International Journal of Hybrid Information Technology
Vol.8, No.3 (2015), pp.243-254
http://dx.doi.org/10.14257/ijhit.2015.8.3.22

An Algorithm on Web Article Automatic Extraction Based on DOM Structure

Weiguo Shen* and Xiaojian Zou**

** School of mechanical Engineering, Tianjin Polytechnic University, Tianjin, 300387, China*
*** Military Transportation University, Tianjin,300160, China*
Emails: shenweiguotianjin@163.com, ala_99@126.com

Abstract

By analyzing a large number of web pages, we proposed a page segment algorithm which is based on DOM tree structural features and visual features. The algorithm segments the pages to the small particles. It produces basic processing units for recognition algorithms. After segmenting the pages, we extracted the structural and visual features of the pages, and proposed a method to identify the body of the web article. The method uses clustering algorithm and heuristic rules to produce an automatic wrapper. A testing experiment demonstrated the efficacy of the algorithm.

Keywords: *visual information, page segmentation, information extraction, heuristic rules, clustering*

1. Introduction

In the field of data mining, it needs to pre-processing a web page when extract information from a web page [1]. Blocking the page is one of the pre-processing methods. At present, there are two kinds of page partition algorithms [2-4]: page blocking based on DOM tree model and visual model.

Page partition algorithms based on DOM model are mostly committed to dividing the page structure of the DOM tree into appropriate sub-tree [5-6], with which making up semantic chunks of the page. This method is relatively simple, but lack of good generality [7-10].

Page partition algorithms based on visual model need to extract visual information from large number of pages and use lots heuristic rules [11-12]. The time efficiency is relatively low for the necessary of rendering the page and extracting visual information [13]. But the page partition tends to be more accurate for visual information which can provide more page features [17]. In order to overcome the defects of the two methods, this paper puts forward a page blocking algorithm of based on DOM Structure and visual information, making effect of page block both efficient and accurate.

After data pre-processing, it needs to extract useful information using the information extraction technology. The Wrapper [18] is the most important and common method. Artificial wrapper is a most popular technology in early information extraction field. The disadvantage is that the maintenance cost is very high, and it needs amend with the change of the page template, but its accuracy is high [19]. Semi-automatic wrapper completes the wrapper with supervision or semi-supervision through integrating the machine learning method. At present, Semi-automatic wrapper methods include RAPIER, SoftMealy, ShopBot, STALKER, SRV, WIEN and WHISK etc. Automatic wrapper automatically generates wrapper with strong robustness using heuristic rules and machine learning methods, based on the analysis and summary structural features of large number of HTML source code under the same type of page. Roadrunner [20] is the most famous of automatic wrapper on the early period. It doesn't need user participation, has the

ISSN: 1738-9968 IJHIT

advantage of low maintenance cost, and has become the main development direction of information extraction technology.

This paper proposed the concept of text Block, completed design of page partition algorithm based on Block node; and then finished the full-automatic, unsupervised identification method based on Single-Pass clustering algorithm for the body of Web articles, and implements a full automatic wrapper.

2. Related Knowledge of Web Article Page Block and Information Extraction

2.1. Web article Model

The effective information of articles pages is title, body, abstract, illustration, subtitle of illustration and related link list. The article model is shown in Figure 1.

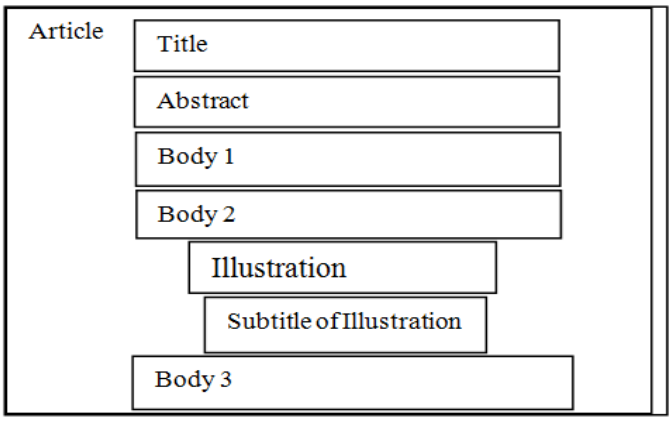

Figure 1. Web Page Article Model

2.2. Visual Feature

Currently, in the field of page information extraction, visual features is widely used, of which the most widely used is the normalized value of distance between page elements and the left margin of the browser and width of browser:

(1) $$L = \frac{Left}{ScreenWidth}$$

Where, Left is the distance between page elements and the left margin of the browser. And ScreenWidth is the width of browser. L is the normalized Left.

