A novel web page text information extraction method

Chongjun Wang[1], Peng Wei[2]

1. Aerospace Engineering University, Beijing, 101400

2. National Digital Switching System Engineering Technology Research Center, Zhengzhou, Henan, 450002

2298238184@qq.com, 2093538947@qq.com

Abstract—**The text information of today's mainstream web pages generally has multiple features, and can be divided into a single text page and a multi-text page. In order to extract the text information of the webpage, the position of the text information can be accurately located by using the multiple features of the text and the rules of the webpage design. According to the above characteristics, this paper proposed a method for extracting webpage text information based on multi-feature fusion. Experiments based on a large amount of data showed that the method has universality and high accuracy for the text information extraction of single text and multi-text web pages, and is very suitable for web pages with various styles.**

Keywords—text information extraction;multi-feature fusion; multi-text body webpage; single text body webpage

I. INTRODUCTION

With the rapid development of Internet technology and the expansion of information, not only is the Internet today flooded with spam, but Web pages are no longer as simple as the previous pages. The page contains a lot of elements such as display styles, scripts , lots of ads and so on. How to find useful information from numerous spam, and find the location of the topic information in the webpage accurately and completely , which has become a hot topic in current research.

In the field of web page text information extraction, there has been a lot of research work and many mature methods. The web page data source that is extracted from the same website or web page structure is mainly based on DOM tree structure and other extension methods. Many researchers combind the two methods for information extraction, such as the Road Runner system[1]; the extracted web page data source is not limited to the same website based on visual feature-based methods[2], based on statistical theory[3] and so on. In statistical theory method, determining the position of the text information by using the textual features of the webpage mainly includes: Song[4] used the three common features of the body information (punctuation, non-hypertext and hyperlink text), which are converted into statistical information values to determine the position of the body information; Zhou[5] continued Song's method and improved in the subsequent processing, and proposed the a method to further improve the extraction effect, adapted to a variety of web pages.

In practice, today's statistical theory-based methods have their limitations. With the diversification of webpage styles, the extraction accuracy is reduced and the versatility is not strong. The purpose of this paper is to develop a real-world application for text information extraction and structuring for different types of web pages. The system is suitable for different styles of web pages and any website. The high accuracy and versatility of the extracted results is a difficult point in designing the web page text information extraction algorithm. Based on the URLs in the well-known navigation websites of this paper, we study the method of extracting the text information of web pages with high accuracy and versatility. In order to meet the diversity of webpage style and the versatility of the algorithm itself, a multi-feature based webpage information extraction method, WFFTE (Webpage multi-feature fusion text extraction) is proposed.

978-1-5386-6243-4/19/$31.00 ©2019 IEEE

II. Proposed method

A. Denoising

In order to improve the processing speed, some obvious noise data in the source file, such as HTML comments and scripts, should be deleted before the tag tree is built to improve the processing efficiency.

Regular expressions are used to filter noise data and analyze HTML documents. The <HEAD> flag contains additional information about the title of the document, the description of the document, the script used by the document, the style definition, and the document name. The information that the user can browse on the browser is stored in the <BODY> section, that is, the body information is in the <BODY> section. Therefore, this article only deals with the contents of <BODY>. Statistical analysis of a large number of HTML documents found that some of the obvious noise information contained in the webpage is mainly[6]:

(1) internal style text, which is the <style>...</style> style block;

(2) JavaScript script[7], which is the <script>...</script> style block;

(3) HTML comments, which is the <!-- ...--> style blocks;

(4) The entire content that is not included in the <body> tag, which is the <head>...</head> style block. According to common sense, the body of a web page must appear after the <body> tag.

The processing method is shown in Table 1. Experiments show that after denoising by regular expressions, the document size is roughly reduced to 30%, which greatly simplifies the subsequent work.

