Hindawi
Security and Communication Networks
Volume 2021, Article ID 2470897, 11 pages
https://doi.org/10.1155/2021/2470897

Research Article

Hierarchical Contaminated Web Page Classification Based on Meta Tag Denoising Disposal

Xiang Song,[1,2] **Yi Zhu**,[2] **Xuemei Zeng**,[2] **and Xingshu Chen**[1,2]

[1]*Cyber Science Research Institute, Sichuan University, Chengdu 610065, China*
[2]*School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China*

Correspondence should be addressed to Xuemei Zeng; zengxm@scu.edu.cn

Received 27 August 2021; Accepted 23 October 2021; Published 16 November 2021

Academic Editor: Habib Ullah Khan

Web page classification is critical for information retrieval. Most web page classification methods have the following two faults: (1) need to analyze based on the overall web page and (2) do not pay enough attention to the existence of noise information inside the web page, which will thus decrease the efficiency and classification performance, especially when classifying the contaminated web page. To solve these problems, this paper proposes a denoising disposal algorithm. We choose the top-down method for hierarchical classification to improve the prediction efficiency. The experimental results demonstrate that our method is about 7 times faster than the full-page method and achieves good classification results in most categories. The precision of 7 parent categories is all above 88% and is 24% higher than the other meta tag-based method on average.

1. Introduction

Web page topic classification is critical for website management and information retrieval. Organizing page categories makes it more convenient to search for web pages [1]. While dealing with web page topic classification, most existing methods need to extract information related to the topic from the entire source code.

However, as the amount of data increases, the methods depending on the web source code are facing some problems. (1) More time and computing resources are needed for processing long texts in the source code of the web page. (2) Existing meta tag-based methods often ignore the importance of denoising disposal, which may lead to a decline in data quality. Web pages are vulnerable to attacks such as web page tampering. Therefore, the web page meta tag text probably contains noise information unrelated to the web page's theme. Noise information usually contains some advertisements or information including pornography, gambling, and drugs. (3) Web page categories organized with a hierarchy may contain a large number of categories, making it more difficult to accurately classify the categories. As organizing Web pages with hierarchical categories can meet the classification needs of different granularity, many websites organize categories into a hierarchical structure, which makes this problem more common and practical.

Due to the special characteristics of hierarchical classification, the error that appears in the high level can cause error in low level, which generally be called the error propagation problem, so the hierarchical classification requires better classification accuracy at high level. Moreover, the number of samples of different classes in hierarchical classification is often unbalanced. The category with fewer samples may perform worse if noise information affects the classification, deepening the problem of data imbalance.

The problems above make it more challenging to classify the web pages. To solve those problems, this paper focuses on classifying the web pages using meta tag text including the Description and the Title, to save computing and storage resources. First, this paper explains the feasibility of using the Title and the Description tags to classify websites. Besides, to address the problem of polluted text, we present an algorithm based on Deterministic Finite Automata (DFA) to quickly detect and clean noise in the meta tag. Finally, this paper chooses the deep learning method for text classification and the top-down method for hierarchical

classification. Classification experiments are carried out in the data set proposed in this paper, and the results show that our method is effective and less affected by data imbalance.

The contributions are shown as follows:

(1) A method for topic classification using the meta tag text tags on the web page and deep learning method is proposed, and it has proven the feasibility and effectiveness.

(2) An algorithm to perform data cleaning and extract effective tag information is designed. To get rid of the "pollution" of the data set, we build a data cleaning algorithm to cope with the pollution information in the tag.

(3) Manually annotated web page category data set containing two-level categories is published, including 7 primary categories and 41 secondary categories. The data set contains a total of 64747 items of data.

The rest of this paper is organized as follows. In Section 2, the mainstream web page classification methods will be summarized. Section 3 introduces the structure of the classification process. Section 4 introduces the data cleaning method of the text noise evaluation algorithm. Then in Section 5, we carry out classification experiments on real data sets to verify the feasibility and effectiveness of the proposed method. Finally, Section 6 concludes this paper with a prospect of future research.

2. Related Works

In this section, we present relevant existing studies in web page classification, particularly in web page information extracting and hierarchical multiclass classification.

2.1. Web Page Information Extracting. Due to semi-structured web data and no uniform layout style. Many existing web page classification methods will need to crawl the entire page and then perform information extraction and classification based on crawled data. There are several methods for web page information extraction.

One type of method is to extract the structural characteristics and text characteristics of the website at the same time. Zhang [2] improved the weight of feature items based on the structural features of the web page to express the web page text. Combined with the improved SVM algorithm, an automatic web page classification system is established. Deng et al. [1] integrated the text features in the key tags of the web page and the structural features of HTML web pages. Based on confidence comparison and other strategies to optimize the reliability of the classification results of combined classifiers, a web page classifier based on fusion features and combined classifiers is proposed, and the effectiveness of the method is verified in the public data set. Do et al. [3] studied the detection and classification of phishing web pages by extracting the structural features of web pages and using four different deep learning algorithms, deep neural network (DNN), convolutional neural network (CNN), short-term memory (LSTM), and gate cycle unit (GRU), to evaluate the effectiveness of deep learning model in phishing web page detection.

Another type of method is to extract the text characteristics of the web page. Some researchers choose to extract text features based on the overall source code of the web page, and some researchers choose to use the text under the specific tag of the web page for feature extraction. Web page data is often organized in the form of a DOM (document object model) tree. A page contains many tags. Each tag is an element node in the DOM tree. The tag contains the text related to the web page. According to the length of input text, it can be divided into web page method based on long text and web page method based on short text.

