks.

3. Webpage Segmentation Methods

There are different methods for vision based page segmentation and these are as:

1. SD-Tree
2. DOM Tree
3. VIPS

3.1 SD Algorithm

SD Algorithm identifies the type of pages (Comments, Multiple areas) and extracts their corresponding regions. This method used combination of non-visual & visual characteristics of a web page in order to achieve the page segmentation and omit noisy areas [11].

Some characteristics used in SD Tree algorithm are:

- **Density**

It represents the size of a node which is represented by the number of characters that are in a specific HTML node.

- **Distance from max density region**

It represents "farness" of a region is from another w.r.t. a density feature. It is calculated by the formula:

dfm(r)=100 - (dr ↤100)/d max

where dr :density of the examined region r

dmax: density of the region with the max density.

- **Distance from root**

How "far" a region is from the root node of the DOM tree.

- **Ancestor title**

It is the title detected in some of the ancestor nodes of a specific node like title is expressed by the <h1>, <h2>, <h3>, etc. HTML tags.

- **Ancestor title level**

It represents distance of ancestor title from a specific node w.r.t. the tree level.

Title nodediff = |anclevel title – nodelevel|

where anclevel title: level of an ancestor title &

nodelevel: level of the examined node.

- **Cardinality**

It is the number of child nodes.

- **Content of HTML nodes**

The content of the HTML nodes is the text that they contain.

SD Algorithm

1. Set dfr=T1 and dfm=T2. (T1 and T2 are threshold values)
2. Construct SD tree based on HTML content.
3. Calculate valid nodes (all the nodes that have density greater than T2 threshold) and remove noisy region from SD tree.
4. Merging all valid nodes into valid groups.
5. Calculate different features for valid groups like cardinality etc.
6. Calculate dfm and distances from max density region for all regions
7. All the regions that have dfm ≤ T1 are the candidate article regions .These regions are detected and grouped based on their CSS classes.
8. Final decision is made based on the candidate article regions detected

- Calculate the article region (region among the candidates, closer to the root node and with an ancestor title to the closest level).
- if article region found

  When the candidates >0 and an article was detected, further examination is made to the candidates in order to detect the page type.
- Scan the comment regions, the CSS classes and Id features of candidate regions are scanned for keywords that belong to the comment keywords specification.

Also check whether or not these regions have a common parent with Article region.

– if comment regions > 0 return Article
with Comments
– else if (all candidate regions in same level)
return Multiple
– else return Article
else return Multiple

3.2 DOM Tree

DOM (Document Object Model) is the application program interface (API) for HTML and XML document. Using the Document Object Model, programmers can build documents, add, modify or delete elements and content. DOM is a set of objects and access, interface dealing the document object. 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 modeled in such a way that it can be used with any programming language. In the DOM, documents have a logical structure which is very much like a tree. Each document contains 0 or 1 doctype nodes, one root node, and 0 or more comments or processing instructions; the root element serves as the root of the element tree for the document [12].

HTML document include the title, head, paragraph, hyperlinks and other various components. DOM parses the HTML file and generates the internal tree structure of the file, called the document's logical structure or logical structure tree. Tree structure can accurately describe the relative position of elements and it is suitable to describe semi structured data. DOM-based page segmentation is commonly based on the predefined syntactic structure, that is, HTML tags. HTML tags are not independent. There is a certain hierarchy relationship among them. Each tag effect on the resulting page is different: some only work on the visual features; some only affect the hierarchy; there's both of the two.

DOM-based page segmentation method to the simple structure page will have better results. However, the present structure of the popular website often complex and not the rule. This method is suitable for used in combination with other methods.