Table I. Details and steps for deleting redundant tags

Delete content	Regular expression
<head>···</head>	<head[^>]*?>[\s\S] *?</head>
<script>···</ script >	< script [^>]*?>[\s\S] *?</ script >
<style>···</ style >	< style [^>]*?>[\s\S] *?</ style >
<!--···-->	<!--[^-]*-->

B. Multi-feature fusion method

After analyzing a large number of web pages, it is found that web pages are mainly divided into single-body web pages and multi-text web pages.

The textual feature of a single-textual web page[8] is mainly concentrated in a container tag, which contains a lot of punctuation marks. The text has some word for the description of the title, and more of the other container tags are closer to the title tag.

The main features of multi-text web pages are that the text is distributed across multiple container tags, and according to the visual habits of web design, the display style of these container tags is likely to be the same, which may also contain many punctuation marks. The text also has a few descriptions of the title, near the title tag.

In summary, the analysis and description of the characteristics of different types of web pages, the web text body has a number of features including: the number of text text, body punctuation, text hyperlink text, non-hyperlink text relationship, the body of the descriptive language of the title, the distance from the title to the title, and the style and location of the body information.

The WFFTE proposed by combining the text features of these web pages is to convert the html document into a DOM tree and calculate the text support of each container label. At the same time, along with some processing in the calculation process and after calculation, the algorithm steps are described in Section 2.3.

In this paper, the third-party jar package of Jsoup.jar is used to construct the DOM tree of the webpage. The function of the jar package is to fix the default label of the webpage first, and then parse the html document to traverse all the tabs with the html tag as the root node. The DOM tree is shown in Figure 1.

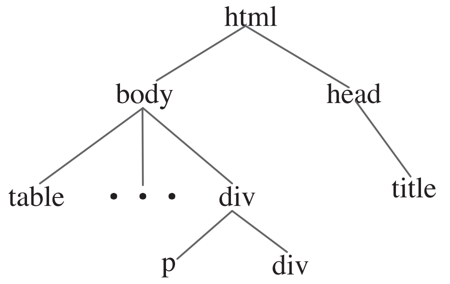

Fig.1. Web page DOM tree

When traversing the DOM tree[9], the uniqueness of each container label position path is achieved by uniquely labeling each container label, for example, by labeling the DOM tree of Figure 1 and obtaining the result as shown in Figure 2.

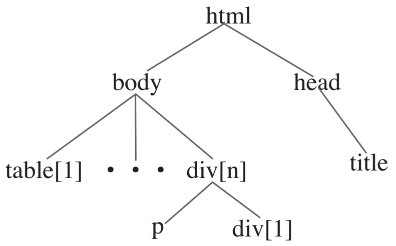

Fig.2. labeled DOM tree

The only paths we can get for each container tag are html/body/table[1],..., html/body/div[n], html/body/div[n]/div[1], and the title tag's path is html/head/title. Calculate the text support (SD) of each container label. Its calculation idea is to divide the text features into three categories. The first category is the distance from the text to the title, thereby calculating the distance support (DSD). The second category is The body's descriptive language for the title, which is used to calculate the title support (TSD); the third is the relationship between the number of body texts, the body punctuation, the body hyperlink text, and the non-hyperlink text, thereby calculating the general support (PSD)[10]. Then, use the support to calculate the total text support to determine the container tag that is most likely to be the body. The calculation method is as follows:

$$SD = DSD \times (TSD + PSD) \quad (1)$$

Where SD is the text support degree, DSD is the distance support degree, TSD is the title support degree, and PSD is the general support degree.

The computational idea of DSD is that a container tag is more closer to the title tag, it is more likely to support it as a container tag containing the body. The specific method is to map all the container labels and title labels into the one-dimensional coordinate system. When calculating the distance support degree, the title is assumed to be the origin on the coordinate system, and the container label can be mapped and converted into points on the coordinate system by using equation (2), thereby calculating the distance support.

$$DSD = 1/(\sum_{i\in n}^{n} rd_i \times q^{i-1}) \quad n \geq 1 \qquad (2)$$

Where rdi is the serial number on the container label path, and q is an integer constant greater than 0. It is reasonable to find q = 10 during the experiment, for example: a container like html /body /div[1]/ div[1], rd1 = 1, rd2 = 1, then DSD = 1/(1×100+1×10-1) = 1/1.1.