(2) $$W = \begin{cases} \frac{Width}{ScreenWidth}, & \text{if Right} < \text{ScreenWidth} \\ \frac{ScreenWidth - left}{ScreenWidth}, & \text{others} \end{cases}$$

Where, Width is the width of page elements. ScreenWidth is the width of browser. Right is the distance between page elements and the right margin of the browser. W is the normalized Width.

2.3. The Structure Characteristics of DOM Tree

The structure characteristics of DOM tree mainly refers to the information characteristics contained in page corresponding DOM tree structure, including the tag name, node types corresponds to page elements and path name from the root node to element nodes, *etc.*

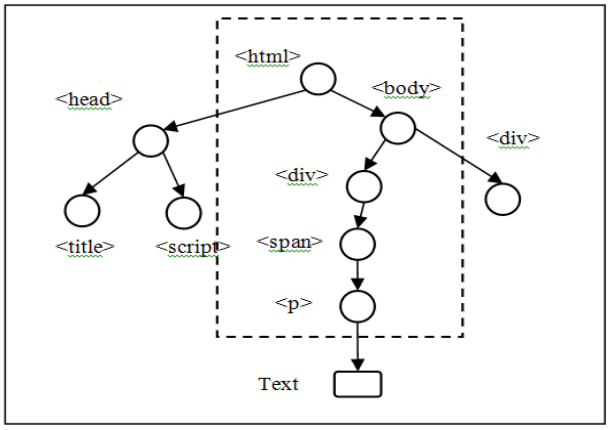

Figure 2. The DOM Tree Path of a Text Node

The DOM tree path is string consist of name of all the nodes from the DOM tree Document root node to a certain node. As is shown in Figure 2, the rectangle is a text node, the DOM tree nodes in the virtual box are the ancestors of the text node, in turn back up, and it's a DOM tree path is:

<HTML>|<BODY>|<DIV>|<SPAN>|<P>

2.4. Page Block Algorithm from Bottom to Top

VIPS algorithm is widely used in field of information extraction of page. But VIPS requires the user to preset the value of PDoC to complete the page block size requirements. For different page types and application requirement, the value of PDoC is often different and difficult to determine. Some algorithms [14-16] based on the applications and the improvements of VIPS are proposed, and page partition algorithm based on visual model is also gradually becoming a key research direction in the field of page partition algorithm.

Different from VIPS algorithm that block the page from top to bottom, using "BLOCK" value of "display" property of node gain the text segment from bottom to top can eliminate semantic BLOCK does not contain text. "Display" is one of the HTML Tag Attributes.

Block node: Element node whose value of "display" property is "BLOCK" in DOM tree.

Block chunk: Information set under Block node in DOM tree.

3. The Body Identification Method of the Web Article

3.1. Review Stage

3.1.1. Text Block

On the basis of Block, the concepts of minimum block of Block, Span and Strong are defined.

If it does not contain any other Block within a Block, we call this Block the smallest Block. If it does not contain any other Block within a Span, we call this Span the smallest Span. If it does not contain any other Block within a Strong, we call this Strong the smallest Strong.

Minimum blocks of Block, Span and Strong are semantic block which do not contain other Blocks. Figure 3 is pseudo code of discriminates weather a block of Block, Span or Strong is the smallest block of Block, Span and Strong or not.

Analysis shows that, the invalid text information that user do not interested in Web pages, such as the Script code and HTML explain, will not displayed on the page after the browser rendering the page. However, the valid text information that user interested in is placed in labels from "< P >", "< DIV >", "< BLOCKQUOTE >", "< CENTER >", "< H1 >" to "< H6 >", "< SPAN >" and "< STRONG >". These labels on the display properties, except "< SPAN >" and "< STRONG >", the default value is "BLOCK".

