Web Page Content Extraction Based on Multi-feature Fusion

Bowen Yu, Junping Du, Yingxia Shao

(Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, School of Computer Science, Beijing 100876)

Corresponding Author: Junping Du, E-mail: junpingdu@126.com

Abstract With the rapid development of Internet technology, people have more and more access to a variety of web page resources. At the same time, the current rapid development of deep learning technology is often inseparable from the huge amount of Web data resources. On the other hand, NLP is also an important part of data processing technology, such as web page data extraction. At present, the extraction technology of web page text mainly uses a single heuristic function or strategy, and most of them need to determine the threshold manually. With the rapid growth of the number and types of web resources, there are still problems to be solved when using a single strategy to extract the text information of different pages. This paper proposes a web page text extraction algorithm based on multi-feature fusion. According to the text information characteristics of web resources, DOM nodes are used as the extraction unit to design multiple statistical features, and high-order features are designed according to heuristic strategies. This method establishes a small neural network, takes multiple features of DOM nodes as input, predicts whether the nodes contain text information, makes full use of different statistical information and extraction strategies, and adapts to more types of pages. Experimental results show that this method has a good ability of web page text extraction and avoids the problem of manually determining the threshold.

Key words: web content extraction; neural network; DOM tree; heuristic strategy; classification

1 Introduction

Web page resources are an important part of the information society and contain a huge amount of high-value data resources [1]. At the same time, the rapid growth of web resources and the huge amount of data also support the rapid development of deep learning technology [2]. Therefore, the information extraction technology [3][4] for webpage body has good practical value. For example, as the data processing part of the NLP [5][6] pipeline operation, the text data is extracted from the complex page structure, which is convenient for the use of subsequent model algorithms [7][8]. At the same time, web page text extraction is also the technical cornerstone of browser reading mode and other functions [9].

In the past research, the method of using a single heuristic strategy often needs to manually determine the threshold. With the rapid growth of the number and types of Web pages, it is not flexible to use one strategy to extract page body information from different sources [10]. Therefore, this paper proposes a web page text extraction algorithm based on multi-feature fusion.

The main contributions of this paper include four aspects:

1) We propose a web page text extraction method based on multi-feature fusion, which parses HTML into a DOM tree, judges and extracts each node, and effectively obtains text information;

2) Taking the DOM tree node as the unit, according to the characteristics of the web page text information, design a variety of statistical features, and design high-level features based on the heuristic strategy of text density;

3) Design and build a neural network classifier, take multiple features of the DOM node as input, predict the node's category and extract (text node/noise node), and avoid manual selection of thresholds through supervised learning;

4) The experimental results show that the proposed
multi-feature-based web page text extraction algorithm has good extraction ability.

2 Related work

Web resources has good research value and practical significance [11][12][13]. In general, the text extraction of web pages is to extract the text information of the main body by removing the noise information in the input page, such as: navigation bar, page header, advertisement column, etc., which has important research value and practical significance. In the early days, there were some extraction algorithms such as the MSS algorithm [14]. This algorithm converts the HTML page into a one-dimensional sequence, and then assigns a value to the token in the sequence, such as giving a negative value to the label token and a positive value to the text token, and then calculates the maximum score in the sequence [15]. And the word sequence with the longest length is extracted as the text [16][17]. This method needs to manually assign a specific value to different token types. The difference of the value will affect the extraction effect, which is not very good in actual performance.

In 2003, Microsoft Research Asia proposed a visual representation-based web content structure analysis method - VIPS algorithm [18], combined with the DOM tree to process the page [19][20]. On the basis of this algorithm, there is an algorithm combining the improved Hidden Markov Model to realize Web information extraction [21]. The webpage text extraction algorithm based on the VIPS algorithm has better performance when facing webpages with a single visual form and a large difference in the code structure level. However, such algorithms require a page to be completely rendered before analysis and extraction can be performed, which consumes a large amount of resources [22].

Another idea of page text extraction is template-based extraction algorithm [23][24]. The core idea of this kind of algorithm is to assume that web pages are constructed using the same or similar templates, so the same or similar parts between pages are non-text, while pages The part with a large difference is the main text [25][26]. In practice, it can also be combined with the URL to determine whether the web pages to be extracted have the same structure. However, such algorithms are more suitable for modeling a single data source to achieve content extraction. When faced with multiple data sources, it is very inflexible. If the extraction code is written separately for each data source, it is labor-intensive, and once the website of the target data source is updated or revised, the old algorithm needs to be re-modified.