Many methods use the web page method based on long text. Gupta and Bhatia [4] proposed an ensemble method for web page classification, bidirectional Bert model is used to learn contextual representation from the long text source code of web pages, and the parallel multiscale semantics is used to fine-tune the target task. In the experiment, it shows better results than other transfer learning methods. Cheng [5] established a word vector based on word2Vec training based on the feature words corresponding to each type of web page topic. A central word vector is established according to the word vectors corresponding to multiple feature words, and the text similarity is calculated combined with the DBSCAN clustering algorithm to take the topic with the highest similarity as the text classification result, which verified the effectiveness of the method in the experimental environment. Zhang [6] set filtering rules to filter the label text when crawling the web page source code and combined with the improved SVM algorithm to realize a large-scale website classification system. Luo and Wang [7] set different weights for the text in labels such as <body>, <a>, <title>, and combined with in-depth learning to classify web pages. These methods have achieved good results in experiments. However, these methods need to deal with the whole web page or web page tags containing long text, such as <body>, which will lead to large consumption of computing and storage resources when the amount of data is large enough and are not suitable for scenes requiring high classification efficiency.

Much web page classification research also adopts the method based on short text. The three meta tags of web pages <title>, <description>, and <keywords> are usually highly related to the theme of web pages. At the same time, the length of text content is shorter than that of backbone tags such as <body>. Meta tags are also a common research object of short text classification. Ebubekir and Banu [8] extracted <description> and <keywords> tags, used 5-layer RNN for web page classification, and explored the impact of transfer learning on the web page classification effect. The classification method based on short text can save the relevant information about the website theme to the greatest extent on the premise of consuming less storage space, so as to reduce the amount of stored information, the cost of extracting information, and the difficulty of filtering irrelevant information. The method based on short text also has the advantage of efficiency. It is more suitable for scenes requiring high classification efficiency.

In addition, there are also research methods to extract features from the reference relationship between web pages. Due to the need for multiple web page access requests to construct the web page reference relationship, this type of method has a certain problem of network access efficiency and is not suitable for scenes requiring high classification efficiency.

2.2. Hierarchical Multiclass Classification. While performing the hierarchical classification of information, the category structure is usually organized into tree or Directed Acyclic Graph. This paper mainly studies hierarchical classification based on the tree structure. In this form of category organization, the classification problem is transformed into a multiclass classification problem with a hierarchical structure. There are many methods to solve the multiclassification problem based on a hierarchical structure, and there is no recognized data set. The related research often focuses on scenarios in different fields. Due to the different data forms and characteristics in different scenarios, the solutions are often different.

At present, the methods of tree-structured hierarchical multiclassification can be divided into three categories: bottom-up method, top-down method, and search classification method. The advantage of the bottom-up method is that the classification process is simple, and only one classification model will be generated during the classification. In contrast, the disadvantage is that the hierarchical structure is ignored, and the probability of nonleaf nodes is not considered. Search classification is a method of searching first and then classifying. First, all paths of the hierarchical tree are searched according to data to find candidate categories. Then, the candidate categories are classified according to the classifiers of candidate categories. Among them, [9, 10] adopted the method of search classification.

The top-down approach method is proposed to decompose the hierarchical structure tree into small-scale local classification problems according to the category hierarchy. After that, local training is carried out in the category of each node, and the categories are classified from top to bottom. Compared with the bottom-up method, many classification models are generated, but the hierarchical structure is better used, and the relationship between parent category and subcategory is considered. The research on the top-down method is very popular. Banerjee et al. [11] used transfer learning to initialize the subcategory classifier with the parameters learned from the parent class, which makes good use of hierarchical structure in essence. Naik and Rangwala [12] optimized the top-down method by modifying the hierarchy. Oh [10] proposed reducing the fault-tolerant rate of top-down classification by modifying the loss function, to minimize the cost of TOP-K classification in each prediction error category. Literature [12] and [13] tried to improve the hierarchical category structure defined by experts by designing an algorithm to adjust the hierarchical structure, so as to improve the classification effect.

3. Classification Framework

Our classification framework contains two main steps: first, we perform preprocessing, especially using an algorithm to clean noise in the sample. Then, we use the deep learning classification method to classify each sample.

3.1. Classification Input and Data Cleaning. For classification input, tags with a short text on the web page are chosen. Among various kinds of tags, tags such as the 'Title' and the 'Description' usually have a stronger correlation with the theme of the web page, and the text content length is relatively shorter than most tags. If these short text tags for classification tasks can be proven useful, the amount of information to crawl and the cost of extracting information can be reduced. Different from the text processing method, which uses the entire page as input, our method only uses the text content under the specified meta tag. Therefore, the method proposed in this paper simplifies the difficulty of preprocessing and improves the efficiency of the procedure.

In this paper, we choose the text content in the 'Description' and the 'Title' tags to perform the web page classification. However, the text in the 'Description' and the 'Title' tags may contain noise text. Noise text refers to the text that contains information unrelated to the web page's topic, such as pornography and gambling. This information will affect the accuracy of classification. Detecting and processing for the noise information will improve the classification effect. To achieve this goal, we have designed a set of noise-cleaning processes, and the details will be introduced in Section 4.

3.2. Classification Algorithm. The deep learning-based method was chosen for the experiment because it does not need artificial feature extraction and performs well in many classification tasks. In our experiments, three typical deep learning methods, TextCNN [14], LSTM, and Bi-LSTM [15], were compared to select the better model.

The top-down approach, bottom-up approach, and search classification approach are three typical approaches for hierarchical classification. We choose the top-down approach because it considers the hierarchy compared with the bottom-up approach and is also more efficient than the search classification method. When the hierarchy tree height in the current classification scenario is limited, the negative impact of the wrong propagation is reduced, ensuring the utilization of the hierarchy and prediction efficiency. We select the hard decision-making algorithm of the top-down algorithm to perform hierarchical classification. The algorithm selects the category with the highest probability as the first classification results at the top level. Then, the algorithm classifies the subcategories of the category, choosing the category in the subcategories with the highest probability as the second-level classification result. The module chooses a method to make hierarchical path selections for the hierarchical classification to improve classification speed.