Further analysis shows that, the minimum block containing the valid text information of Block, Span, and Strong does not contain the non-text noise information such as pictures, video and flash animations, and within a Block is the full text information

```
input：Element node responding to one block of Block, Span or Strong
output：Weather an input block is a smallest block of Block, Span and Strong or not.
isMinBlock(Element)
    1 Init Queue，The elements of Queue is all the children of node Element.
    2 While Queue not null
    3    E = Poll from Queue
    4    If E is Block node
    5       Then return false
    6    Else pull all children of E to Queue
    7    End if
    8 End while
    9 Return true
```

Figure 3. Pseudo Code of Discriminates the Smallest Block of Block, Span and Strong

Usually, the valid text in minimum block of Block, Span and Strong will take palace overlap content but different visual effect. However, according to different visual features, the result of the same operation on different semantic Block with the same content is often different. So the smallest block of Block, Span and Strong having the same text can be processed as different semantic block.

Based on the above analysis, we can draw the following conclusions: for Web pages, text Semantic Block can be seen as all smallest blocks of Block, Span and Strong containing the effective visual text within the page. In order to facilitate discussion, all those blocks are generally called text Block. As is shown in Figure 3-2, rectangle represents the Element node, oval represents a text node, and blue rectangle is a text Block, namely the smallest block of Block, Span and Strong.

In Figure 4, the far left blocks of Span and Strong are not text Block, because they contain the Block. And the far right Block is a text Block, because in the text Block can contain the text Block consists of the minimum block of Span or Strong.

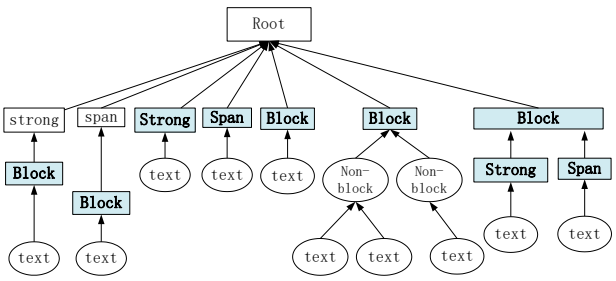

Figure 4.Text Block (Gray)

3.1.2. Page Block Algorithm Based on Block Node

To divide semantic block with the Web page article, an algorithm, which can output all text blocks within the page by entering the document node of one page, is needed to

implement. According to the analysis of the section 3.1.1, this paper proposed a page blocking algorithm based on node Block for Web page article.

```
input：Nodes of document of a page
output：Text block list
Find(document)
    1  Traverse to get the List of all Text nodes within one page.
    2  For each node in List
    3     Parent equals node's father node
    4     While Parent != null and Parent != document
    5        If parent can form a block of Block, Span and Strong
    6           Then If parent is not the smallest block of Block, Span or Strong
    7                 Then Break
    8              End if
    9              If set contain parent
    10                Then Break
    11             End if
    12             Add parent into Set and ResultList
    13             If parent is not block of Span or Strong
    14                Then Break
    15             End if
    16       End if
    17    End if
    18End for
    19Return ResultList
```

Figure 5. Pseudo Code of Page Blocking Algorithm Based on Block Node

The pseudo code of the algorithm is shown in Figure 5, the main process is as follows:

1. Traverse the DOM tree; get the List of all Text nodes within one page.

2. For each Text node in Text node list, searching ancestor nodes contains the Text node from bottom to top.

3. When searching into a non Block node, if the node forms a block of Span or Strong, and there is no record of this block in the results list, then the algorithm will add the Block to the results list, and continue to search up.

4. When searching into a Block node, if the block corresponding to this node is the smallest Block, then stop the search on the current Text node. Also, if there is no record of this block in the results list, the Block will be added to the results list.

5. When searching into the node of document, the search on the current Text node will be stopped.

6. Continue searching to the next Text node.

7. All Text nodes within one page is searched and returned to the Text block list.

3.2. Design for Identifying of Web Text

On basis of page partition algorithm based on Block node, this section first completes design of semi-automatic, semi-supervised identification method based on Bayesian classification algorithm, then complete design of automatic, unsupervised recognition method based on Single - Pass clustering algorithm.