At present, the best and widely used methods of extraction are generally to use the information of HTML pages to design heuristic strategies for extraction, these heuristic strategies include: text density[27], synthetic text density, label ratio[28], path ratio [29] and so on. The paper [30] defines text density as the ratio of all words within a label to the number of all labels. The paper [31] proposes an entropy-based information content density algorithm. The paper [32] proposes a paragraph extractor to cluster HTML paragraph tags and local parent titles to identify main content in news articles. Such algorithms generally use heuristic functions to calculate and score label nodes as the extraction criteria. However, such algorithms often use a single strategy and need to manually determine the threshold, which is not highly adaptable in the face of multiple data sources. The paper [33] proposes an adaptive extraction algorithm based on decision tree to solve this problem, extracts features from each node of the DOM tree, and uses the decision tree algorithm for binary classification judgment. However, this method has many pre-assumptions, such as Only leaf nodes are considered, and only fewer HTML tags are considered.

3. Web page text extraction algorithm based on multi-feature fusion

In this section, we mainly introduce the webpage text extraction algorithm based on multi-feature fusion. The label node of the DOM tree is used as the calculation unit. By designing and extracting various statistical features and heuristic high-order features of the node, combined with the neural network classifier, predict nodes containing textual information and extract them. The method in this paper solves the problem of adaptability of a single strategy to different data source web pages by fusing multiple features, and at the same time avoids the difficulty of manually determining the threshold through supervised learning. This method can be divided into several important steps:

1. Preprocess the input HTML page, delete CSS
and Development

2. Traverse the DOM tree, perform feature extraction on each label node, and obtain its feature vector representation.

3. Input the feature vector of each node into the trained neural network to obtain its output prediction, and extract the node information whose prediction result is the text node.

The algorithm flowchart is shown in Figure (1):

![Algorithm flowchart](image)

### 3.1 Page preprocessing

First, the input of the algorithm is an HTML page, and this paper first preprocesses the input. From practical experience, the CSS format of the page and the code script have no substantial impact on the text information, so the tag content such as `<script>` and `<style>` can be removed in the preprocessing stage, and then the cleaned HTML data can be converted into DOM objects. DOM is a document object model (DOM), which represents a document as a tree structure. Each node of the tree represents an HTML tag or a text item within the tag. By representing the input data as a DOM tree structure, it is convenient for us to extract data features in units of label nodes.

### 3.2 Node feature representation and extraction

#### 3.2.1 Label Semantics

Compared with image or natural language data, HTML language itself is a kind of data with semantic information and structure. HTML provides numerous tags, each with a clear function and semantics, such as `<p>` for text paragraphs, `<a>` for hyperlinks, `<ol>`, `<li>` for list formatting, tags `<div>` is often used in typography. From the actual development experience, although there are page codes that do not comply with the specification, on the whole, there is a relatively clear correlation between the label and its internal information. Therefore, this paper takes the label semantics of nodes as an important feature of nodes, and uses mutual information as a measure of correlation. The calculation formula of mutual information is as formula (1):

$$MI(t, c_i) = \sum_{i=1}^{n} P(c_i) \log \frac{P(t, c_i)}{P(t)P(c_i)}$$

Among them, it $t$ represents the label and $c_i$ represents the category. There are two categories in this paper: the text node and the noise node; it represents $MI$ the mutual information between the $P(t, c_i)$ tag $t$ and the category $c_i$, and the joint distribution $c_i$ represents the probability that the tag $t$ belongs to the category $c_i$, the probability of the $P(t)$ tag $t$, and the probability of the $P(c_i)$ category $c_i$.

Mutual information is a measure of the degree of interdependence between random variables, which can be used to measure the statistical correlation between features and categories. This paper collects policy pages of policy portal websites such as the Ministry of Industry and Information Technology and the Ministry of Science and Technology, as well as news pages such as Phoenix.com, and annotates more than 14,000 label nodes, which are divided into two categories: text nodes and noise nodes, and some labels and mutual information The amount is shown in Table 1:

| Label | Number of text nodes | Number of noisy nodes | Mutual information |
|-------|----------------------|-----------------------|--------------------|
| `<p>` | 7969                 | 22866                 | 0.188              |
| `<a>` | 496                  | 55500                 | 0.158              |
| `<li>` | 195                  | 36979                 | 0.117              |
| `<span>` | 983                  | 9319                  | 0.085              |

By calculating mutual information, it can be found that there is a correlation between tags and categories, so HTML tags are used as a feature component of nodes. Since HTML supports a large number of tags, and the common tags in actual use are only a small part, in the feature representation, more than ten tags such as `<p>`, `<a>`, `<li>`, `<h1>` are supported.其余标签修改为< unk >表示。

#### 3.2.2 Node Statistical Characteristics

In addition to label semantic features, a variety of statistical features are also fused. The statistical features used in this paper include:
The number of labels under the subtree represented by the node \(i\).