4. Text Noise Process

4.1. Introduction about Deterministic Finite Automaton. In real cases, web pages may be injected with some noise text unrelated to the page's topic. Based on data observation, we find a "local pollution" phenomenon in the web page source code. It indicates that even web pages with noise information tags are likely to have some uncontaminated tags. For example, if the Description tag is injected with noise text, the Title tag may not be injected at the same time. If the web page sample with only a few noisy text tags is directly discarded, the normal text in the sample will be ignored, thus causing a waste of data. Therefore, this paper designs an algorithm to identify noisy text tags, using the uncontaminated tags of the web page as an alternative input for the following classification tasks when noisy text tags are detected.

Our noise process method needs to identify noisy text before cleaning. To achieve the goal, we select a scheme based on the sensitive word dictionary to conduct matching processing to detect whether the text in the tag is noise information.

Deterministic Finite Automaton (DFA) algorithm is a commonly used algorithm in text-matching against dictionary. It is a sensitive word filtering algorithm with high matching efficiency. Figure 1 shows the form of saving filter words in DFA. DFA records the word prefix, which means the method does not need to record the same prefix for different words repeatedly. It adds a final state flag of the character at the end of the word. The end character with the final sign is marked with a circle in Figure 1 to decide whether the match was successful. The core of the algorithm is establishing a tree based on sensitive words to identify sensitive words. The DFA algorithm has been widely used. Najam et al. [16] modified the DFA algorithm for DPI (deep packet detection) field. Xue and Wuxur [17] and other use of variants of the DFA algorithm to filter out sensitive information. The complexity of many existing string-matching algorithms is directly proportional to the number and word's length in the dictionary. With the expansion of the scale of sensitive words, the matching process will cost a lot of time. In contrast, the complexity of the DFA algorithm is only related to the length of the sensitive word and has nothing to do with the size of the sensitive words dictionary. It will not reduce the matching efficiency greatly with the increase of sensitive words.

4.2. Introduction about the Text Abnormal Rate Evaluation Algorithm. In this paper, an algorithm to identify the abnormal text is proposed. We use the DFA algorithm to perform text matching and record the matched noise word number ratio. We call the ratio a text abnormal degree.

Algorithm 1 shows the algorithm for calculating text abnormal degree. During the matching process, the algorithm updates the abnormal word counter whenever the character matches the sensitive word prefix in the DFA. The algorithm is able to match variant sensitive words which have the same prefix as the sensitive word's prefix in the dictionary. This feature improves the ability to recognize variant sensitive words. The algorithm can calculate the text abnormal rate only through the text itself. Therefore, anyone can evaluate the anomaly distribution of samples in a data set. By finding the suitable text abnormality threshold in different scenarios, the method can adapt to different data sets.

Based on the DFA filter, we can set different abnormal thresholds according to data distribution. The sample which has an abnormal ratio greater than the threshold is defined as noise text. Because the algorithm uses the method of counting the prefix word number that matches sensitive words in the dictionary, it is certain that some normal words have the same prefix as the sensitive word prefix. Therefore, it is necessary to set the threshold reasonable. If the threshold is too large, it will lead to the leakage of the abnormal sample. On the contrary, if the threshold is too small, it may cause misjudgment.

In general, the text length of the Description is several times that of the Title. The descriptions tag will contain more information because the text is generally longer, which helps the classifier to learn features when training data is not enough. The Title is relatively shorter and contains more concise information, making it more difficult to learn enough characteristics that are good for classification from category data with less training data. In the hierarchical category tree, we define the nodes at the top level as the parent nodes, and the nodes at the bottom level are the child nodes. At the same time, because the differences between parent categories are often greater than those between subcategories, this paper defines the classification of subcategories as fine-grained classification tasks and the classification of parent categories as coarse-grained classification tasks. Different filtering algorithms are designed for the coarse-grain and fine-grain.

Based on the analysis, the filtering algorithm for coarse-grained classification tasks is shown in Algorithm 2. The algorithm takes the Description as the default input first, aiming at improving the classification effect of the category with few training samples. The algorithm first calculates the anomaly of the Description, and if the Description is recognized as noisy text, it is replaced with the Title as input. Otherwise, the Description is used as input.

Algorithm 3 shows a filtering algorithm for fine-grained classification. The algorithm uses the Title as input. Fine-grained classification task contains more categories, and the overall accuracy requirements are higher. Algorithm 3 calculates the anomaly of the Description and the Title at the same time. If the Title is noisy text and the Description is normal, the Title is replaced with the Description for input.

5. Experiment

5.1. Data Set. The experiment is based on 64747 manually annotated data. Each row includes two levels' labels, domain, Title, Description, and keywords. In the data set, 22471 records' 'keywords' are empty, while only 592 records' 'Description' is empty. If 'keywords' are used for classification, too many web pages without keywords cannot be classified. Therefore, the scheme of using keywords for

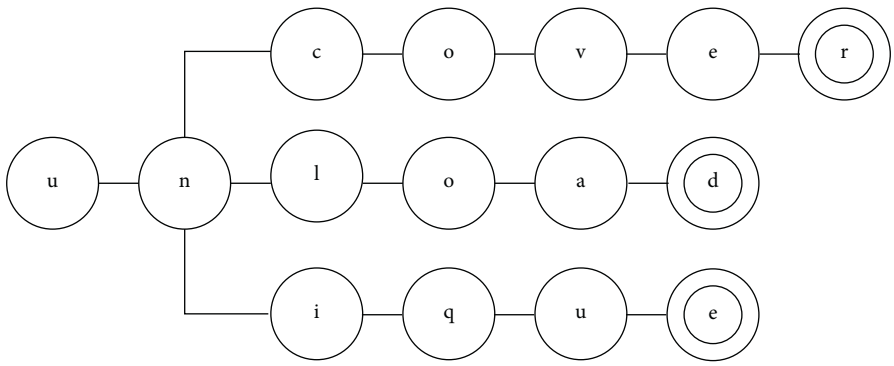

Figure 1: Sensitive word forms preserved by DFA.

