Intelligent Web Robot for Content Extraction

Wenxing HONG[1,*], Jie LI[1], Weiwei WANG[1], Yang WENG[2]

(1. *Automation Department*, *Xiamen University*, *Xiamen* 361005;

2. *College of Mathematics*, *Sichuan University*, *Chengdu* 610065

* *Corresponding Author*: *hwx@xmu.edu.cn*)

Abstract: The main content of a news web page is a source of data for Natural Language Processing (NLP) and Information Retrieval (IR), which contains large quantities of valuable information. This paper proposes a method that formulates the main content extraction problem as a DOM tree node classification problem. In terms of feature extraction, we use the DOM tree node to represent HTML document and then develop multiple features by using the DOM tree node properties, such as text length, tag path, tag properties and so on. In consideration that the essence of the problem is the classification model, we use Xgboost to help select nodes. Experimental results show that the proposed approach is effective and efficient in extracting main content of new web pages, and achieves about 98% accuracy over 1083 news pages from 10 different new sites, and the average processing time per page is within 10ms.

Key words: Content Extraction; DOM; Machine Learning; Xgboost

1 Introduction

A news web page contains large quantities of valuable information, which is a data source for natural language processing (NLP) and information retrieval (IR). However, a news web page always contains ads, navigation, hyperlink lists, and so on. This extra information may not be relevant to the main content of the web and has often been shown to have negative effects on the performance of the corresponding applications. Thus, extracting the main content of a news web page in the presence of extra information is a key basis of further data analysis.

Numerous techniques have been adopted in the past two decades to deal with the problem of main content extraction. These methods are either less accurate or less efficient. Fig. 1 shows the difference between the approach developed in the present paper from the normal method. In this paper, we propose a simple and efficient approach for using machine learning. Our approach does not need to design wrappers or matching rules for each website. We use DOM (Document Object Model) as a model to represent webpages. After that we can easily filter out the leaf node, which contains text. Based on these selected leaf nodes, we extract relevant features as the inputs for machine learning. Finally, we employ Xgboost (extreme gradient boosting) to automatically learn nonlinear feature combination. Compared with the traditional machine learning methods, this model has advantages and achieves better precision, recall, and F1 scores. The complete pipeline of our approach is shown in Fig.2.

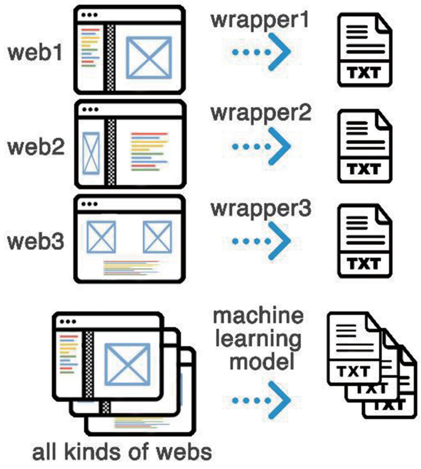


Fig. 1 Difference between the proposed approach and the normal method.

The rest of the paper is organized in the following manner. Section II examines the related work in the field. Then we describe the proposed technique in Section III. After that, we present experimental results, in Section IV. We conclude the paper in Section V.

2 Related Work

Many different news web extraction techniques have been proposed in the past two decades, such as wrappers, template detection, visual cues, statistics, and so on.

Kushmerick et al.[1] and Liu et al.[2] constructed wrappers (information extraction procedures) for semi-structured webpages. Bar-Yossef et al.[3] and Chakrabarti et al.[4] designed algorithms to detect the templates of web pages to extract the content of web pages. However, constructing wrappers or templates is a complicated task because it requires expert users to write a large number of extraction rules for the extraction process. Besides, these approaches cannot perform well on a different website that has different templates, because that depends on the template of the web page. Also, whenever a web page is changed, its wrapper or template must be rebuilt.

Deng et al.[5] proposed a new web content structure based on visual representation, which simulates how a user understands the web layout structure based on his visual perception. After that, they proposed Vision-based Page Segmentation (VIPS) algorithm, which aims to extract the semantic structure of a web page based on its visual presentation[6]. These methods may produce effective performance in some web pages, but lack generality for complex web page layouts.

Machine learning methods are currently the most used and studied, and give better output than the previous techniques. Kohlschütter et al.[7] employed machine learning (SVM) to classify the HTML blocks, Spousta et al.[8] used conditional random fields to take advantage of correlations between the labels of neighboring content blocks.