Structure of DOM-Tree

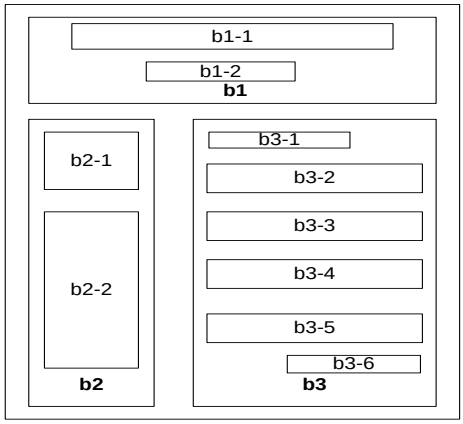

Fig 2: The Presentation Structure

DOM Tree Algorithm

```
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;
}
}
```

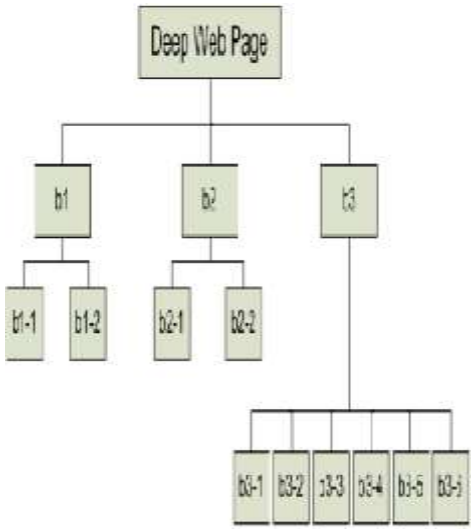

Fig 1: Its Visual Block Tree

3.3 VIPS

Vision based page segmentation algorithm is used to extract the semantic structure of a web page. Unlike DOM tree it is independent of HTML documentation representation.

A web page $\Omega = (O, \Phi, \delta)$. $O = \{\Phi_1, \Phi_2, \ldots, \Phi_N\}$ represents the finite set of blocks. Each block can be iteratively viewed as a sub-webpage related with sub-structure induced from the whole page structure. $\Phi = \{\Phi_1, \Phi_2, \ldots, \Phi_T\}$ is a finite set of separators, these separators are horizontal and vertical . Every separator has some weight which represents separator's visibility, and all the separators in the same block have same weight. δ represents relationship of every two blocks in O and can be expressed as: $\delta = O \times O \rightarrow \Phi \cup \{NULL\}$.

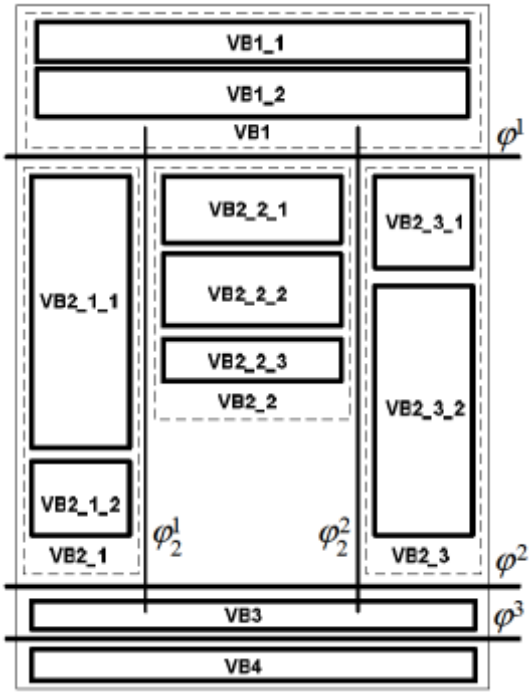

Fig 3: Layout structure and vision-based content structure of a web page [4]

The vision-based content structure of a page is obtained by joining the DOM structure and the visual cues. It includes three steps process: First block is extracted, then detecting the separators and third one is to construct content structure. These three steps as a whole are regarded as a encircling. The algorithm uses top-down tag approach. Web page is firstly segmented into several big blocks and the hierarchical structure is traced. For each big block, the same segmentation process is carried out recursively until we get sufficiently small blocks whose DoC (measure how coherent it is) values are greater than pre-defined PDoC (to achieve different granularity content structure for different application). For each round, the DOM tree with its visual information corresponded to the current block is obtained from a web browser. Then, from the root node(s) of the DOM tree the block extraction process is started to extract blocks from the DOM tree based on visual cues. Every DOM node is checked to judge whether it structures a single block or not. If its children are there then its children will be processed in the same way. A DoC value is assigned to each extracted block based on the block's visual property.

