Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Web page classification based on heterogeneous features and a combination of multiple classifiers*

Li DENG, Xin DU, Ji-zhong SHEN†‡
College of Information Science & Electronic Engineering, Zhejiang University, Hangzhou 310027, China
†E-mail: jzshen@zju.edu.cn

Received May 13, 2019; Revision accepted Aug. 21, 2019; Crosschecked June 6, 2020

Abstract: Precise web page classification can be achieved by evaluating features of web pages, and the structural features of web pages are effective complements to their textual features. Various classifiers have different characteristics, and multiple classifiers can be combined to allow classifiers to complement one another. In this study, a web page classification method based on heterogeneous features and a combination of multiple classifiers is proposed. Different from computing the frequency of HTML tags, we exploit the tree-like structure of HTML tags to characterize the structural features of a web page. Heterogeneous textual features and the proposed tree-like structural features are converted into vectors and fused. Confidence is proposed here as a criterion to compare the classification results of different classifiers by calculating the classification accuracy of a set of samples. Multiple classifiers are combined based on confidence with different decision strategies, such as voting, confidence comparison, and direct output, to give the final classification results. Experimental results demonstrate that on the Amazon dataset, 7-web-genres dataset, and DMOZ dataset, the accuracies are increased to 94.2%, 95.4%, and 95.7%, respectively. The fusion of the textual features with the proposed structural features is a comprehensive approach, and the accuracy is higher than that when using only textual features. At the same time, the accuracy of the web page classification is improved by combining multiple classifiers, and is higher than those of the related web page classification algorithms.

Key words: Web page classification; Web page features; Combined classifiers
https://doi.org/10.1631/FITEE.1900240 **CLC number:** TP391

1 Introduction

The huge amount of information on the Internet continues to expand over time, which provides people access to valuable resources. Web page classification is critical for website management and information retrieval, such as developing and maintaining web directories, improving the efficiency of search engines, and filtering web pages (Qi and Davison, 2009). Web page classification can be achieved by evaluating the textual features of web pages (Kumari and Reddy, 2012; Li et al., 2017), structural features of web pages (Cai et al., 2003; Panchekha and Torlak, 2016), and the relationships between web pages (Qi and Davison, 2006). However, web page classification based on a single feature is subject to bias. For example, some web pages lack textual information, so it is difficult to accurately classify them based just on textual features. Better classification results can be obtained using other effective features and the fusion of heterogeneous features.

Moreover, multiple classification results can be obtained through various classification algorithms, such as support vector machine (SVM) (Ali et al., 2017), deep neural network (Gogar et al., 2016), and

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 61471314) and the Welfare Technology Research Project of Zhejiang Province, China (No. LGG18F010003)

ORCID: Li DENG, https://orcid.org/0000-0002-9978-3337; Xin DU, https://orcid.org/0000-0002-6215-9733; Ji-zhong SHEN, https://orcid.org/0000-0002-9031-2379

decision tree (Onan, 2016). Due to the differences in features and classifiers, a sample may be classified incorrectly by one classifier but can be classified correctly by other classifiers. The classification performance can be improved by combining multiple classifiers, such as voting, bagging, and boosting (Baskin et al., 2017). Zhu et al. (2016) used a decision matrix to construct a model with multiple SVM classifiers to classify web pages, but the combination of different types of high-performance classifiers could be better. Elsalmy et al. (2017) enhanced the predictive power of web page classification models by stacking, but stacking is complicated.

In this study, we propose a web page classification method based on heterogeneous features and a combination of multiple classifiers. It fuses textual and structural features to comprehensively evaluate web page features and combines deep neural network and SVM classifiers to accurately classify web pages. The main contributions of this study are as follows:

1. We use the tree-like structure of web pages as features. HyperText Markup Language (HTML) tags in the HTML documents of web pages are exploited and converted into vectors to characterize the structural features of web pages. Heterogeneous textual and structural features of web pages are fused by vector concatenation.

2. We propose confidence here as a criterion to compare the classification results of different classifiers. Then, multiple classifiers are combined with decision strategies, such as voting and confidence comparison, to give a good classification result at a high confidence interval.

3. Experiments are conducted on the famous Amazon dataset, 7-web-genres dataset, and DMOZ dataset, on which the accuracies of the web page classification are increased to 94.2%, 95.4%, and 95.7%, respectively.

2 The proposed method

2.1 Extracting structural and textual features of web pages

Different categories of web pages contain different kinds of content, and the structural styles of web pages are also different. The key to effective web page classification is identifying the features of high distinction. Therefore, it is critical to extract the effective features of web pages.