The calculation idea of TSD is that the more entity words that contain the title in the container tag, the more likely it is to support it as the container tag containing the body. The specific method is to first segment the title, extract the entity words inside, then count the total number of occurrences in all container tags, select the two words with the most occurrences (FirstWord and SecondWord), use the formula (3) to get the title support.

$$TSD = \alpha \times FW + \beta \times SW \qquad (3)$$

Where FW is the number of times that FirstWord appears in the container tag, SW is the number of times that SecondWord appears in the container tag, α and β are two constants, and α<β, in order to balance the calculated text support and consider the degree of importance of these two words, their values in the experiment are α = 0.5, β = 1.

The design idea of PSD is: every web page has universal features, namely punctuation super, link text, and non-hyperlink text. Through the following calculation methods, establish the relationship between them and calculate their support for the container label, the specific formula is as follows:

$$PSD = FP \times (NC / HC) \qquad (4)$$

$$FP = \begin{cases} 0 \leq p < 3 & FP = 0.001 \\ 3 \leq p < 6 & FP = 0.1 \\ p \geq 6 & FP = 0.5 \end{cases} \qquad (5)$$

Where NC is the number of words of unlinked text, HC is the number of words of the linked text, FP is the punctuation support, and p is the number of punctuation. The less the punctuation is, the less likely it is to have the body, so reduce the information support for the body itself.

Since the calculation of the path distance is involved in Section 2.3, its related concepts and calculation methods are first introduced here. The path distance is a description of the distance between the container labels.

$$juli(i,j) = (len(i) - pre(i,j)) + (len(j) - pre(i,j)) - 1 \qquad (6)$$

Where juli(i,j) is the path distance between the container tag i and the container tag j, len(i) is the number of nodes through which the root node passes to the container tag i, and len(j) is the number of nodes of the root node to the container tag j, pre(i,j) is the number of identical nodes experienced by the container tag i and the container tag j. For example, the path of container tag i is html/body/div[1]/div[2]/div[2] and the path of container tag j is html /body /div[1]/div[2]/div[3], at this time len(i)=5, indicating that it takes 5 nodes to go from the root node to the label i; the same can be considered len(j) = 5, pre(i,j)= 4, and finally juli can be calculated. i, j)=1. Another example:

if there is a label path html/body/div[1]/div[1]/div[1]/div[0] to html/body/div[1]/div[1]/div[2]/div[1], the distance of the path is 3.

C. Algorithm step

The specific processing steps of the WFFTE method are described as follows:

Input: Source code of a web page

Output: the result set of the extracted information

Step:

(1) Clear the noise tags such as script, meta, style, and save the 'title' title tag and subtitle tag.

(2) Traverse the web page DOM tree, extract the container labels in turn and save them in the form of key-value, where key is the path of the label and value is the content contained in the container label. Calculate the PSD using equation (4) in units of container labels.

(3) Segment the title tag and the sub-title tags at each level, and count the two most frequently occurring words (FirstWord and SecondWord). Calculate the TSD value of each container label using equation (3).

(4) Traverse the container label set again, calculate the DSD of each container label using equation (2), and calculate the SD of each container label using equation (1).

(5) Calculate the path distance between the container tags using equation (6). When juli=1, compare the class attributes of the two container tags. If not, compare their style attributes with other attributes. If the attributes are the same, merge their contents and add their SD values.

(6) Select the container label of the first seven SD values, and select the container label with the largest SD value. The text length is the proportion of the total length of the text length of the first seven container labels. If it is greater than or equal to 0.5, set the path distance threshold JULI = 2, and if less than 0.5, set the path distance threshold JULI = 4.