Require: *De scription*, threshold, *filter*
Ensure: text abnormal rate
function F1(*filter*, *De scription*, *threshold*)
len ⟵ *De scription.length*
abLen ⟵ 0
abRate ⟵ 0
start ⟵ 0
result ⟵ 0
while *start* < *len* **do**
level ⟵ *filter.wor dc hain*
step ⟵ 0
for *char* ∈ *De scription*[*start*:] **do**
if *char* ∈ *level* **then**
step ⟵ *step* + 1
if *filter.en df lag* ∉ *level*[*char*] **then**
level ⟵ *level*[*char*]
else
abLen ⟵ *abLen* + 1
start ⟵ *step* − 1
end if
else
end if
end for
start ⟵ *start* + 1
end while
abRate ⟵ *abLen*/*len*
return *abRate*
end function

Algorithm 1: Text abnormal rate evaluation algorithm.

Require: *Title*, *Description*, *threshold*, filter
Ensure: training input
function F2(*filter*, *threshol d*, *Description*, *Title*)
input ⟵ *De scription*
abnormalRate ⟵ F1(*filter*, *Description*, *threshold*)
if *abnormalRate* > *threshhold* **then**
input ⟵ *Title*
end if
return *input*
end function

Algorithm 2: Coarse-grained classification algorithm.

Require: *Title*, Description, *threshold*, filter
Ensure: training input
function F3($filter, threshold, Description, Title$)
$input \longleftarrow Title$
$TitleRate \longleftarrow$ F1($filter, Title, threshold$)
$De\ scriRate \longleftarrow$ F1($filter, Description, threshold$)
if $TitleRate > threshhold$ **and** $DescriRate < threshold$ **then**
$input \longleftarrow Description$
end if
return *input*
end function

Algorithm 3: Fine-grained classification algorithm.

classification is excluded. The data set has been open-source on our website. The link is http://csri.scu.edu.cn/info/1012/2827.htm.

Table 1 shows the category structure and data distribution in data sets. Parent categories tend to have thousands of samples, while the sample size of subcategories is usually less than 1000. Besides, the text feature difference between different parent categories is greater, which is easier to distinguish. In contrast, the samples of subcategories of the same parent category are less different. The number of subcategories in the hierarchical tree is large, with many subcategories lacking sufficient training samples. The business services category has 6429 records, while the other cultural category has only 355 records. So, it is easy to encounter data skew.

5.2. Experiment on the Abnormal Rate Threshold. First, we conduct experiments to find a reasonable threshold range in our data set. As shown in Figure 2, if the threshold is set above 0.2, a few samples are detected as abnormal samples, and if the threshold is set below 0.05, almost all samples are determined to be abnormal samples. It indicates the experimental reasonable algorithm threshold should be set between 0.05 and 0.20 for our data set.

In Figure 3, UA represents the sample number that the Title or the Description is abnormal, while UL represents the number of samples that both the Title and the Description are abnormal. The difference between UA and UL indicates the number of samples with only one tag as noisy text. The method proposed in this paper tries to input the other tag's text while one tag's text is abnormal, so the larger the gap between UA and UL, the more effective the proposed method is.

Therefore, the algorithm proposed in this paper can adapt to different data sets by adjusting the threshold. In general, the method proposed in this paper has a certain generality and scalability.

5.3. Experiment on Classification Performance. We randomly choose 10,000 sampled data in the data set and crawl the web source code using data to make a comparison between tag text length and the entire page length. The experiment shows the average length of the entire web page is about 77960 words, the Title's average length is only 7 Chinese characters, and the Description's average length is 78 Chinese characters. Compared to the method of crawling the entire web page, crawling information under the Title and the Description separately only takes up little storage space, reducing the complexity of text storage and processing.

5.3.1. Evaluation Criteria. The effectiveness of classification tasks is often measured by recall rate and F1-score. In this paper, the coarse-grained classification effect is evaluated using a precision/recall rate and F1-score. The calculation formula is shown as follows:

$$\text{accuracy} = \frac{\text{TP} + \text{TF}}{\text{TP} + \text{TF} + \text{FP} + \text{FN}}, \tag{1}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{2}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3}$$

$$F1 - \text{score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{4}$$

The fine-grained classification needs to classify many categories. This paper selects the average/max/min of each parent category's subcategories to make a comparison. For example, if there are 7 subcategories under the parent category 'Entertain', the max precision value in the 7 categories is 0.89, and the min precision value is 0.82; the average is 0.87. The values will be presented in the form of "0.87/0.89/0.82."

5.3.2. Comparison and Analysis of Deep Learning Algorithm. To select the better deep learning classification algorithm, the effects of TextCNN, LSTM, and Bi-LSTM (bidirectional LSTM) on coarse-grained classification are initially compared in Table 2. We use the Description as input for this experiment. Considering the imbalance of the number of category samples, the parameter "class_weight" is set to "balanced" during training.

Experiments show that LSTM has achieved better results on coarse-grained classification in the data in this paper.

Table 1: Category structure.