Based on the work of computing the frequencies of different HTML tags (Zhu et al., 2016), we capture the structural features of web pages by constructing vectors according to the tree-like structure of HTML tags. In web pages, HTML tags are arranged in a tree-like structure. For example, the following is an HTML document:

```
    <html><head><title>Example</title></head>
    <body>
    <p class="title"><b>Example</b></p>
    <p class="description">There are three examples:
    <a href="http://example.com/1" class="example"
id="link1">Example1</a>,
    <a href="http://example.com/2" class="example"
id="link2">Example2</a> and
    <a href="http://example.com/3" class="example"
id="link3">Example3</a>.</p>
    </body></html>
```

The tree-like structure of the HTML tags is shown in Fig. 1.

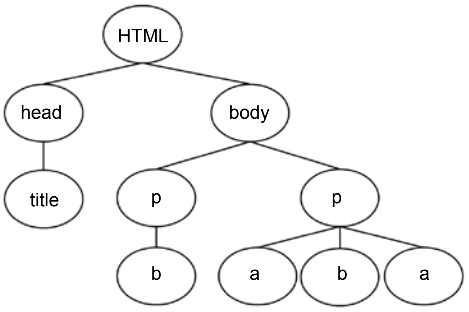

Fig. 1 Tree-like structure of HTML tags

Evaluating the distribution of HTML tags helps reveal the structural features of web pages. According to the statistical data (Heinrich, 2017), 20 HTML tags account for 95% of all tags in the HTML documents. These tags are <a>, <div>, <li>, <span>, <img>, <td>, <p>, <ul>, <option>, <meta>, <tr>, <link>, <input>, <table>, <tbody>, <dd>, <h2>, <h3>, <hr>, and <dt>. We then construct 20-dimensional vectors to represent these tags and traverse the HTML tags in order as the structural features of web pages. For example, the following vectors represent the web page structure in Fig. 1:

[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].

The web pages of www.amazon.com (Fig. 2), www.nba.com (Fig. 3), and www.cancer.gov (Fig. 4) are taken as cases to show the tree-like structures.

Fig. 2 shows part of the web page and tree-like structure of www.amazon.com. Area 1 in the dotted