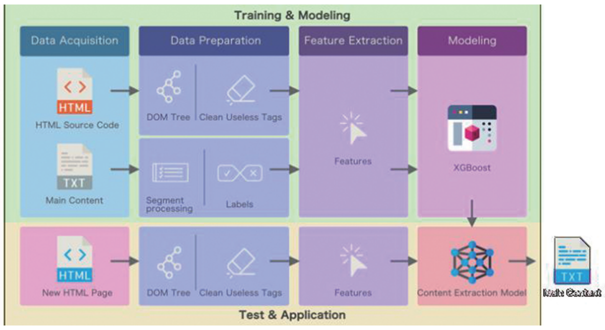


Fig. 2 The pipeline of news web content extraction.

3 Proposed Technique

Now, we discuss the steps of the proposed method.

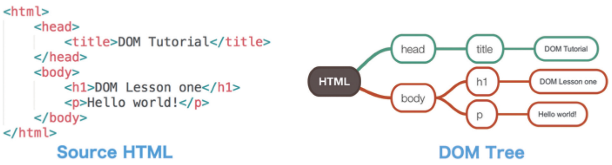
3.1 Data Preparation

The algorithm for performing data preparation is given in Algorithm 1. Input H is the raw HTML text, C is the main content, which is obtained by manual recognition, and output T is a list of nodes with features and labels.

Algorithm 1: Data Preparation Algorithm.

```
INPUT: $H$, $C$
    OUTPUT: $T$
    $H \leftarrow getDOMTree(H)$
    $H \leftarrow removeHeadTags(H)$
    $H \leftarrow removeScriptTags(H)$
    $H \leftarrow removeCommentTags(H)$
    $H \leftarrow removeStyleTags(H)$
    $H \leftarrow removeImageTags(H)$
    for all $i \leftarrow 1$ to $H.length$ do
        $t_i \leftarrow getTextFeatures(H_i)$
        $s_i \leftarrow getStructuralFeatures(H_i)$
        if$H_i.text$ in $C$ then
            $y_i \leftarrow 1$
        else
            $y_i \leftarrow 0$
        end if
        $T_i \leftarrow [t_i, s_i, y_i]$
    end for
```

a) *Pre-Processing*: Almost all web pages are written in HTML markup. Hence, we can parse HTML to DOM tree using lxml package, which is the most feature-rich and easy-to-use library for processing XML and HTML in the Python language[9]. The source HTML and DOM tree are illustrated in Fig.3.



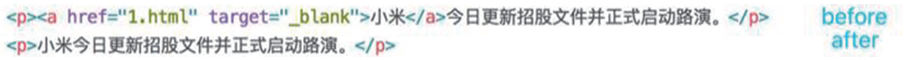
```
<html>
    <head>
        <title>DOM Tutorial</title>
    </head>
    <body>
        <h1>DOM Lesson one</h1>
        <p>Hello world!</p>
    </body>
</html>
```

Fig. 3 source HTML and DOM tree.

Many websource files at present do not correspond to the official W3C standard. Hence, we need to parse web source file into standard HTML format first, and lxml library can help us achieve this easily. After that, the useless tags with their content are removed, such as:

- inline Javascript: <script>
- style information: <style>
- head: <head>
- image: <img>
- HTML comments:<! -->

In order to align the source file with the main content, a list of tags in the paragraph are removed but its content is retained; e.g., <a>, <li>, <center>, <span>, <strong> and so on, as in Fig. 4.



```
<p><a href="1.html" target="_blank">小米</a>今日更新招股文件并正式启动路演。</p>
<p>小米今日更新招股文件并正式启动路演。</p>
```

Fig. 4 Removing tags but retaining their content.

After completing the above steps, we can get the DOM tree with initial cleaning. Then we select the DOM leaves, which contain text, and Discard the leaves that do not contain text. These DOM leaves (text nodes) are labelled in the main content or boilerplate automatically using the program based on the manually tagged main content.

b) *Feature Extraction*: Features are properties of a node that may be indicative of their being content or boilerplate. We implemented a Python program for extracting feature vectors from Web pages that have previously been parsed to DOM format. The features can be grouped into text and structural features.