First-level category (abbreviation)	Second-level category
Entertainment (enter)	Photo fiction: 741, video music: 1285, entertainment: 791, games and animation: 732, chat and make friends: 552, leisure and fitness: 430
Networks (net)	Business portal: 1656, network marketing: 706, network others: 627, website resources: 3849, software and hardware communication: 1988
Economy (eco)	Business services: 6429, construction environment: 1618, legal finance: 1020, agriculture, forestry, animal husbandry, and fishery: 981, transportation and logistics: 464, industrial products: 4974, machinery and electronics: 6284
Life service (life)	Fashion and beauty: 1185, health care: 2843, department store: 1744, tourism and transportation: 2379, catering and food: 930, real estate and home: 2409, common sense of life: 807, daily necessities: 2409, other life: 1996
Education (edu)	Higher education: 733, cultural other: 355, higher education: 642, human resources: 2046, sports and arts: 1246
Blog (blog)	Computer network: 640, life service: 847, leisure and entertainment: 537, blog others: 366
Others (others)	Group organization: 410, personal website: 995, others: 925, comprehensive website: 2157, news synthesis: 317

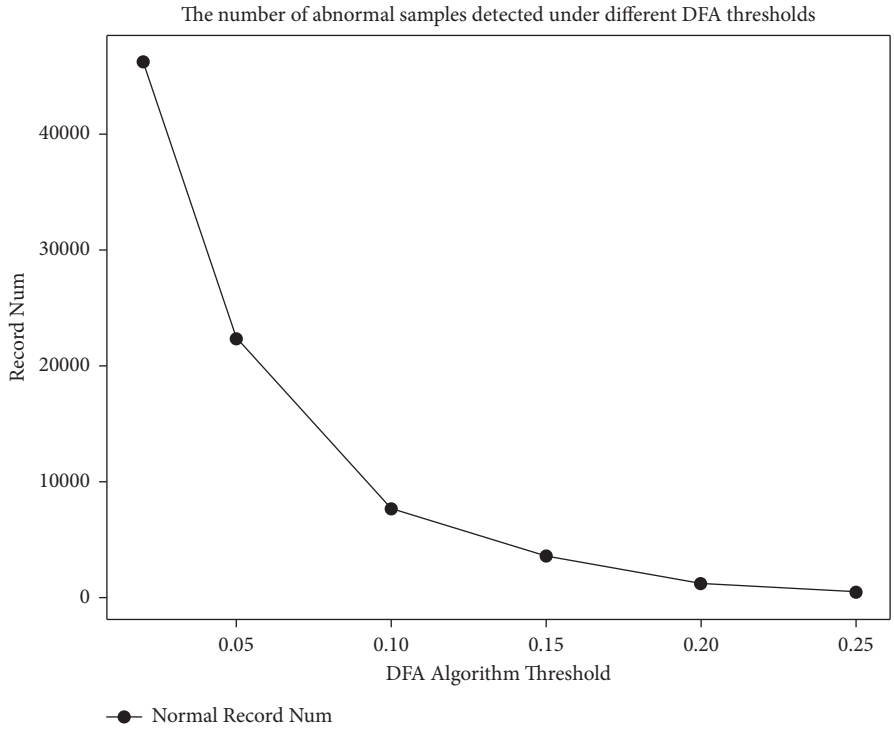

Figure 2: Trend of abnormal samples with DFA threshold.

5.3.3. Comparison and Analysis on Text Preprocess Performance in Time and Classification Effect. In the process of classification, first, preprocessing, such as word segmentation and removing stop words, is carried out. Then, we use a data cleaning algorithm to produce training input. After preprocessing, we train the deep learning classifier for classification. In the output layer, we set 'Sigmoid' as the activation function to convert multiclassification problems into multiple two-classification problems. We choose the category of neurons with the highest probability of two classifications, to get the category output with a hierarchical relationship.

As shown in Table 2, we try to find a better deep learning model for the web page classification task. The results show that LSTM achieved better performance in most categories. TextCNN ignores the word order, so it works well in scenarios where the word order is not sensitive. In contrast, LSTM and Bi-LSTM can capture sequence information and work better in application scenarios where word order is important.

In [8], the method uses both the Title, Description, and keywords as input. Five layers of RNN are used for training. The batch size is set to 1000 according to the paper, and

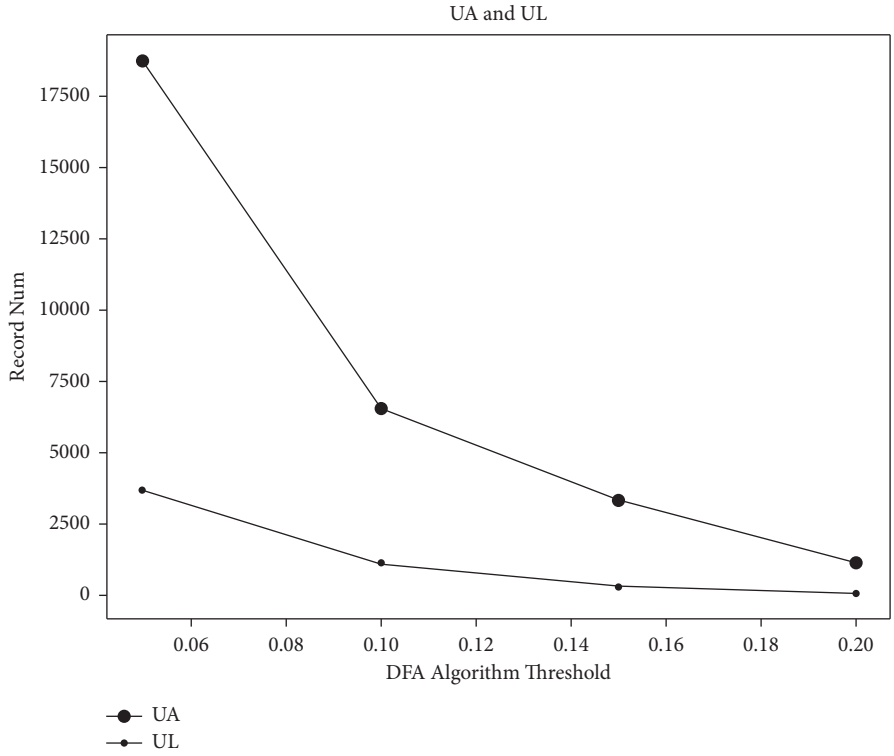

Figure 3: Trend of UA and UL with DFA threshold.