\(TL_i\) The number of linked labels under the subtree represented by the node \(i\).

\(TC_i\) The length of the string under the subtree represented by the node \(i\).

\(TLC_i\) The length of the linked string under the subtree represented by the node \(i\).

\(TP_i\) The number of punctuation marks under the subtree represented by the node \(i\).

By observing the characteristics of the page text information, it can be found that these statistical characteristics have certain heuristic significance, and are often used as function items of heuristic strategies. For example, advertisements or navigation panels are often hyperlink texts, which do not belong to body information, and their \(TLC_i\) values are often larger, while body nodes often contain long unformatted text strings, and their \(TC_i\) feature values are often larger.

By extracting the statistical features of nodes and combining with neural network training, the different feature patterns between text nodes and noise nodes can be better learned.

### 3.2.3 Heuristic Higher-Order Features

According to the universal approximation theorem of neural networks, more complex features and functions can theoretically be fitted according to the provided statistical features. However, due to the lack of data sets available for training in the field of information extraction, this paper constructs a method for training and a dataset of labeled nodes for testing. Due to the relatively small amount of data, based on the existing statistical features, this paper designs and uses heuristic high-order features to help model training.

The heuristic strategy used in this paper is the more popular and mature text density strategy, by calculating the text density value of each node as a feature component of the node. The text density calculation formula is:

\[
TD(i) = \frac{TC_i - TLC_i}{T_i - TL_i}
\]

where, \(TD(i)\) the text density value of the node \(i\). The text density strategy is based on the observation that a text node tends to contain longer plain text strings, while a noisy node tends to contain more linked strings, or more linked child nodes.

Therefore, when the \(TD\) value of a node is larger, it means that the node has more unformatted and unlinked information, less formatted and linked information, and is more likely to contain textual information.

Since the DOM representation structure of the page is a tree, the depth-first strategy can be used to recursively extract the feature representation of each node. The specific algorithm is shown in Table 2:

| Table 2: Feature Extraction Algorithms of Label Nodes |
|-----------------------------------------------|
| **Feature Extraction Algorithm for Label Nodes** |
| **Input:** DOM tag node |
| **Output:** feature set of label nodes |
| **nodeFeature (node):** |
| TC = TLC = T = TL = TD = 0 |
| if node is null: |
| return \{TC, TLC, T, TL, TD\} |
| for each child in node: |
| childInfo = nodeFeature (child) |
| update( TC, TLC, T, TL) <- childInfo |
| if node is <a>: |
| TL++ |
| TLC += node.txtLength |
| else: |
| T++ |
| TC += node.txtLength |
| TD = (TC - TLC) / (T - TL) |
| return \{TC, TLC, T, TL, TD, node.tag\} |

### 3.3 Neural Network Classifier

As shown in Figure 3, after obtaining the label semantic features, statistical features and heuristic features of a node, a classifier model can be combined to predict whether the node is a text node. This paper uses a fully connected neural network with two hidden layers. The network accepts a 18-dimensional node feature vector as input, which uses one-hot one-hot encoding for label semantic features. Use Relu as the activation function in the hidden layer to achieve nonlinear fitting,
formula (3): \[
\sigma(x) = \max(0,x)
\] (3)

There are two nodes in the output layer, which are normalized using Softmax, formula (4):

\[
\text{Softmax}(o_i) = \frac{\exp(o_i)}{\sum_j \exp(o_j)}
\] (4)

The model loss function uses the cross entropy loss, formula (5):

\[
l(y, \hat{y}) = \sum_{i=1}^{q} y_i \log \hat{y}_i
\] (5)

where \(y\) represents the true label, \(\hat{y}\) represents the predicted label, and \(q\) represents the encoding length of the label.