When all blocks are extracted, they are put into a group. Separators among these blocks are recognized and the weight of these separators is set on the basis of properties of neighboring blocks. The layout hierarchy is build on the basis of separators. After building the layout hierarchy of the current round, every leaf node of the content structure is checked to see whether it meets the granularity requirement or not. If not, this leaf node will be treated as a sub-page and will be further segmented likely.

4. Conclusion

Web Information Extraction has received a lot of attention by researchers over the years. However, most of the works are based on examining the HTML or the DOM tree of the Web pages. Web documents can be viewed as complex objects which often contain multiple entities each of which can represent a standalone unit. In this paper we presented existing approaches that can extract informative parts from web pages. The main challenge in most is the difficulty in finding the meaningful blocks that encompass the target data from the blocks that encompasses irrelevant data such

as advertisements, menus, or copyright statements. Although many algorithms can distinguish multiple topics in web pages, they do not consider the document length normalization problem and may other issues. In future we will propose an n-gram based web page segmentation algorithm that can be implemented for extraction of web segments without relying on the DOM tree for the segmentation process.

References

[1]. Hj G. Hattori, K. Hoashi, K. Matsumoto, and F. Sugaya. Robust web page segmentation for mobile terminal using content-distances and page layout information. In *WWW*, pages 361–370, 2007.

[2]. Y. Diao, H. Lu, S. Chen, and Z. Tian. Toward learning based web query processing. Proceedings of the 26th International Conference on Very Large Data Bases, pages 317–328, San Francisco, CA, USA, 2000.

[3]. S.-H. Lin and J.-M. Ho. Discovering informative Content blocks from web documents. Proceedings of the 8th international conference on Knowledge Discovery and data mining (SIGKDD), 2002.

[4]. D. Cai, S. Yu, and J.-r. Wen. VIPS: a Vision-based Page Segmentation Algorithm. Microsoft Technical Report (MSR-TR-2003-79), 2003.

[5]. S. Alcic and S. Conrad. Page segmentation by web content clustering. In Proceedings of the International Conference on Web Intelligence, Mining and Semantics, WIMS '11, pages 24:1–24:9, New York, NY, USA, 2011.

[6]. Nguyen, Cong Kinh, Laurence Likforman-Sulem, J-C. Moissinac, Claudie Faure, and Jérémy Lardon. "Web document analysis based on visual segmentation and page rendering." In Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on, pp. 354-358. IEEE, 2012.

[7]. Bhardwaj, Aanshi, and Veenu Mangat. "A novel approach for content extraction from web pages." Engineering and Computational Sciences (RAECS), 2014 Recent Advances in. IEEE, 2014

[8]. Srivastava, Shobhit, Mohd Haroon, and Abhishek Bajaj. "Web document information extraction using classes attribute approach." Computer and Communication Technology (ICCCT), 2013 4th International Conference on. IEEE, 2013.

[9]. Chan, Chee Sheen, Adel Johar, Jer Lang Hong, and Wei Wei Goh. "Ontological based webpage segmentation." In Fuzzy Systems and Knowledge Discovery (FSKD), 2013 10th International Conference on, pp. 883-887. IEEE, 2013.

[10]. Sanoja, Andres, and Stephane Gancarski. "Block-o-Matic: A web page segmentation framework." In Multimedia Computing and Systems (ICMCS), 2014 International Conference on, pp. 595-600. IEEE, 2014.

[11]. Nikolaos Pappas, Georgios Katsimpras, and Efstathios Stamatatos. "Extracting informative textual parts from web pages containing user-generated content." In Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies, p. 4. ACM, 2012.

[12]. Yaohui Li, Li Xia Wang, Jian Xiong Wang, Jie Yue, and Ming Zhan Zhao. "An approach of web page information extraction." Applied Mechanics and Materials347 (2013): 2479-2482.