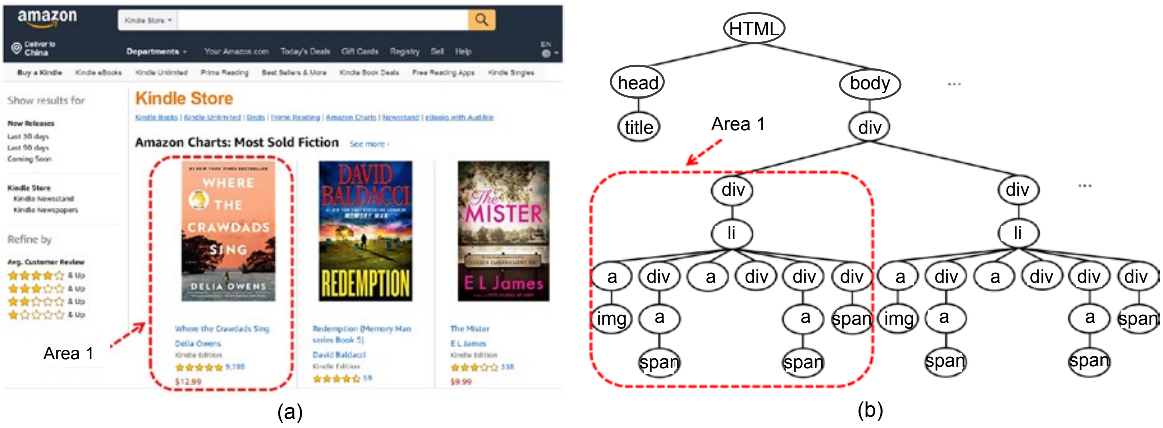

Fig. 2 Web page (a) and tree-like structure (b) of www.amazon.com

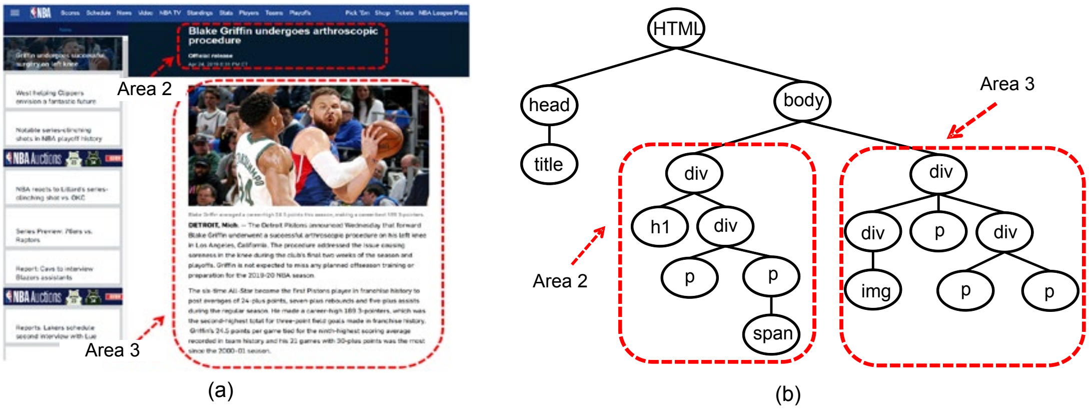

Fig. 3 Web page (a) and tree-like structure (b) of www.nba.com

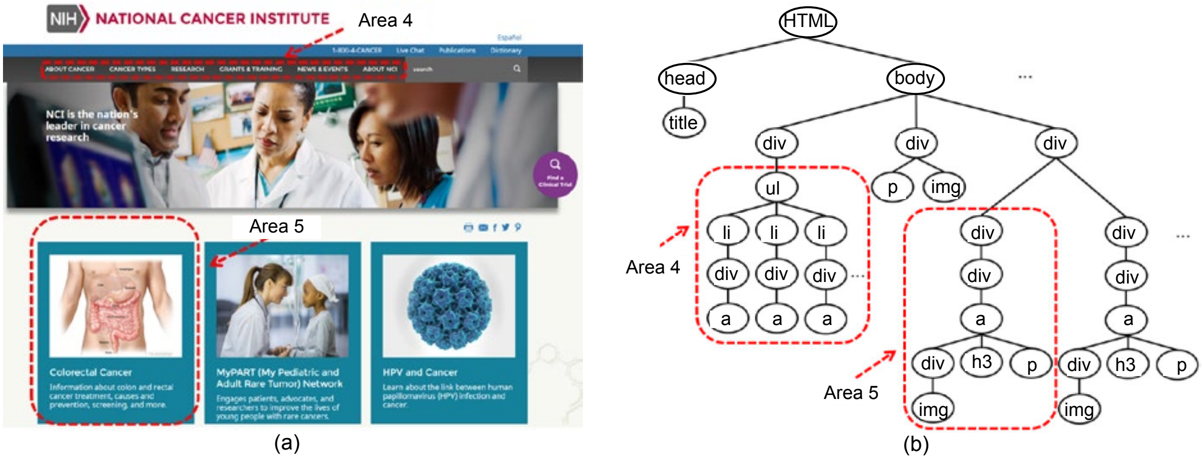
NIH NATIONAL CANCER INSTITUTE
Area 4
Area 5
NCI is the nation's leader in cancer research
Colorectal Cancer
MyPART (My Pediatric and Adult Rare Tumor) Network
HPV and Cancer
(a)
HTML
head
body
title
div
ul
li
a
p
img
h3
Area 4
Area 5
...
(b)

Fig. 4 Web page (a) and tree-like structure (b) of www.cancer.gov

box represents a book for sale and there are several books on the page, so the structure of area 1 will be repeated in the whole tree-like structure.

By analogy, the entire web page is represented as a "huge tree," which reveals the structural features of web pages.

Similarly, as shown in the dotted boxes in Fig. 3, area 2 represents the title of the article and area 3 the content of the article. In Fig. 4, area 4 in the dotted box represents the navigation of the web page and area 5 the report about cancer. The tree-like structure of HTML tags is exploited to characterize the web page structural features.

Vectors are constructed to represent the web page structure of HTML tags. For example, the following vectors represent the structure of HTML tags in Fig. 4 (the first dotted box represents area 4 and the second dotted box represents area 5):

```
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].
```

The tree-like structures in Figs. 2 and 3 can be converted into vectors in the same way to capture the structural features of web pages.

Text information is extracted from the content in HTML tags of <title>, <meta>, <Hn>, <a>, <b>, and <p>, representing the title, meta information, headings, hyperlinks, bold, and paragraph, respectively. The extracted texts are then preprocessed; i.e., all letters are converted into lowercase with garbled codes, abbreviations, numbers, and stop words removed. Stop words are words which are filtered out before or after processing of natural language data. The texts are afterwards converted into vectors using word2vec (Mikolov et al., 2013), whereby each text is mapped to a unified dimensional vector in the vector library of word2vec with the textual features preserved. In this way, similar words are mapped close to each other in the vector space, and the vectors are therefore tinged with semantic information.

2.2 Fusion of features for training and classification

Due to the difference in vector dimensions of the textual and structural features, these vectors need to be adjusted to be vectors with the same dimension by the numpy.reshape function (https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.reshape.html). The heterogeneous textual and structural features of web pages are fused by vector concatenation and fed into classifiers for training. After training, the vectors are input to the classifiers for classification.