3 Experimental results

Due to the lack of public data sets in the field of text extraction, this paper writes a Python program to crawl and collect three different types of pages: news, policies and blogs. The data sources include the Ministry of Industry and Information Technology, the Ministry of Science and Technology, the Ministry of Commerce, Phoenix.com, Blog Park, etc. A data source to test the stability of the algorithm when faced with different data sources and page types.

After collecting pages, the nodes are divided into text nodes and noise nodes by manual annotation. The total number of labeled node samples exceeds 100,000, and the ratio of positive and negative samples is 1:3.2. Since only the content under the <body> tag in the HTML page is the display part of the page, the label node of each page is limited to the subtree part corresponding to the <body> tag.

The algorithm in this paper is a supervised algorithm. In the experiment, the collected data set is divided into training set, cross-validation set and test set according to the ratio of 6:2:2.

Evaluation metrics include precision, recall, and F1 score. Due to the lack of public open source code, this paper uses Python to implement two contrasting algorithms: the MSS algorithm and the text density-based extraction algorithm (TDEX).

MSS algorithm converts the HTML page into a sequence of tokens, assigns different types of tokens a score, and extracts the longest word sequence with the largest score as the text. In the implementation of this article, the label token is assigned a value of -3, and the character token is assigned a value of 1. In addition, in the text density-based extraction algorithm (TDEX), the implementation of this paper uses the heuristic features of Section 2.1.3 as the scoring function, and determines the score threshold by randomly sampling the test set. The two comparison algorithms are also compared on the test set data, and the experimental results are shown in Table 2:

| Algorithm | accuracy | recall | F1    |
|-----------|----------|--------|-------|
| MSS       | 82.13%   | 71.05% | 76.19%|
| TDEX      | 91.12%   | 93.10% | 92.10%|
| Algorithm | 95.88%   | 97.44% | 96.65%|

The experimental results show that the proposed method has the best performance in terms of precision, recall and F1 score. Through the analysis of the specific examples of the test set, the MSS algorithm is often difficult to deal with the pages with noise in the text.
because of extracting the subsequence with the longest maximum score, so the recall rate is significantly lower. Compared with the extraction method that simply uses text density, the neural network that integrates multiple features can better adapt to different page types, and at the same time, through the learning of the training set, it avoids manual selection of thresholds and has the best performance.

4 Conclusion

This paper proposes a web page text extraction algorithm based on multi-feature fusion. By representing HTML pages as a DOM tree, extracting and fusing the tag semantic features, statistical features and heuristic features of each node, it has good web page text extraction capabilities. At the same time, the influence of manual selection of thresholds is avoided. At the same time, it was found in the research that the main research methods in this field are unsupervised algorithms or heuristic algorithms, and there is a lack of open and sufficient data sets for training and evaluation. In the future, we will further explore the labeling rules and collection of page data sets. Explore more feature fusions and more complex network structures on larger data volumes.