Text Features. Text features capture information on the text of each node. Our framework implements the following features:

- the total length of text in the node;
- number of sentences in the node;
- the total length of space in the node;
- the number of punctuation marks in the node.

The intuition behind most of the text features is that boilerplate is likely to contain large quantities of space, longer text, fewer sentences or sentences that cannot be formed.

Structural Features. Structural features mainly provide information about the tag in a target node. Some examples of structural features are:

- depth of the target node within the DOM tree;
- the type of the last tag on the DOM tree path;
- the type of second-last tag on the DOM tree path;
- number of node that is the child node of current node's parent, with its tag type the same as the current node;
- the relative position on all web pages.

The reason for choosing these structural features is that dirty text usually has less depth of the DOM tree, the tags of the main content node are similar to each other, and has the same parent node. Besides, the main content node has a high probability of being in the middle of the web.

3.2 Training

The results of the previous two steps cannot be directly used as input for the model. We should transform non-numerical labels into numerical labels, and then standardize all features to be approximately Gaussian with zero mean and unityvariance. At this point, the training data set is ready. In terms of the training model, we present herein the findings on an impressive ensemble of tree method called Extreme Gradient Boosting (Xgboost)[10] for classification. Xgboost is short for "Extreme Gradient Boosting", which is an efficient and scalable variant of the Gradient Boosting Machine (GBM)[11]. Recently, Xgboost has been a winning tool in several Machine learning competitions[12] due to its features such as ease of use, ease of parallelization, and impressive prediction accuracy. The training set is passed to the python script to build a Xgboost model, and the optimal model parameters are determined using 5-fold cross-validation.

3.3 Content Extraction for Unseen News Webpages

Given an unseen news web page to be extracted, we first pre-process the page as in the first two-step described in this section. Afterwards, a set of vectors for the nodes are passed to the Xgboost model for prediction. The model returns the predicted class (either clean or dirty) of the node. Finally, the content of the clean node is extracted to be aggregated into the final main content.

4 Experiments

The experiments in this paper are designed to demonstrate the performance of our proposed method. We carry out experiments on a MacBook Pro (Retina, 13-inch, Early 2015) with a 2.7 GHz Intel Core i5 processor and 8 GB 1867 MHz DDR3 memory. The program is written in Python, and executed with a single thread.

4.1 Datasets

As there is no standard test set, we build a dataset with 1,083 news pages crawled with 10 online Chinese news sites. Table 1 shows some statistics about our data sets.

Table 1 Overview of Data Sets.

Domains	Pages	Nodes	Clean	Dirty
economy.caijing.com.cn	102	11782	2060	9722
news.hexun.com	103	6074	2162	3912
www.eeo.com.cn	102	7166	2437	4729
www.nbd.com.cn	146	4873	1651	3222
finance.people.com.cn	107	6230	1585	4645
new.qq.com	103	2274	2118	156
finance.sina.com.cn	108	8332	2497	5835
money.163.com	115	19425	2139	17286
www.xinhuanet.com	102	5515	750	4765
www.yicai.com	105	6423	1962	4461
Total	1083	78094	19361	58733

The raw datasets are made up of nodes from the HTML files of the websites mentioned above. The raw nodes are split from the HTML files, which are notlabelled with "clean" and "Dirty." Thus, people are asked to label every node manually depending on their understanding of the main content and the ads in the web pages. Then, the data in all the labelled nodes are passed on to the feature extraction process to get the statistic feature based on the process of Algorithm 1. The result, the output of Algorithm1, $[t_i, s_i, y_i]$, is the paradigm of processed data for machine learning.

4.2 Evaluation

To evaluate the efficiency of our method, the well-known precision, recall and F1 score measures are adapted to our use case.

Precision (P) is defined as the ratio between nodes correctly identified by the model and the whole nodes identified,

$$P = \frac{\text{Correct Nodes}}{\text{Identified Nodes}} \tag{1}$$

Recall(R) is defined as the ratio between nodes correctly identified by the model and nodes that correspond to the main content,

$$R = \frac{\text{Correct Nodes}}{\text{Actual Nodes}} \tag{2}$$

F1 score (F1) is the weighted geometric mean of precision and recall rate, and is calculated as,

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{3}$$

4.3 Results

To verify the performance of the proposed method, three sets of experiments were designed. *a*) Test the adaptability of the method against different data sources; *b*) Verify the effectiveness of the method using contrast tests with similar machine algorithms; *c*) Count the execution time per page and test the execution efficiency of this method.

a) *Different Data Sources*: In the beginning, each web-site was extracted separately, and then, we mixed all the websites. Training set and test set are divided according to the ratio of 7:3. Table 2 shows the detailed results of our proposed approach from each web site. We can see that the extraction accuracy on each site is greater than 95%, which demonstrates the performance stability of our approach. Although the number of testing pages increases with all mixed in this experiment, the approach still achieves an F-measure of greater than 97%. This demonstrates that our proposed approach can deal with a large variety of news pages.