3.2.1 Analysis of Characteristics of Web Text

Analysis on source code and visual layout of large number of Web page shows that, A Web text has the following visual and structural features:

1. The body consists of several texts.
2. The text is always centred on the most significant position of the page.
3. The page layout of most text is left-aligned.
4. The width of the most text is visually equal.

5. The font of the text uses the same style and size
6. All texts are in the same layout area of the page, with the same DOM tree path.
7. Generally, the number of period in the text is the largest of the whole page.

Feature 3, 4, and 6 are very significant among these features, which are embodied in having strict equal. For example, left-aligned of text segment is strictly aligned, equal width is strictly equal width, and the DOM tree path is strictly equal.

3.2.2 Identification Method Based on Single-Pass

In the process of Single - Pass clustering for page text blocks, algorithm using text feature distance measure dissimilarity degree between two text semantic blocks.

(3) $$D_{ij} = V_{ij} + S_{ij}$$

Where, i and j represent two different text blocks. Dij represents the text feature distance between i and j; Vij represents visual distance between i and j; Sij represents structural distance between i and j.

Feature distance between i and j consists of two parts, they respectively are Visual features distance and structural features distance. Visual features distance is defined as follows：

(4) $$V_{ij} = \sqrt{(l_i - l_j)^2 + (w_i - w_j)^2 + (f_i - f_j)^2}$$

Structural feature distance is defined as follows:

(5) $$S_{ij} = \frac{\theta}{\cos(P_i, P_j) + 0.1}$$

Where, cos (Pi, Pj) represents cosine similarity of DOM tree path between i and j；Θ represents predefined values.

In the process of Single-Pass clustering, the algorithm uses text feature distance measure dissimilarity degree between instance and representative instances.

The pseudo code using Single-Pass Clustering to partition text block is shown in fig. 6, the specific workflow is as follows:

(1) Instances in text block list are processed serially in the algorithm.

(2) The first instance in text block list is allocated to the first cluster and as its representative.

(3) Clustering each subsequent instance in text block list, calculating dissimilarity degree between the instance and all representatives of generated clusters using text feature distance, recording the cluster with the smallest dissimilarity degree.

(4) If the text feature distance between instance and recorded clusters is not greater than the predefined threshold, adding the instance into this cluster and calculating the representative of the cluster again.

(5) If the text feature distance between instance and recorded clusters is greater than the predefined threshold, building a new cluster, allocating the instance to this cluster and taking it as the new cluster’s representative.


```
Input：A text block list
Output：Several text block lists
Cluster (input)
    1  Init Queue List<List>，a List represents a cluster，element is a text block
    2  Index = the length of input
    3  C = the number of clusters in a List<List>
    4  Threshold = predefined threshold
    5  Construct a new cluster L in List<List>, add the 0th element of input into the new
         cluster.
    6  For i from 1 to Index
    7     minJ = 0
    8     minDis = input[i] and dissimilarity degree represented by the first cluster in
         List<List>
    9     For j from 1 to Index
    10       If input[i] < minDis
    11          Then minDis = input[i] and dissimilarity degree represented by the jth cluster in
         List<List>
    12               minJ = j
    13       End if
    14    End for
    15    If minDis < Threshold
    16       Then add input[i] into the minJth clusters in List<List>， compute the
         representative of the new cluster。
    17    Else construct a new cluster in List<List>, add input[i] into the new cluster.
    18    End if
    19 End for
    20 Return List<List>
```

Figure 6. Pseudo Code of Using Single-Pass Clustering for Text Block

After Single-Pass clustering for the text blocks, it needs complete the recognition for Web text. According analysis of the general characteristics of the Web text in section 3.2.1, Web text consisting of a number of text blocks is always in the most prominent position of the page, and the number of periods in the entire page text is the largest. Therefore, this paper uses the following two rules to identify class of the body.

Rule 1：The distance between the first text block within the class and margin-top of the browser is not greater than two times of the height of the browser.

Rule 2：Statistics all the texts according to Punctuation table for each class, the text classes is class with the largest number of punctuations.

There are two punctuations in punctuation table of this paper, making the method usable both for domestic pages and foreign pages. One is period “ 。 ” in Chinese webpage, the other is full stop “ .” in English webpage.