Table 2: Comparison of coarse-grained classification results of different deep learning algorithms.

Category/input Metrics	TextCNN Precision/recall/F1-score	LSTM Precision/recall/F1-score	Bi-LSTM Precision/recall/F1-score
Enter	0.70/0.27/0.39	**0.88/0.93/0.90**	0.86/0.84/0.85
Net	0.49/0.51/0.50	**0.90/0.94/0.92**	0.84/0.87/0.85
Eco	0.59/0.77/0.67	**0.91/0.95/0.93**	0.86/0.93/0.89
Life	0.58/0.69/0.63	**0.94/0.85/0.89**	0.92/0.82/0.87
Edu	0.37/0.37/0.37	**0.88/0.92/0.90**	0.85/0.87/0.86
Blog	0.00/0.00/0.00	**0.84/0.87/0.85**	0.84/0.73/0.78
Others	0.67/0.01/0.01	0.74/0.73/0.74	**0.81/0.80/0.80**

The bold values are used to highlight the best results achieved in each indicator.

'Categorical Cross Entropy' is set as the loss function. Data is trained for 5 epochs.

We conduct experiments to compare the methods in [8] with our algorithm using the data set proposed in this paper. We also use LSTM to classify the full web page source code to make a comparison. In Table 3's experiment, we try to find the advantages of our method through experiments. We set the filter threshold to 0.10 and batch size to 16. We conduct the experiment 5 times, and the average result is shown in Table 3. Experiments show that, in coarse-grained classification tasks, the filtration method improved in precision, recall rate, and F1-score for coarse-grained classification tasks compared to using one meta tag as input. The precision of 7 parent categories is above 88%, and the accuracy of the parent categories is 24% higher than the meta tag-based method in [8] on average, which means our method can successfully improve data quality to improve the classification effect.

Figures 4 and 5 show the confusion matrix in one experiment to compare our method and the method in [8]. Results show that our method is superior to the method in paper [8] in the classification effect of all coarse-grained categories, especially in the category of "entertainment" and "blog"; the accuracy of our method in these two categories is significantly higher than that in [8]. The two categories are the categories with relatively few samples. This phenomenon proves that our method can effectively improve the data skew problem.

In Table 4's experiment, we evaluate our method's accuracy through comparison. Our method performs better than the method in [8] and the full-page method as input. Table 5's experiment shows that our method needs less training and preprocessing time than the full-page-based method because we use short text for classification. The method in [8] costs less time because the method sets the batch size to 1000. We set lower batch size because we find

Table 3: Comparison of the algorithm in [8] and the coarse-grained algorithm in this paper.

Category/input Metrics	Method in [8] Precision/recall/F1-score	Coarse-grained algorithm Precision/recall/F1-score	Full page-based using LSTM Precision/recall/F1-score
Enter	0.64/0.85/0.73	**0.89/0.93/0.91**	0.79/0.42/0.55
Net	0.68/0.89/0.77	**0.88/0.94/0.90**	0.90/0.49/0.64
Eco	0.88/0.90/0.89	**0.93/0.94/0.93**	0.68/0.69/0.69
Life	0.85/0.85/0.85	**0.91/0.88/0.89**	0.45/0.77/0.57
Edu	0.80/0.81/0.80	**0.89/0.91/0.90**	0.88/0.50/0.64
Blog	0.29/0.05/0.09	**0.89/0.83/0.86**	0.83/0.29/0.43
Others	0.45/0.17/0.25	**0.89/0.83/0.86**	0.79/0.44/0.57

The bold values are used to highlight the best results achieved in each indicator.

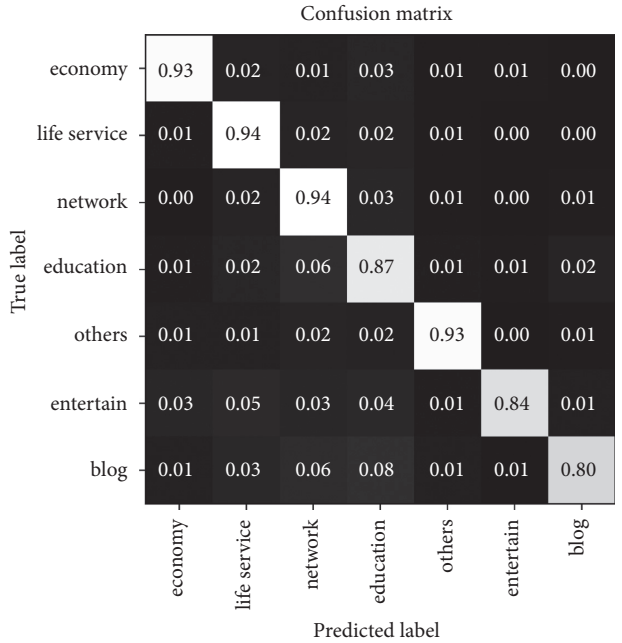

Figure 4: Confusion matrix of coarse-grained classification using filter algorithm in one experiment.

