An Algorithm to Extract and Judge the Main Text Based on the Law of Total Probability

Qingsong Lv[1, a], Shulin Cao[1, b], Yifan Wang[1, c], Qian Yin[1, d*] and Xin Zheng[1, e*]

[1]College of Information Science and Technology, Beijing Normal University, Beijing, China

[a]201511210102@mail.bnu.edu.cn, [b]201511210110@mail.bnu.edu.cn, [c]201511210101@mail.bnu.edu.cn, [d]yinqian@bnu.edu.cn, [e]zhengxin@bnu.edu.cn

* The Corresponding Authors

Keywords: Web page extraction; Web page classification; Law of total probability.

Abstract. Since Internet web pages have diverse contents and complex structure, it is of great significance to use a uniform algorithm to tackle them. In this paper, we proposed an algorithm called P value algorithm to extract the main text of one webpage. By calculating the P value of each tag in an HTML page, we can locate where the main text is. Moreover, the P value of a web page can also represent the probability of "This web page has main text". The experiments show that the accuracy of extracting web pages is 95.42% and the accuracy of judging whether a page has main text is 93.98% without any prior knowledge.

Introduction

Besides main text, most web pages also include noise information such as navigation bar, sidebar and advertisement. The goal of extracting the main text of a web page is to remove the noise information and extract the core text information.

To resolve this problem, there are many algorithms so far such as heuristic algorithm[1], ECON algorithm[2], TTR algorithm[3], MSS algorithm[4], CTD algorithm[5] and machine learning algorithm[6][8][9][10][11]. These algorithms have some problems as follows:

1. For the algorithms in reference [1] and [3], threshold values are needed to be set manually, which leads to the uncertainty of the algorithm performance.
2. For the algorithms in reference [4] and [5], the formulas are too complex, which makes the algorithm are not so efficient.
3. For the algorithm in reference [2], the accuracy of algorithm is not high enough.
4. For the algorithm in reference [6][8][9][10][11], prior knowledge is necessary to extract specific web pages.

In this paper, we propose one P value algorithm on the basis of CTD algorithm. The formula of P value is based on the law of total probability and is very concise. And the P value describes both the probability of a tag being the main text and a web page having a main text area.

Main Text Extraction

Hypotheses. Above all, we will propose two hypotheses.

- The accurate main text area is exactly included by a tag (such as <div></div>).
- There is a value P of a tag, which represents the probability of the tag being the main text area.

Algorithm Procedure. Under the premise of the hypotheses, the procedure of P value algorithm is here.

Step 1. Remove the contents which are certainly not main text area, including three parts.

- annotation.
- <script>, <noscript>, <style>, <embed>, <label>, <form>, <input>, <iframe>, <head>, <meta>, <link>, <object>, <aside>, <channel>, <img> tags.
- "id", "class" and "style" attributions in other tags.

Step 2. Calculate P value of each tag and choose the tag with the maximum P value as the main text area.

Step 3. Remove all the tag heading (such as <div>) and tag tailing (such as </div>) in the chosen tag and the remaining character string is the main text.

Method to Calculate P Value. There are three definitions below.

Definition 1: Length of a tag. The character string length of a tag including tag heading and tag tailing and all the character between them.

Definition 2: Text length of a tag. The character string length of a tag after removing all the tag heading and tag tailing in the tag.

Definition 3: Valid text length of a tag. Text length of a tag minus text length of all the <a> tag in this tag.

The reason why valid text length of a tag is text length of a tag minus text length of <a> is in the article [5].

The formula of P value of a tag:

$$P = \frac{l_t}{l_s} * \frac{l_{vt}}{L_{VT}} \tag{1}$$

l_t represents text length of the tag. l_s represents length of the tag. l_{vt} represents valid text length of the tag. L_{VT} represents valid text length of the whole html.

Explanation of Eq. 1. The formula can be viewed from probability theory.

Assuming that event A is “Text in the tag can represent the main information of the tag” and event B is “The tag is the main text area of the web page”.

If semantic factors are not taken into account, $\frac{l_t}{l_s}$ is $P(A)$ and $\frac{l_{vt}}{L_{VT}}$ is $P(B|A)$. Besides, we consider $P(B|\neg A)$ is very small because $\neg A$ means the text cannot represent the main information of the tag, so the probability of the text being main text is even smaller. That is, $P(B|\neg A) \approx 0$.

Now we can get the Eq.1 from the law of the total probability.

$$P(B) = P(B|A)P(A) + P(B|\neg A)P(\neg A) \approx \frac{l_{vt}}{L_{VT}} * \frac{l_t}{l_s} + 0 * P(\neg A) = \frac{l_t}{l_s} * \frac{l_{vt}}{L_{VT}} \tag{2}$$

When we utilize P value to extract main text, we only care about the relative P value of each tag and L_{VT} for a specific web page is constant. So we can also use P’ as follows.

$$P' = \frac{l_t * l_{vt}}{l_s} \tag{3}$$

Web Page Judgement and Classification

A definition.