(7) Calculate the path distance juli of the container label from the first seven container labels to the maximum SD value. If juli ≤ JULI, add the container label to the set 'lastset' which will be returned.

(8) Traverse the elements in the lastset. If there is only one element in the 'lastset', it will return directly. If not, it will judge the elements inside. The method of judging is as follows: if there are words in the content of the element (3) or more of the lexicon (the lexicon of the copyright information) and there is no punctuation, the element will be directly discarded.

III. EXPERIMENT AND EVALUATION

The experiment selected 10 websites: Jinritoutiao, Sohuxinwen, Baidutieba, Tengxuntiyu, Ubuntu, Wangyixinwen, Sinaweibo, Bokeyuan, CSDN, and Ganjiwang. 200 web pages were randomly selected from these websites, and the experimental results are shown in Table 2. Use the accuracy P and completeness R of the extracted text information to evaluate the experimental results[11], and their calculation formula is as follows:

$$P = \frac{C2}{C1} \times 100\% \tag{7}$$

$$R = \frac{C3}{C2} \times 100\% \tag{8}$$

Where C1 represents the total number of web pages in the experiment, C2 represents the number of web pages that correctly extract the text information, and C3 represents the number of web pages that completely extract the text information. The accuracy rate is based on the total number of web pages, and the complete rate is based on the number of web pages that correctly extract the text information[12].

Table II. Experimental results

Page source	Total number of pages	Correct number of extractions	Accuracy rate (%)	Number of complete extractions	Complete rate (%)
Jinritoutiao	300	290	96.7	276	95.2
Sohuxinwen	300	293	97.7	282	96.2
Baidutieba	300	282	94	273	96.8
Tengxuntiyu	300	278	92.7	271	97.5
Ubuntu	300	281	93.7	270	96.1
Wangyixinwen	300	279	93	268	96.1
Sinaweibo	300	295	98.3	291	98.6
Bokeyuan	300	278	92.7	269	96.8
CSDN	300	287	95.7	271	94.4
Ganjiwang	300	286	95.3	279	97.6

The method in this paper is compared with the LORI and SUN methods for determining the position information of the text based on the textual features of the literature[4], [5]. The experimental results are shown in Table 3 below.

Table III. Experimental comparison results

method	Method of this paper	LORI	SUN
Web source	Accuracy rate (%)		
Jinritoutiao	97.3	94.7	92.4
Sohuxinwen	98.1	95.3	93.2
Baidutieba	95	92	89.4
Tengxuntiyu	93.8	91.2	88.6
Ubuntu	95.3	92.3	90.2
Wangyixinwen	96	91.4	89.6
Sinaweibo	97.3	95.6	92.5
Bokeyuan	93.1	89.3	87.3
CSDN	96.4	93.2	91.2
Ganjiwang	96.3	91.4	87.8

As can be seen from Table 3, the extraction accuracy of single-text webpages and multi-texture webpages is higher than that of the conventional methods. Figure 3 is an intuitive bar graph corresponding to Table 3.

In Figure 3, the higher accuracy of the method herein can be visually found. More and more perfect webpage feature factors are added to the webpage information extraction method, which is very helpful for the extraction effect. The extraction result of the single-text style webpage and the multi-textual style webpage has a high correct rate.

IV. CONCLUSION

Since the method is based on large-scale training, the experimental results are not much different from the actual application, which can meet certain scientific research and practical applications. The extraction effect of web pages with multiple texts is slightly worse than that of single texts. The reason for the analysis is that the tags in the web structure are deeply nested or their display styles are different and the distances are far apart. There are many features for webpage texts, and especially the multi-text body has its own uniqueness, so it will be further studied in the later stage, and the algorithm has the space of improvement.

REFERENCES

[1] Crescenzi V, Mecca G. Road Runner: towards automatic data extractionfrom large Web sites[C] / /Proc of the 27th VLDB Conference. SanFrancisco: Morgan Kaufmann Publishers, 2001: 109-118.