Table 2 Experimental Result in Different Websites.

Domains	P	R	F1
economy.caijing.com.cn	97.72%	95.98%	96.84%
news.hexun.com	95.72%	97.41%	96.56%
www.eeo.com.cn	98.47%	96.74%	97.60%
www.nbd.com.cn	96.24%	98.34%	97.28%
finance.people.com.cn	97.49%	95.97%	96.72%
new.qq.com	96.93%	100.00%	98.44%
finance.sina.com.cn	96.96%	97.59%	98.44%
money.163.com	99.43%	98.77%	99.10%
www.xinhuanet.com	99.34%	97.30%	98.31%
www.yicai.com	98.13%	98.51%	98.32%
All websites mixed	97.11%	98.16%	97.63%

b) *Different Machine Learning Algorithm*: Since we regard this problem as a classification problem, we choose the three most popular classification models: SVM, Logistic Regression, and Xgboost. Table III shows the comparison results of Xgboost with the Support vector machines (SVM)[13] and Logistic Regression[14]. Support Vector Machine is useful in traditional supervised learning, especially for classification problems. Logistic Regression is a basic method in statistical learning. The data set is a mix of all the websites, and training set and test set are divided according to the ratio of 7:3. We can see that Xgboost performs better than the other two. This may be because Xgboost is an ensemble method, which uses many trees to make a decision, so it gains power by repeating itself, and it can take huge advantage in a fight by creating thousands of trees. In the other algorithms, it is difficult to choose a good kernel or parameters for the model.

Table 3 Experimental Result with Different Algorithms.

Algorithm	P	R	F1
SVM	96.32%	96.47%	96.39%
Logistic Regression	97.09%	94.56%	95.81%
Xgboost	97.11%	98.16%	97.63%

c) *Running Time*: We evaluate the running time of our approach on all web sites, which takes 95ms per Web page on average, 82ms for DOM parsing and feature extraction, and 13ms for the node classification. The average running time meets industrial standards. Fig. 5 shows the running time of each news webpage.

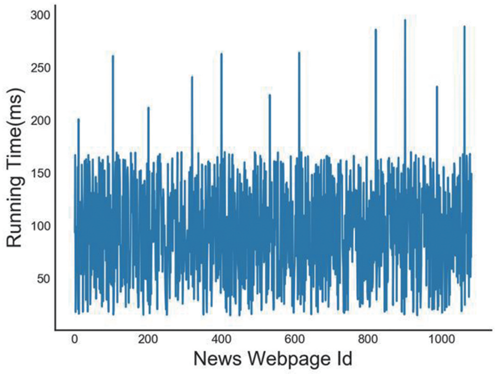


Fig. 5 Running time of each news webpage.

5 Conclusion

In this paper, we presented an approach for extracting the main content of news web pages using machine learning, which showed that a machine learning approach to the problem of main content extraction was feasible and that high classification accuracy for "clean" and "dirty" text blocks could be achieved.

The weakest points of our approach were the automatic identification of "target nodes" in the HTML tree. As future work, we plan to develop a program to help users label the main content quickly and to broaden our approach to large-scale applications.

ACKNOWLEDGMENT

This project was supported by National Key R&D Program of China (Grant No. 2018YFC0830300), Science and Technology Program of Fujian, China (Grant No. 2018H0035), Science and Technology Program of Xiamen, China (3502Z20183011) and Fund of XMU-ZhangShu FinTech Joint Lab.

References

[1] Kushmerick, N., Weld, D. S., & Doorenbos, R. (1997). *Wrapper induction for information extraction*. Washington: University of Washington, pp. 729-737.

[2] Liu, L., Pu, C., & Han, W. (2000, March). XWRAP: An XML-enabled wrapper construction system for web information sources. In:*Proceedings of 16th International Conference on Data Engineering (Cat. No. 00CB37073)*. San Diego: IEEE, pp. 611-621.