Definition 4: P value of a web page. The maximum P value of all the tags in a web page is the P value of the web page.

Method.

After a large amount of tests, we found that P value of a web page has correlation with the probability of whether a web page has a main text area. And we get this conclusion: P value of a web page is equal to the probability of whether a webpage has a main text area.

Consequently, if the P value of a page is not smaller than 0.5, we consider the web page has a main text area, which means the page has an article in it. Otherwise, if the P value of a page is smaller than 0.5, we consider the web page does not have an article in it.

Experiments

Main Text Extraction.

First, we use the dataset Chaos in the reference [6] to experiment. Since the links in the dataset is from three years ago, many of them are not valid and we just use the valid links. We also use the

Precision, Recall, F1 score [7] to evaluate the results so that we can make a comparison with the result in reference [6]. Table 1 shows the result.

Table 1 Result compare between CTD and P value algorithm

Algorithm	Dataset	Precision	Recall	F1 score
CTD	Chaos	96.86%	89.64%	92.83%
P value	Chaos	96.97%	98.21%	97.62%

From Table 1, we can see that P value algorithm performs better in all the three aspects.

Moreover, we use all the valid links in the dataset in reference [6] including 262 web pages. Then we count the number of accurately extracted pages which means Precision, Recall and F1 score are all bigger than 95%. Table 2 shows the results.

Table 2 Accuracy testing result

Algorithm	Number of all the pages	Number of accurately extracted pages	Accuracy ratio
P value	262	250	95.42%

Web Page Judgement and Classification

We picked out 83 web pages from BBC news website and classified them manually. There are 28 web pages with an article in it, and 55 not. Table 3 shows the result.

Table 3 Classification result

	Have article	Don not have article
P>=0.5	23	0
P<0.5	5	55

Total number of web pages is 83. The number of correctly classified pages is 78. The accuracy ratio is 93.98%.

From Table 3, we can also see that we can make a judgement that this web page has an article in it if P value of the page is not smaller than 0.5. And we can say a web page very likely does not have an article in it if P value of the page is smaller than 0.5.

Conclusion and Future Work

Source code of this paper including algorithm and some of the experiments is available at https://github.com/1049451037/Webpage_Text_Extraction.

The advantage of the P value algorithm is that it is concise and efficient and it does not need any prior knowledge. Besides, it can also finish two tasks (extraction and classification) concurrently.

However, this algorithm is not perfect. First, in the web page extraction, we found that it does not perform very well if there are many comments in the article on a web page, which is an important reason why the Precision is smaller than Recall (more than the main text information is extracted). Second, in the web page judgement and classification, it does not perform well when the article is very short.

Therefore, our future job will concentrate on the semantic analysis to distinguish between text area and comment area and recognize article web page even if the article is very short. We anticipate it will make our P value algorithm more accurate in both extracting and classifying.

References

[1] Chandrasekaran M, Covington M A. Heuristics to Extract the Main Text from a Captured Web Page[C]//Proceedings on the International Conference on Artificial Intelligence (ICAI). The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2012: 1.

[2] Guo Y, Tang H, Song L, et al. ECON: an approach to extract content from web news page[C]//Web Conference (APWEB), 2010 12th International Asia-Pacific. IEEE, 2010: 314-320.

[3] Weninger T, Hsu W H. Text extraction from the web via text-to-tag ratio[C]//Database and Expert Systems Application, 2008. DEXA'08. 19th International Workshop on. IEEE, 2008: 23-28.

[4] Pasternack J, Roth D. Extracting article text from the web with maximum subsequence segmentation[C]//Proceedings of the 18th international conference on World wide web. ACM, 2009: 971-980.

[5] Sun F, Song D, Liao L. Dom based content extraction via text density[C]//Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. ACM, 2011: 245-254.

[6] Zhou Z, Mashuq M. Web Content Extraction through Machine Learning [J]. 2014.

[7] Yao J, Zuo X. A Machine Learning Approach to Webpage Content Extraction [J].

[8] Sebastiani F. Machine learning in automated text categorization [J]. ACM computing surveys (CSUR), 2002, 34(1): 1-47.

[9] Knoblock C A, Lerman K, Minton S, et al. accurately and reliably extracting data from the web: A machine learning approach [M]//Intelligent exploration of the web. Physica-Verlag HD, 2003: 275-287.

[10] Alani H, Kim S, Millard D E, et al. Automatic ontology-based knowledge extraction from web documents [J]. IEEE Intelligent Systems, 2003, 18(1): 14-21.

[11] Chau M, Chen H. A machine learning approach to web page filtering using content and structure analysis [J]. Decision Support Systems, 2008, 44(2): 482-494.