[2] Cunhe L, Juan D, Juntang Chen. Extraction of Informative Blocksfrom Web Pages Based on VIPS[J] . Journal of Computational Information Systems, 2010: 271-277.

[3] Sun C J, Guan Y. Research on the Method of Extracting Body Text Information Based on Statistics[J]. Chinese Journal of Information Science, 2004(5): 17-22.

[4] Song M Q, Wu X T. Content extraction from Web pages based on Chinese punctuation number[C] //Proc of International Conference on Wireless Communications, Networking and Mobile Computing, 2007.

[5] Zhou J Y, Zhu Z M, Gao X F. Research on Chinese Web Page Text Extraction Based on Statistics and Text Features[J]. Chinese Journal of Information Science, 2009(5): 80-85.

[6] Kong S,Wang Y.A Method of News Web Page Extraction Based on Text Features[J].Journal of Information, 2010, 29(8): 122-124.

[7] Rostami S, Eshkevari L, Mazinanian D, et al. Detecting Function Constructors in JavaScript[C]// IEEE International Conference on Software Maintenance and Evolution. IEEE, 2017.

[8] Xiang-Dong H U, Liu K, Zhang F, et al. Financial phishing detection method based on sensitive characteristics of webpage[J]. Chinese Journal of Network & Information Security, 2017.

[9] Kim Y, Lee S. SVM-based web content mining with leaf classification unit from DOM-tree[C]// International Conference on Knowledge and Smart Technology. IEEE, 2017.

[10] Mirończuk M M. The BigGrams: the semi-supervised information extraction system from HTML: an improvement in the wrapper induction[J]. Knowledge & Information Systems, 2018, 54(3):711-776.

[11] Yusufu M, Yusufu G. Repetitive Pattern Recognition Algorithms and Applications in Web Information Extraction and Clustering Analysis[J]. Computer Science, 2017.

[12] Peng Y B, Xie X T. The adaptive Web information extraction based on single DOM tree characteristics and classification[J]. Electronic Design Engineering, 2017.

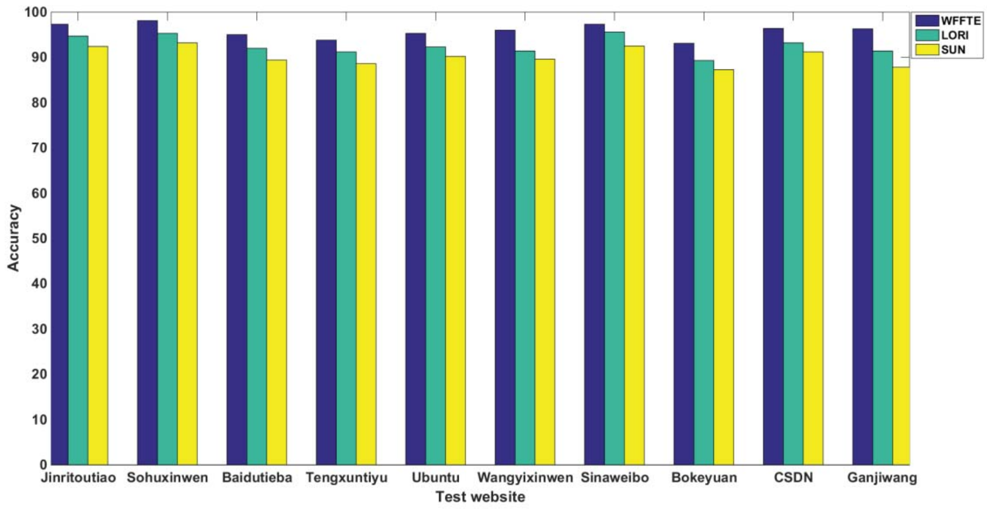

Fig.3. Comparison of experimental results