[3] Bar-Yossef, Z., & Rajagopalan, S. (2002, May). Template detection via data mining and its applications. In: *Proceedings of the 11th international conference on World Wide Web*. Honolulu: Association for Computing Machinery, pp. 580-591.

[4] Chakrabarti, D., Kumar, R., &Punera, K. (2007, May). Page-level template detection via isotonic smoothing. In: *Proceedings of the 16th international conference on World Wide Web*. Banff: Association for Computing Machinery, pp. 61-70.

[5] Cai, D., Yu, S., Wen, J. R., & Ma, W. Y. (2003, April). Extracting content structure for web pages based on visual representation. In: *Asia-Pacific Web Conference*. Xian: Springer, Berlin, Heidelberg, pp. 406-417.

[6] Cai, D., Yu, S., Wen, J. R., & Ma, W. Y. (2003).Vips: a vision-based page segmentation algorithm, [online] p.28. Available at: https://www.microsoft. com/en-us/research/publication/vips-a-vision-based-page-segmentation-algorithm [Accessed Nov. 2003\]

[7] Kohlschütter, C., Fankhauser, P., & Nejdl, W. (2010, February). Boilerplate detection using shallow text features. In: *Proceedings of the third ACM international conference on Web search and data mining*. New York: Association for Computing Machinery, pp. 441-450.

[8] Spousta, M., Marek, M., & Pecina, P. (2008, June). Victor: the web-page cleaning tool. In: *4th Web as Corpus Workshop (WAC4)-Can we beat Google*. Marrakech, pp. 12-17.

[9] Behnel, S., Faassen, M., & Bicking, I. (2005). lxml: XML and HTML with Python.

[10] Chen, T., &Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. New York: Association for Computing Machinery, pp. 785-794.

[11] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine.*Annals of statistics*, pp.1189-1232.

[12] Adam-Bourdarios, C., Cowan, G., Germain-Renaud, C., Guyon, I., Kégl, B., & Rousseau, D. (2015). The Higgs machine learning challenge. In: *Journal of Physics: Conference Series*. IOP Publishing, p. 072015.

[13] Hearst, M. A.,Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, 13 (4), pp.18-28.

[14] Harrell, F. E. (2001). Regression modeling strategies. Springer series in statistics.

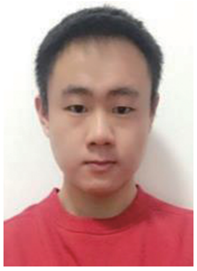
Authors' Biographies

Wenxing HONG received Ph. D. degree in System Engineering from Xiamen University, China, in 2010. He is a member of IEEE and CCF. Since 2010, he has been with the Automation Department, Xiamen University, China, where he is currently an Associate

Professor.

He is the author or co-author of more than 25 papers. He has led and participated in more than 15 research projects and funds, including National Natural Science Foundation of China. His current research interests are in the areas of data mining, big data, artificial intelligence, recommendation system, and FinTech. He is the Dean of the Research Center for Systems and Control, Xiamen University. He has served as the General Secretary of the *International Conference on Computer Science and Education* (*ICCSE*) and Fujian Systems Engineering Society, in 2006 and 2010, respectively.

E-mail: hwx@xmu.edu.cn



Jie LI, is a graduate student in the Department of Automation, Yanshan University, China. He obtained his Bachelor of Engineering from Yanshan University, China. His current research interests at the System and Control Center Laboratory, Xiamen University, are big data and cloud computing.

E-mail:812126839qq@gmail.com

Weiwei WANG, is a graduate student at the Department of Automation, Xiamen University, China. He obtained his Bachelor of Engineering from Xiamen University, China. His current research interest at the System and Control Center Laboratory, Xiamen University is financial text mining, such as stock price movements prediction.

E-mail:www@stu.xmu.edu.cn

Yang WENG received his B.S. and Ph.D. degrees from the Mathematics Department, Sichuan University, Chengdu, China, in 2001 and 2006, respectively. Since 2006, He has been with College of Mathematics, Sichuan University, where he is currently an Associate Professor. He was a Postdoctoral Fellow at the Nanyang Technological University, Singapore, from August 2008 to July 2010. His current research interests include statistical machine learning and nonparametric Bayesian inference.

E-mail:wengyang@scu.edu.cn