The classifiers we use are the long short-term memory (LSTM) (Gers et al., 2000) network and SVM (Xue et al., 2006). LSTM is a special type of deep neural network, which has internal memory to allow long-term dependencies to affect the output (Sze et al., 2017). LSTM is effective in processing time series data and is widely used in speech recognition, machine translation, text classification, and other fields. SVM is based on structural risk minimization and the Vapnik-Chervonenkis (VC) dimension theory, and a hyperplane is built to separate different types of samples (Wei et al., 2017). SVM has good performance in terms of classification and generalization.

2.3 Acquisition of confidence

Confidence is used here to measure the reliability of classification results. Given a sample, the classifier yields predicted scores for each category and the classification result is the category with the maximum predicted score. The higher the maximum predicted score, the more likely the classification result being correct. However, predicted scores of different classifiers are not comparable, making it difficult to compare results of different classifiers. Thus, we propose confidence as a criterion to compare the classification results of different classifiers.

Assume that all web page data of a dataset is marked at 100%, of which 60%, 20%, and 20% are assigned as datasets A, B, and C, respectively. We train the classifier with dataset A and then classify dataset B. The classification results of dataset B are sorted as dataset B^* according to the maximum predicted scores from high to low. For example, for the 7-web-genres dataset (Zhu et al., 2016), Table 1 lists 10 results with the low predicted score for dataset B^*, and Table 2 lists 10 results with the high predicted score for dataset B^*. Numbers 1–7 represent seven categories, and the real categories are the known labels in the dataset. It can be seen that the predicted scores in Table 1 are low, and only 4 of the 10 classification results are the same as the real categories. In Table 2, the predicted scores are high, and all the 10 classification results are correct.

There are a total of M samples in dataset B^*, and the maximum predicted score of the m^{th} classification result is d_m. We take a set of samples to calculate the confidence, and the number of samples is $2n$. We select the samples from the $(m-n)^{\text{th}}$ to the $(m+n)^{\text{th}}$ samples, and calculate the classification accuracy of the $2n$ samples as the confidence of the m^{th} classification result. Considering the boundary value, if $m-n\leq 0$, the maximum predicted score of the m^{th} classification result is high, the reliability of the classification result will be considered high, and the confidence is set to 1. If $m+n\geq M$, the maximum predicted score of the m^{th} classification result is low, the reliability of the classification result will be considered low, and the confidence is set to 0.

After the maximum predicted scores and confidences for the M samples are obtained, the relationship between the maximum predicted scores and confidences is shown in Fig. 5.

Table 1 Classification results with low predicted scores

Maximum predicted score	Classification result (category)	Real category
0.223 208	1	5
0.217 058	6	5
0.199 370	6	2
0.194 452	5	3
0.193 946	7	2
0.193 552	5	5
0.177 247	3	5
0.171 164	7	7
0.167 516	5	5
0.159 069	5	5

Table 2 Classification results with high predicted scores

Maximum predicted score	Classification result (category)	Real category
1.637 814	7	7
1.631 327	7	7
1.570 772	1	1
1.539 094	7	7
1.538 281	1	1
1.538 209	7	7
1.537 890	1	1
1.537 027	1	1
1.529 811	1	1
1.523 830	1	1

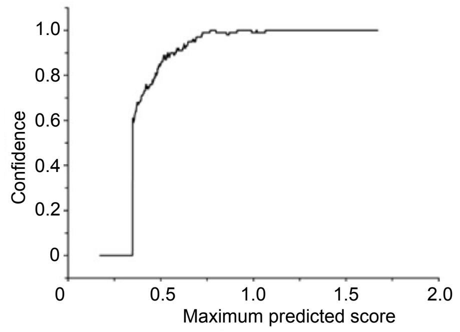

Fig. 5 Relationship between the maximum predicted scores and confidences

In dataset C, the classifier is tested. After a test sample is inputted to the classifier, the maximum predicted score and a classification result are obtained from the classifier. Clearly, according to the relationship between the maximum predicted scores and confidences, the confidence of the test sample can be obtained. For each test sample, classification results and confidences for multiple individual classifiers can be obtained.

The process of obtaining classification results and confidences of multiple individual classifiers is shown in Fig. 6.

2.4 Combination of multiple classifiers

The main idea behind our combined classifiers is to make decisions at the highest confidence interval.