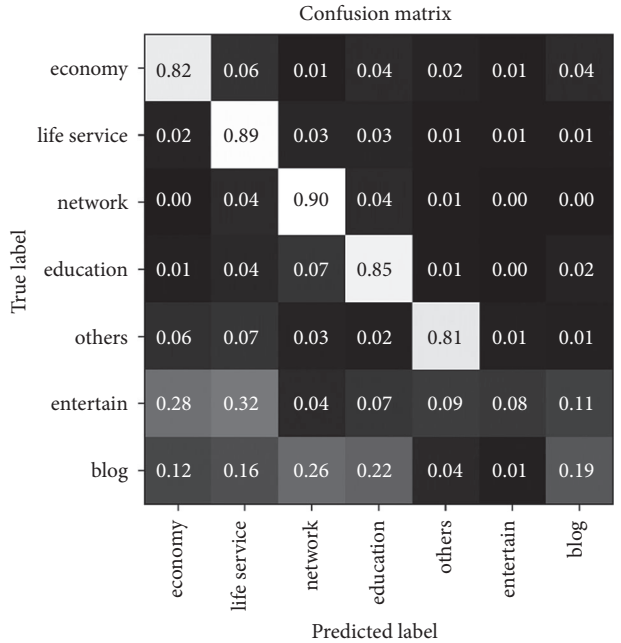

Figure 5: Confusion matrix of coarse-grained classification using the method in [8] in one experiment.

that the classification effect will be affected when setting higher batch size. The experimental results show that our method is about 7 times faster than the full-page input method, and the training efficiency is greatly improved by using the short text.

The hard decision-making algorithm of the top-down algorithm classifies only the subcategories of the category with the largest probability and chooses the category with the largest probability as the second-level classification result. It can guarantee classification efficiency.

In Table 5, the classification effects of the subcategories are compared. We also use title-based input as a control group. Based on comparative experiments, Bi-LSTM is selected for our method because it performs better in fine-grained classification tasks due to our experiments. When we conduct the experiment, some categories with few samples may not be sampled. We choose to rerun the experiment program under such circumstances. As can be seen from Table 5, the classification method using short text achieved the best effects in most categories in the fine-grained classification tasks and successfully improved the classification effect in the categories which have fewer samples. Our method performs better than the other two types of input in the classification task in the category with fewer samples. While comparing with the title-based input method, compared to 0.12 for the worst categories in the subcategories of education and culture and blog forums, our method increases the recall rate to 0.47 and 0.73, respectively. In contrast, the method in [8] has poor performance in many subcategories because of many '0.00' metrics. Compared with the methods in [8], our method has obvious advantages from different angles. In a word, the performance of this algorithm is more effective and stable. Tables 4 and 5 jointly prove that our method can achieve a better classification effect in less time.

All the results show that our method successfully improves the utilization of effective data and reduces the interference of polluted data, to significantly improve the classification effect in most categories. For the categories with poor classification effect due to a small amount of data, the negative impact caused by insufficient samples is successfully reduced after the improvement of data utilization.

Table 4: Comparison of the algorithm in [8] and the coarse-grained algorithm in this paper about training time cost.

Category/time	Method in [8]	Coarse-grained algorithm	Full page using LSTM
Preprocessing time cost/minutes	**1**	**1**	45
Training cost/minutes	**21**	86	106

The bold values are used to highlight the best results achieved in each indicator. Less time cost means better result.

Table 5: Comparison of the fine-grained classification performance.

Metrics	Precision (average value/max value/min value)			Recall (average value/max value/min value)			F1-score (average value/max value/min value)		
Method	Title-based classification	Algorithm in [8]	Fine-grained algorithm	Title-based classification	Algorithm in [8]	Fine-grained algorithm	Title-based classification	Algorithm in [8]	Fine-grained algorithm
Enter	0.92/0.99/0.83	0.08/0.29/0.00	**0.92/0.99/0.86**	0.92/0.99/0.81	0.17/1.00/0.00	**0.92/0.99/0.86**	**0.92/0.98/0.86**	0.08/0.45/0.00	0.92/0.86/0.86
Net	**0.85/1.00/0.73**	0.42/0.65/0.00	0.87/0.91/0.71	0.85/0.93/0.13	0.40/0.92/0.00	**0.87/0.94/0.13**	0.85/0.90/0.24	0.40/0.76/0.00	**0.87/0.93/0.23**
Eco	**0.90/0.94/0.84**	0.53/0.88/0.38	0.90/0.94/0.82	0.90/0.94/0.82	0.31/0.73/0.10	**0.90/0.96/0.82**	0.90/0.92/0.85	0.34/0.52/0.17	**0.90/0.93/0.86**
Life	**0.92/0.96/0.86**	0.77/0.91/0.47	0.91/0.93/0.86	**0.92/1.00/0.77**	0.77/0.86/0.60	0.91/1.00/0.74	**0.92/0.96/0.85**	0.76/0.86/0.60	0.91/0.97/0.83
Edu	0.88/1.00/0.68	0.24/0.87/0.00	**0.88/0.92/0.76**	0.88/0.96/0.12	0.34/0.94/0.00	**0.88/0.93/0.47**	0.88/0.94/0.21	0.26/0.82/0.09	**0.88/0.94/0.58**
Blog	0.75/0.85/0.00	0.21/0.47/0.00	**0.92/0.97/0.80**	0.75/1.00/0.00	0.32/0.68/0.00	**0.92/1.00/0.73**	0.75/0.83/0.00	0.25/0.56/0.00	**0.92/0.95/0.80**
Others	0.90/0.97/0.84	0.10/0.42/0.00	**0.91/0.97/0.89**	0.88/0.95/0.79	0.25/1.00/0.00	**0.91/0.98/0.70**	**0.89/0.95/0.79**	0.15/0.59/0.00	0.91/0.94/0.81

The bold values are used to highlight the best results achieved in each indicator, including precision (average value/max value/min value), recall (average value/max value/min value), and F1-score (average value/max value/min value).