After identifying of text class, adding the missing text blocks to the text class for the complement operation. Specific workflow of complement operation is as follows:

1. The text blocks are divided into text and non-text blocks after Clustering of text blocks.

2. Obtain the visual and structural features of text blocks via text block list, including: common font size, DOM tree path, and top-bottom-left-right boundaries of text block. The common font size of text block is the font size used mostly in all the text blocks.

3. Non-text blocks are processed serially.

4. When the font size of the text block is the common font size, DOM tree path is the DOM tree path of the text block, and the text blocks is in the area of text block, then add the text block to the text blocks.

5. Complete the complement operation.

Thus, the identification of the article body is completed based on Single-Pass clustering algorithm.

3.3. Text Extraction of Web Article

For a web page, the browser interpreted the source code of the page from top to bottom, and for the text within the text block, the content is formed from top to bottom in the source code. Therefore, the extraction rule for contents in text block is: preorder traversal DOM sub-tree corresponding to text blocks, extract all text nodes in turn, then joint into the full text in a text block. Figure 7 shows the DOM sub-tree corresponding to a text block, this text Block is a minimum Block, and its text joint of five leaf nodes ABCDE.

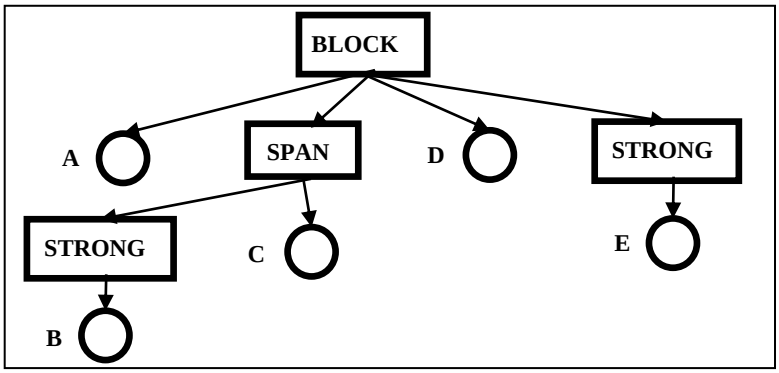

Figure 7. The DOM Tree Corresponding to a Text Block

4. Experimental Results and Analysis

The experimental environment in this section is operating system Windows 7 32bit, JDK1.6.0 _21 32bit, Intel (R) Core (TM) 2 Quad CPU 2.50 GHz, and 2 GB of RAM. The experimental data is article page from NetEase, tencent, CNN, Web sites domestic and overseas such as the United States Newsweek. We use measure F1 of β=1 for metric F.

Table 1 is the experiment result of VIPS algorithm when PDoC= 10, Table 2 is result of page block based on node Block. Experimental results` statistic is some information about text segments in Web article. If a text segment is included in a semantic block with only text, then the text segment is properly blocked. Table 3 is result of identification of Web text using Naive Bayesian classification algorithm. Table 4 is result of identification of the Web text using Single-Pass clustering algorithm.

Table 1. Experiment Results of VIPS Block (10)

website	The number of text segment	the number of recall	Correct number	Precision	Recall	F1
NetEase	334	334	334	1.000	1.000	1.000
Sina	655	655	655	1.000	1.000	1.000
CNN	450	450	427	0.949	1.000	0.974
total	3577	3449	3426	0.958	0.964	0.961

Table 2. Experiment Results of Page Block Based on Node Block

website	The number of text segment	the number of recall	Correct number	Precision	Recall	F1
Netease	334	334	334	1.000	1.000	1.000
Sina	655	655	655	1.000	1.000	1.000
CNN	450	450	450	1.000	1.000	1.000
total	3577	3495	3495	0.977	0.977	0.977

Table 3. Experiment Results of Identification the Web Text Using Naive Bayesian Classification Algorithm

website	Token Count	identification number	Correct number	Precision	Recall	F1
Sina	655	659	655	0.994	1.000	0.997
CNN	450	429	417	0.972	0.927	0.949
total	3534	3481	3152	0.905	0.892	0.899

Table 4. Experiment Results of Identification the Web Text Using Single-Pass Clustering Algorithm

website	Token Count	identification number	Correct number	Precision	Recall	F1
Sina	655	612	609	0.995	0.930	0.961
CNN	450	449	448	0.998	0.996	0.997
total	3534	3454	3413	0.988	0.966	0.977