The process of combining multiple classifiers is shown in Fig. 7, and the detailed steps are as follows:

Step 1: Several confidence intervals are obtained according to the confidence thresholds C_1, C_2, …, C_i (C_i represents the minimum threshold, $C_1>C_2>\ldots>C_i$), and the confidence thresholds are set according to the accuracy of the combined classifiers by trials, i.e., $C_1=0.95$, $C_2=0.90$, $C_3=0.80$, and $C_4=0.70$ (On the datasets in Section 3, the combination of multiple classifiers can be effectively achieved through experiments when $i=4$).

Step 2: For a test sample, after the classification results and confidences for multiple individual classifiers are obtained, we count the number N of classification results with confidence no less than the confidence threshold C_1. If the number N is greater than 0, a decision strategy will be implemented to give the final result by combining these classifiers, and the classification process ends; otherwise, step 3 is performed.

Step 3: For a test sample, we count the number N of classification results with confidence no less than

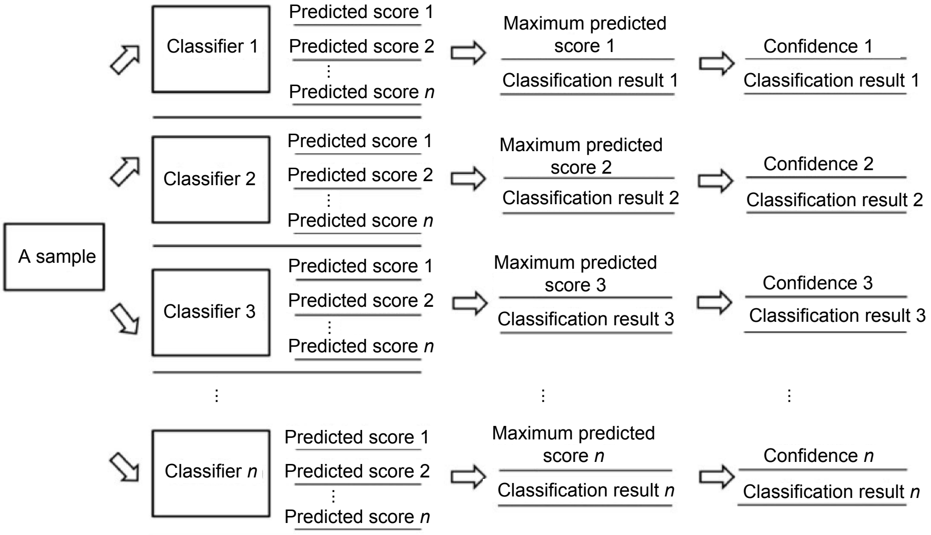

Fig. 6 Process of obtaining classification results and confidences for multiple individual classifiers

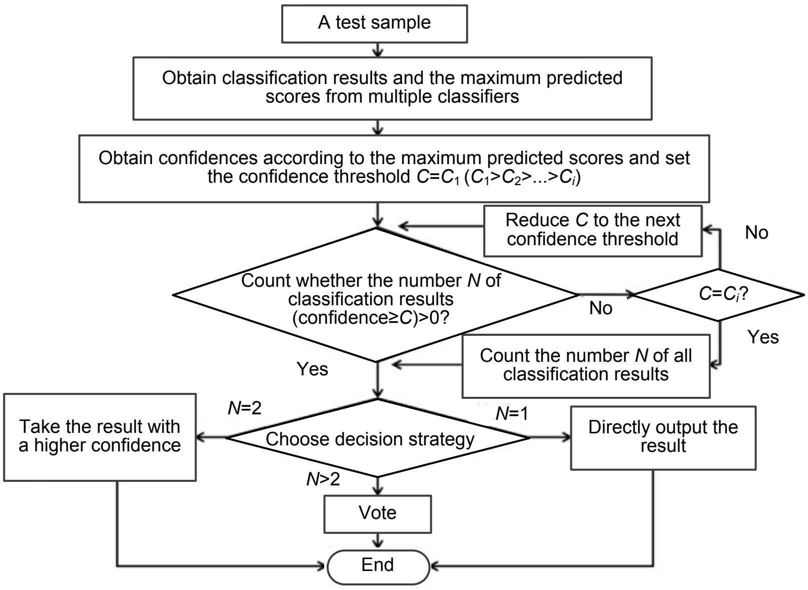
A test sample
Obtain classification results and the maximum predicted scores from multiple classifiers
Obtain confidences according to the maximum predicted scores and set the confidence threshold C=C1 (C1>C2>...>Ci)
Reduce C to the next confidence threshold
No
Count whether the number N of classification results (confidence≥C)>0?
No
C=Ci?
Yes
Count the number N of all classification results
Yes
N=2
N=1
Take the result with a higher confidence
Choose decision strategy
Directly output the result
N>2
Vote
End

Fig. 7 Process of combining multiple classifiers

the confidence threshold C_2 but less than the confidence threshold C_1. If the number N is greater than 0, a decision strategy will be implemented to give the final result by combining these classifiers, and the classification process ends; otherwise, step 4 is performed.