Moreover, the improvement of the classification effect in the category with sufficient data also proves that this method can effectively remove the polluted data and reduce the classification interference. According to our test in a real scene. The model generated by the method proposed in this paper can be used for prediction, and the hierarchical classification prediction of 100-web-page data can be completed in about 67 seconds in a single-machine environment.

6. Conclusion

Hierarchical web page classification can provide fine-grained data support for web page data organization and management. To solve the problem of classification efficiency, this paper proposed a web page classification method based on meta tag text such as the Title and the Description. Besides, web page data may be attacked and injected with some irrelevant information affecting classification. In order to reduce the negative impact of these injected data on data quality, a data cleaning method based on the ratio of sensitive words is designed to recognize and clean noisy text in the meta tags. We also published a manually annotated hierarchical web page classification data set including two-level categories. The experiments' results show that our method significantly improves the classification effect in most categories and shows good efficiency. The classification effect has also proven to be effective in hierarchical classification scenarios.

There are other short text tags in web pages that can be used for web page classification, and there are a small number of web pages that do not have the Description tags, which can only rely on the Title while classifying. In the future work, we plan to expand further our data set and consider more short text tags that are useful to serve the classification. More advanced deep learning models such as the Transformers and the Bert can also be taken into account. These methods may improve the generality and performance of web page classification [18–23].

Data Availability

The data supporting this paper are from our manually annotated data, which have been cited in Section 5.1. The processed data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Xiang Song and Yi Zhu conceived the experiments; Xiang Song conducted the experiments; and Xiang Song, Xuemei Zeng, and Xingshu Chen analyzed the results. All authors reviewed the manuscript.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. U19A2081, 61802270, and 61802271), Fundamental Research Funds for the Central Universities (no. SCU2021D048), and Science and Engineering Connotation Development Project (no. 2020SCUNG129).

References

[1] L. Deng, D. Xin, and J. Z. Shen, "Web page classification based on multiple features and combined multi-classifiers," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 07, pp. 995–1005, 2020.

[2] D. Zhang, *Research and Implementation of Content-Oriented Web page Classification*, Nanjing University of Posts and Telecommunications, Nanjing, China, 2017.

[3] N. Q. Do, A. Selamat, O. Krejcar, T. Yokoi, and H. Fujita, "Phishing webpage classification via deep learning-based algorithms: an empirical study," *Applied Sciences*, vol. 11, no. 19, 2021.

[4] A. Gupta and R. Bhatia, "Ensemble approach for web page classification," *Multimedia Tools and Applications*, vol. 80, pp. 1–22, 2021.

[5] Y. Cheng, "Subject of website classification based on Word2Vec," *Computer & Digital Engineering*, vol. 47, no. 1, pp. 169–173, 2019.

[6] T. Zhang, *The Implementation of Large-Scale Chinese Website Classification System Based on Improved SVM Algorithms*, Beijing University of Posts and Telecommunications, Beijing, China, 2019.

[7] C. Luo and S. Wang, "Research on web page classification algorithm based on deep learning and part of speech tagging," *Computer Technology and Development*, vol. 28, no. 8, pp. 71–74, 2018.

[8] A. B. Ebubekir and B. D. Banu, "Web page classification using RNN," *Procedia Computer Science*, vol. 154, pp. 62–72, 2019.

[9] L. Jia, Y. Liu, B. Wang, H. Liu, and G. Xin, "A hierarchical classification approach for tor anonymous traffic," in *Proceedings of the 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, pp. 239–243, IEEE, Guangzhou, China, May 2017.

[10] S. Oh, "Top-k hierarchical classification," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[11] S. Banerjee, C. Akkaya, F. P. Sorrosal, and K. Tsioutsiouliklis, "Hierarchical transfer learning for multi-label text classification," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6295–6300, Florence, Italy, July 2019.

[12] A. Naik and H. Rangwala, "Filter based taxonomy modification for improving hierarchical classification," 2016, https://arxiv.org/abs/1603.00772.

[13] A. Naik and H. Rangwala, "Inconsistent node flattening for improving top-down hierarchical classification," in *Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 379–388, IEEE, Montreal, QC, Canada, October 2016.

[14] A. Rakhlin, *Convolutional Neural Networks for Sentence Classification*, EMNLP, Doha, Qatar, 2014.

[15] X. Ma and E. Hovy, *End-to-end Sequence Labeling via Bi-directional Lstm-Cnns-Crf*, The Association for Computational Linguistics, PA, USA, 2016.

[16] M. Najam, U. Younis, and R. U. Rasool, "Speculative parallel pattern matching using stride-k DFA for deep packet inspection," *Journal of Network and Computer Applications*, vol. 54, 2015.

[17] P. Xue and N. I. Wuxur, "Sensitive information filtering algorithm based on text information network," *Computer Engineering and Design*, vol. 37, no. 9, pp. 2447–2452, 2016.

[18] S. Dumais and H. Chen, "Hierarchical classification of web content," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 256–263, Association for Computing Machinery, Athens Greece, July 2000.

[19] Y. Fu, *Research on Chinese Book Classification Based on LSTM Model*, Nanjing University, Nanjing, China, 2017.

[20] Kuerban, *Research and Implementation of Web page Topic Classification Method Based on LSTM and Transfer Learning*, Xinjiang University, Xinjiang, China, 2019.

[21] S. G. Ramirez, B. Krawczyk, and S. Garcia, "A survey on data pre-processing for data stream mining: current status and future directions," *Neurocomputing*, vol. 239, no. MAY24, pp. 39–57, 2018.

[22] Z. Chen, *The Research of Web Information Extraction Technique and Application Based on NFA Regular Matching*, Hangzhou Dianzi University, Hangzhou, China, 2015.

[23] J. Zhong, *Research on Text Classification Based on Deep Learning*, University of Electronic Science and Technology of China, Chengdu, China, 2020.