The experiment results show that, the page block algorithm based on node Block can reduce the noise information in the Block better compared with the general page block algorithm VIPS. And the recognition method based on Single-Pass clustering algorithm has good effect, has stronger applicability than any other method based on classification algorithm, and recognition method based on Single-Pass clustering algorithm can achieve an automatic, no supervision wrapper for Web text. Although the effect is good, but as is show in time efficiency contrast Figure 8, this unsupervised approach takes more time than other half supervision methods. The difference is more obvious in the more complex pages.

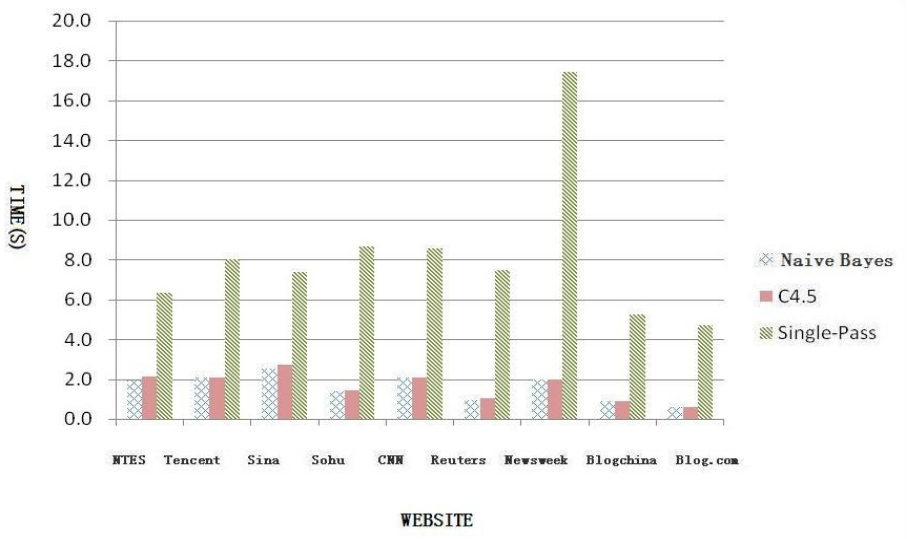

Figure 8. Time Efficiency Comparison of Three Identification Methods

5. Conclusion

The paper analyzed DOM tree structure and visual information of a lots of Web pages, proposed a automatic extraction method of the Web page information, and achieved a common automatic wrapper of extracting information from Web page articles.

Firstly, through the analysis of a large number of Web pages, the paper puts forward a Web page partition algorithm based on node Block. Algorithm successfully converts half structure pages to structured data. Then, on the basis of page block, the paper proposed and realized the automatic identification algorithm for Web text using Sing - Pass clustering method, and gave the distance metric between text semantic blocks. The

experimental results show that, this method can achieve an efficient, without any user involved automatic wrapper. Compared with recognition method using Naive Bayesian and C4.5 decision tree classification algorithm, this method has higher values of F1 and good effect.

Although the effect of proposed page partitioned and web text recognition method is good, compared with a semi-supervised automatic wrapper, the time efficiency of this unsupervised automatic wrapper needs to be improved.

Acknowledgments

During writing the paper, prof. Zhao Zheng gives many better advices and provides much data, Student Lian Yangyang helps making tests. Thank these people who helped me.