Step 4: By analogy, the classification process goes on until the final result of the combined classifiers is obtained at one of the confidence intervals or all confidences of classification results are less than the minimum confidence threshold. Then count the number N of all classification results and a decision strategy will be implemented to give the final result by combining all classifiers.

The decision strategy is that if N is greater than 2, vote; if N equals 2, compare the confidence values and take the result with a higher confidence; if N equals 1, directly output the corresponding classification result.

The combined classifiers include an SVM classifier based on textual features, an LSTM classifier based on textual features, an SVM classifier based on fusion of textual and structural features, and an LSTM classifier based on fusion of textual and structural features. After combining the four individual classifiers, we try to omit one of the individual classifiers in turn. If the accuracy of the combined classifiers increases, the individual classifier is omitted from the combination; otherwise, the individual classifier is retained in the combination. Using this process, the combination of multiple classifiers can be obtained. There is a need for heterogeneous features and classifiers to make full use of their complementary information; otherwise, the process is only a simple repetition and cannot effectively improve the classification performance.

3 Experimental results and analysis

To verify the effectiveness of the proposed method, 120 000 labeled web pages were crawled from the Alexa website of Amazon (https://www.alexa.com/topsites). There were 10 categories in the dataset, i.e., arts, business, computers, health, recreation, reference, science, shopping, society, and sports.

Our experiments were implemented in the environment of Ubuntu 16.10, Python 2.7.12, TensorFlow 1.2.0, and Numpy 1.14.1. The hyper parameter configuration of the LSTM was as follows: batch size 256, learning rate 0.002, and layer number 3. We implemented the SVM classifier through the open-source library TextGrocery, which is based on LibLinear. Parameters in TextGrocery were automatically set and were free of manual tuning.

The classification accuracy (ACC) is calculated by

$$\mathrm{ACC}=\frac{\mathrm{TN}+\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}, \tag{1}$$

where TP is the number of positive samples classified as positive samples, TN the number of negative samples classified as negative samples, FP the number of negative samples classified as positive samples, and FN the number of positive samples classified as negative samples.

Our test results on the Amazon dataset are shown in Table 3, and the confidence thresholds are C_1=0.95, C_2=0.90, C_3=0.80, and C_4=0.70. Table 3 shows that the method of fusing textual and structural features is comprehensive and effective. The accuracy is 4.5% higher than that when using only textual features in LSTM and 2.1% higher than that when using only textual features in SVM. The accuracy of web page classification is improved by fusing textual and structural features compared with the method using just textual features. The differences between various web page categories are effective features, so text and tree-like structure are used as features to classify web pages. In terms of pure textual features, SVM is better than LSTM, but in terms of fusion of textual and structural features, LSTM is better than SVM. Overall, the performances of LSTM and SVM are close on the Amazon dataset.

Table 3 Classification accuracy on the Amazon dataset

Method	ACC (%)
LSTM classifier based on textual features	89.2
SVM classifier based on textual features	90.3
LSTM classifier based on fusion of textual and structural features	93.7
SVM classifier based on fusion of textual and structural features	92.4
Combination of LSTM and SVM classifiers based on fusion of textual and structural features	94.2

SVM: support vector machine; LSTM: long short-term memory; ACC: accuracy

Founded on the high accuracy of both the SVM and LSTM classifiers based on the fusion of textual and structural features, the combination of these two classifiers further improves the accuracy of classification, reaching 94.2%. Using confidence, the classification results of different classifiers can be compared. Different types of classifiers are combined to make full use of their advantages; i.e., deep neural network has the ability to extract high-level features from a large amount of raw data (Sze et al., 2017), and SVM is widely used for text categorization because of its high generalization performance and high tolerance ability of processing high-dimensional vector classification (Xue et al., 2006). The combined classifiers are an LSTM classifier based on fusion of textual and structural features and an SVM classifier based on fusion of textual and structural features. Their performances are close and the complementarity of the classifiers is used to improve the classification accuracy.

The 7-web-genres dataset (Zhu et al., 2016) has a total of 1400 HTML pages in seven categories, i.e., blog, eshop, FAQ, online newspaper front page, listing, personal home page, and search page. They are functions of web pages.