references

[1] Soderland S. Learning to Extract Text-Based Information from the World Wide Web[C]//SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’97), 1997: 251-254.
[2] Xue Z, Du J, Du D, et al. Deep low-rank subspace ensemble for multi-view clustering[J]. Information Sciences, 2019, 482: 210-227.
[3] Xu L, Du J, Li Q. Image fusion based on nonsubsampled contourlet transform and saliency-motivated pulse coupled neural networks[J]. Mathematical Problems in Engineering, 2013.
[4] Piskorski J and Yangarber R. Information extraction: Past, present and future[C]//Multi-source, multilingual information extraction and summarization, 2013: 23-49.
[5] Shi C, Han X, Song Li, et al. Deep collaborative filtering with multi-aspect information in heterogeneous networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 33(4): 1413-1425.
[6] Yang Y, Du J, and Ping Y. Ontology-based intelligent information retrieval system[J]. Journal of Software, 2015, 26(7): 1675-1687.
[7] Xue Y, Hu Y, Xin G, et al. Web page title extraction and its application[J]. Information processing & management, 2007, 43(5): 1332.
[8] Hu W, Gao J, Li B, et al. Anomaly detection using local kernel density estimation and context-based regression[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 32(2): 218-223.
[9] Gupta S, Kaiser G, Neistadt D, et al. DOM-based content extraction of HTML documents[C]//Proceedings of the 12th international conference on World Wide Web, 2003: 207-214.
[10] Kou F, Du J, He Y, et al. Social network search based on semantic analysis and learning[J]. CAAI Transactions on Intelligence Technology, 2016: 1(4): 293-302.
[11] Shah P, Pandit H B A Review: Web Content Mining Techniques[J]. Data Engineering for Smart Systems, 2022: 159-172.
[12] Pujar M, Mundada M R A Systematic Review Web Content Mining Tools and its Applications[J]. International Journal of Advanced Computer Science and Applications(IJACSA), 2021, 12(8): 752-759.
[13] Li A, Du J, Kou F, et al. Scientific and Technological Information Oriented Semantics-adversarial and Media-adversarial Cross-media Retrieval[J]. arXiv preprint arXiv:2203.08615, 2022.
[14] Pastermack J and Roth D. Extracting article text from the web with maximum subsequence segmentation[C]//Proceedings of the 18th international conference on World wide web (WWW ’09). 2009: 971–980.
[15] Zhao H, Liu Q, Zhu H, et al. A sequential approach to market state modeling and analysis in online p2p lending[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017, 48(1): 21-33.
[16] Zachariasova M, Kamenycz P, Hudec R, et al. A Novel Imaging Approach of Web Documents Based on Semantic Inclusion of Textual and Non-Textual Information[J]. AASRI Procedia, 2014, 9: 31-36.
[17] Li Q, Du J, Song F, et al. Region-based multi-focus image fusion using the local spatial frequency[C]//2013 25th Chinese control and decision conference (CCDC), 2013: 3792-3796.
[18] Cai D, Yu S, Wen JR, et al. Vips : a vision-based page segmentation algorithm[J]. 2003.
[19] Lv F. Research on Similarity Computing Technology of Fishing Web Pages Based on Visual Features [D]. Harbin Institute of Technology, 2015.
[20] Shen Y. Research on automatic summarization technology of Chinese web pages based on visual features and text structure analysis [D]. East China Normal University, 2014.
[21] Li W, Li S, Jing X, et al. Web Information Extraction Based on Simulated Annealing Algorithm and Second-Order HMM[J]. Computer Engineering and Design, 2014, 35(4): 1264-1268.
[22] Meng D, Jia Y, Du J, et al. Stability Analysis of Continuous-time Iterative Learning Control Systems with Multiple State Delays[J]. Acta Automatica Sinica, 2010, 36(5): 696-703.
[23] Bar-Yossef Z, Rajagopalan S. Template detection via data mining and its applications[C]//Proceedings of the 11th international conference on World Wide Web. 2002: 580-591.
[24] Li W, Xiao X, Liu J, et al. Leveraging graph to improve abstractive multi-document summarization[J]. arXiv preprint arXiv:2005.10043, 2020.
[25] Yang Y. Research on Webpage Information Extraction Technology Based on
[26] Gu Y, Gao G, Gao B, et al. Research on DeepWeb Information Extraction Based on Template and Domain Ontology [J]. Computer Engineering and Design, 2014, 35(1): 327-332.

[27] Sun F, Song D, and 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, 2011: 245-254.

[28] Weninger T, Hsu W, and Han J. CETR: content extraction via tag ratios[C]/Proceedings of the 19th international conference on World wide web, 2010: 971-980.

[29] Wu G, Li L, Hu X, et al. Web news extraction via path ratios[C]/Proceedings of the 22nd ACM international conference on Information & Knowledge Management. 2013: 2059-2068.

[30] Song D, Sun F, Liao L. A hybrid approach for content extraction with text density and visual importance of DOM nodes[J]. Knowledge and Information Systems, 2015: 42(1): 75-96.

[31] Annam M, Sajeev G P. Entropy based informative content density approach for efficient web content extraction[C]/2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI). 2016: 118-124.

[32] Carey H, Manic M. HTML web content extraction using paragraph tags[C]/2016 IEEE 25th International Symposium on Industrial Electronics (ISIE).2016: 1099-1105.

[33] Lv R, Liu J. Adaptive Webpage Text Extraction Based on Decision Tree[J]. Modern Computer (Professional Edition), 2019.

Bowen Yu, (1995-), male, master, member of CCF. His main research areas are natural language processing and data mining.

Junping Du (corresponding author), born in 1963. She is currently a professor and doctoral supervisor at the School of Computer Science and Technology, Beijing University of Posts and Telecommunications. Her main research interests include artificial intelligence, machine learning and pattern recognition.

Yingxia Shao, (1988-), male, associate professor, senior member of CCF. The main research areas are large-scale graph analysis, parallel computing framework and knowledge graph analysis.