References

[1] J. F. Wang, C. Chen, C. Wang, "Can We Learn a Template-Independent Wrapper for News Article Extraction from a Single Training Site", Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, **(2009)**, pp. 1345-1353.
[2] S. Mukherjee, G. Yang, W. Tan, "Automatic Discovery of Semantic Structures in HTML Documents", Proceedings of the 7th International Conference on Document Analysis and Recognition, **(2003)**, pp. 245.
[3] J. Luo, J. Shen and C. Xie, "Segmenting the Web Document with Document Object Model", Proceedings of the IEEE International Conference on Services Computing, **(2004)**, pp. 449-452.
[4] S. H. Lin and J. M. Ho, "Discovering Informative Content Blocks from Web Documents", Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, **(2002)**, pp. 588-593.
[5] C. Kang, "DOM-based Web Pages to Determine the Structure of the Similarity Algorithm", Intelligent Information Technology Application. Third International Symposium, **(2009)** February, pp. 245-248.
[6] R. Liu, R. Xiong and K. Gao, "Web Object Block Mining Based on Tag Similarity", Intelligent Computation Technology and Automation International Conference, **(2010)** March, pp. 1159-1162.
[7] N. Hui and H. Guipeng, "The application of tree edit distance in Web information extraction and implementation", Modern information technology, vol. 5, **(2010),** pp. 29-34.
[8] D. C. Reis, P. B. Golgher, A. S. Silva, "Automatic Web News Extraction Using Tree Edit Distance", Proceedings of the 13th International Conference on World Wide Web, **(2004)**, pp. 502-511.
[9] Y. Kim, J. Park, T. Kim, "Web Information Extraction by HTML Tree Edit Distance Matching", Proceedings of International Conference on Convergence Information Technology, **(2007),** pp. 2455-2460.
[10] Y. H. Zhai and B. Liu, "Web Data Extraction Based on Partial Tree Alignment", Proceedings of the 14th International Conference on World Wide Web, **(2005)**, pp. 76-85.
[11] D. Cai, S. Yu, J. R. Wen, "Vips: a Vision Based Page Segmentation Algorithm", Microsoft Technical Report. MSR-TR-2003-79, **(2003),** pp. 10.
[12] D. Cai, S. Yu, J. R. Wen, "Extracting Content Structure for Web Pages based on Visual Representation", Proceedings of the 5th Asia-Pacific web conference on Web technologies and applications, **(2003)**, pp. 406-417.
[13] S. Yu, D. Cai, J. R. Wen, "Improving Pseudo-Relevance Feedback In Web Information Retrieval Using Web Page Segmentation", Proceedings of the 12th International Conference on World Wide Web, **(2003)**, pp. 11-18.
[14] R. R. Mehta, P. Mitra and H. Karnick, "Extracting Semantic Structure of Web Documents Using Content and Visual Information", Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, **(2005)**, pp. 928-929.
[15] W. Liu, H. L. Yan, J. W. Yang, "A Unified Approach for Extracting Multiple News Attributes from News Pages", Proceedings of the 11th Pacific Rim International Conference on Trends in Artificial Intelligence, **(2010)**, pp. 157-169.
[16] W. Liu, H. L. Yan, J. G. Xiao, "Extracting multiple news attributes based on visual features", Journal of Intelligent Information Systems, vol. 38, no. 2, **(2012),** pp. 465-486.
[17] P. Luo, J. Fan, S. Liu, "Web Article Extraction for Web Printing: a DOM+Visual based Approach", Proceedings of the 9th ACM Symposium on Document Engineering, **(2009),** pp. 66-69.
[18] S. Flesca, G. Manco, E. Masciari, "Web Wrapper Induction: a brief survey", AI Commun, vol. 17, no. 2, **(2004),** pp. 57-61.

[19] A. Sahuguet and F. Azavant, "Building Lightweight Wrappers for Legacy Web Data-sources Using W4F", Proceedings of the 25th International Conference on Very Large Data Bases, **(1999)**, pp. 738-741.
[20] V. Crescenzi, G. Mecca and P. Merialdo, Road Runner: Towards Automatic Data Extraction from Large Web Sites", Proceedings of the 27th International Conference on Very Large Data Bases, **(2001)**, pp. 109-118.

Author

Weiguo Shen is born in Tianjin, China, December 1963. He received the degree in Mechanical Engineering from the DongHua University, in 1985. He received the master's degree in Textile Engineering from Tianjin Polytechnic University, in 2011. Currently, he is a lecturer at School of Mechanical Engineering & Automation of Tianjin Polytechnic University. His research interests include intelligent control and Intelligent Testing algorithm.

Zou Xiaojian is born in Dehui, China, May 1978. He received the degree in Information and Computing Science from the Jilin University, in 2003. He received the master's degree in operational research & cybernetics from the Nankai University, in 2011. Currently, he is a lecturer at Military Transportation University. His research interests include intelligent control and intelligent algorithm.