The 7-web-genres dataset is a small-sample dataset, so a 10-fold cross-validation method (Onan, 2016) is used to make full use of the data. Table 4 shows the test results on the 7-web-genres dataset, and the confidence thresholds are C_1=0.95, C_2=0.90, C_3=0.80, and C_4=0.70. In Table 4, the accuracy of the SVM classifier based on fusion of textual and structural features is slightly improved compared with the SVM classifier based on textual features, and it provides complementary information for the combined classifiers. The accuracies of the LSTM and SVM classifiers based on textual features are 90.3% and 92.8%, respectively, the accuracy of the SVM classifier based on fusion of textual and structural features is 93.0%, and the combination of these three improves the accuracy of web page classification to 95.4%. Note that the LSTM classifier based on fusion of textual and structural features is not included in the combination. The reason is that the combined classifiers require the performances of the individual classifiers be close. If this rule is violated, the poor performance of one of the individual classifiers will decrease the overall performance of the combined classifiers, which is the case for the LSTM classifier based on fusion of textual and structural features here.

Table 4 Classification accuracy on the 7-web-genres dataset

Method	ACC (%)
LSTM classifier based on textual features	90.3
SVM classifier based on textual features	92.8
LSTM classifier based on fusion of textual and structural features	88.6
SVM classifier based on fusion of textual and structural features	93.0
Combination of the LSTM classifier based on textual features, the SVM classifier based on textual features, and the SVM classifier based on the fusion of textual and structural features	95.4

SVM: support vector machine; LSTM: long short-term memory; ACC: accuracy

We compared the proposed classification method with the related web page classification algorithms. Pritsos and Stamatatos (2013) proposed a random feature subspacing ensemble (RFSE) algorithm. Kumari and Reddy (2012) used a combined stemming approach (CSA) for web page classification. Zhu et al. (2016) used a decision matrix to construct a model with multi-classifier combination (MCC). Precision, recall, and F-measure were used to evaluate the performances of web page classification algorithms in these three works. We calculated these three parameters for comparison. Precision is the number of true positive samples divided by the total number of samples classified as positive samples. Recall is the number of true positive samples divided by the total number of real positive samples. F-measure is the harmonic mean of Precision and Recall.

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

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

$$F\text{-measure}=\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}. \tag{4}$$

Table 5 shows the comparison results of the proposed method with related algorithms on the 7-web-genres dataset. It can be seen that the results of the proposed method are excellent in Precision, Recall, and F-measure, all reaching 95.4%.

The DMOZ website is a famous open directory project built and maintained by volunteers from all over the world. The DMOZ-50 dataset is obtained from the DMOZ website and divided into 50 small datasets (Onan, 2016). The number of topics in the dataset ranges from 3 to 10. This dataset is preprocessed and the main information on the web page is extracted and saved.

Table 6 gives the comparison results of our method with several works on the DMOZ-50 dataset. Onan (2016) combined many feature selection algorithms and classification algorithms, the best one of which is the combination of AdaBoost, naive Bayes, and consistency-based feature selection, and the accuracy was 88.1%. Elsalmy et al. (2017) investigated the method of stacking with model trees and achieved an accuracy of 91.2%. Onan (2015) proposed an artificial immune system based algorithm, namely Immunos-1, for web page classification, and the accuracy was 92.4%. A higher accuracy of 95.7% was obtained using our proposed method, which demonstrates excellent effectiveness compared with the related web page classification algorithms.

Table 5 Comparison results of several works on the 7-web-genres dataset

Method	Precision (%)	Recall (%)	*F*-measure (%)
MCC	92.5	90.0	91.2
RFSE	91.2	89.9	90.5
CSA			91.5
Proposed method	95.5	95.4	95.4

MCC: multi-classifier combination; RFSE: random feature subspacing ensemble; CSA: combined stemming approach

Table 6 Comparison results of several works on the DMOZ-50 dataset

Method	ACC (%)
Combination of AdaBoost, naive Bayes, and consistency-based feature selection	88.1
Stacking with model trees	91.2
Immunos-1	92.4
Proposed method	95.7

ACC: accuracy

4 Conclusions and future work

To comprehensively evaluate the features of web pages and improve classification accuracy, a web page classification method based on heterogeneous features and a combination of multiple classifiers has been proposed. We have captured the structural features of web pages by constructing vectors according to the tree-like structure of HTML tags, and fused the heterogeneous features, namely the structural and textual features, by vector concatenation. The fusion of textual features and the proposed tree-like structure features is comprehensive and effective. Confidence has been proposed as a criterion of the reliability of the classification results. Deep neural network and SVM classifiers have been combined with decision strategies such as voting, confidence comparison, and direct output to give the final classification result at the highest confidence interval. Experimental results on the Amazon dataset, 7-web-genres dataset, and DMOZ dataset showed that the accuracies are increased to 94.2%, 95.4%, and 95.7%, respectively, by our proposed method, and demonstrated higher accuracy than the related web page classification algorithms. In future work, we will explore more effective features and combine different classifiers to improve the classification performance.

Contributors

Ji-zhong SHEN and Xin DU designed the research. Li DENG processed the data and drafted the manuscript. Ji-zhong SHEN helped organize the manuscript. Li DENG, Xin DU, and Ji-zhong SHEN revised and finalized the paper.

Compliance with ethics guidelines

Li DENG, Xin DU, and Ji-zhong SHEN declare that they have no conflict of interest.

References

Ali F, Khan P, Riaz K, et al., 2017. A fuzzy ontology and SVM-based web content classification system. *IEEE Access*, 5:25781-25797. https://doi.org/10.1109/ACCESS.2017.2768564

Baskin II, Marcou G, Horvath D, et al., 2017. Bagging and boosting of classification models. In: Varnek A (Ed.), Tutorials in Chemoinformatics, Wiley Online Library, p.241-247. https://doi.org/10.1002/9781119161110.ch15

Cai D, Yu SP, Wen JR, et al., 2003. Extracting content structure for web pages based on visual representation. Asia-Pacific Web Conf, p.406-417. https://doi.org/10.1007/3-540-36901-5_42

Elsalmy F, Ismail R, Abdelmoez W, 2017. Enhancing web page classification models. Int Conf on Advanced Intelligent Systems and Informatics, p.742-750. https://doi.org/10.1007/978-3-319-48308-5_71

Gers FA, Schmidhuber J, Cummins F, 2000. Learning to forget:

continual prediction with LSTM. *Neur Comput*, 12(10): 2451-2471.
https://doi.org/10.1162/089976600300015015

Gogar T, Hubacek O, Sedivy J, 2016. Deep neural networks for web page information extraction. IFIP Int Conf on Artificial Intelligence Applications and Innovations, p.154-163.
https://doi.org/10.1007/978-3-319-44944-9_14

Heinrich G, 2017. Evaluation of a distribution-based web page classification. In: Friedrichsen M, Kamalipour Y (Eds.), Digital Transformation in Journalism and News Media. Springer, Cham, p.55-68.
https://doi.org/10.1007/978-3-319-27786-8_6

Kumari KP, Reddy AV, 2012. Performance improvement of web page genre classification. *Int J Comput Appl*, 53(10): 24-27. https://doi.org/10.5120/8457-2265

Li HK, Xu Z, Li T, et al., 2017. An optimized approach for massive web page classification using entity similarity based on semantic network. *Fut Gener Comput Syst*, 76: 510-518. https://doi.org/10.1016/j.future.2017.03.003

Mikolov T, Chen K, Corrado G, et al., 2013. Efficient estimation of word representations in vector space.
https://arxiv.org/abs/1301.3781

Onan A, 2015. Artificial immune system based web page classification. In: Silhavy R, Senkerik R, Oplatkova Z, et al. (Eds.), Software Engineering in Intelligent Systems. Springer, Cham, p.189-199.
https://doi.org/10.1007/978-3-319-18473-9_19

Onan A, 2016. Classifier and feature set ensembles for web page classification. *J Inform Sci*, 42(2):150-165.
https://doi.org/10.1177/0165551515591724

Panchekha P, Torlak E, 2016. Automated reasoning for web page layout. *ACM SIGPLAN Not*, 51(10):181-194.
https://doi.org/10.1145/3022671.2984010

Pritsos DA, Stamatatos E, 2013. Open-set classification for automated genre identification. European Conf on Information Retrieval, p.207-217.
https://doi.org/10.1007/978-3-642-36973-5_18

Qi XG, Davison BD, 2006. Knowing a web page by the company it keeps. Proc 15th ACM Int Conf on Information and Knowledge Management, p.228-237.
https://doi.org/10.1145/1183614.1183650

Qi XG, Davison BD, 2009. Web page classification: features and algorithms. *ACM Comput Surv*, 41(2):12.
https://doi.org/10.1145/1459352.1459357

Sze V, Chen YH, Yang TJ, et al., 2017. Efficient processing of deep neural networks: a tutorial and survey. *Proc IEEE*, 105(12):2295-2329.
https://doi.org/10.1109/JPROC.2017.2761740

Wei YL, Wang W, Wang BL, et al., 2017. A method for topic classification of web pages using LDA-SVM model. Chinese Int Automation Conf, p.589-596.
https://doi.org/10.1007/978-981-10-6445-6_64

Xue WM, Bao H, Huang WM, et al., 2006. Web page classification based on SVM. 6th World Congress on Intelligent Control and Automation, p.6111-6114.
https://doi.org/10.1109/WCICA.2006.1714255

Zhu J, Xie Q, Yu SI, et al., 2016. Exploiting link structure for web page genre identification. *Data Min Knowl Discov*, 30(3):550-575.
https://doi.org/10.1007/s10618-015-0428-8