Journal of Information & Computational Science 12:4 (2015) 1549–1556 March 1, 2015
Available at http://www.joics.com

A Web Page Classification Algorithm Based on Feature Selection ⋆

Hongfang Zhou [a,∗], Jie Guo [a], Xinyi Wang [a], Wencong Duan [a]
Peng Wang [a], Wenquan Cao [b]

[a] *School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China*
[b] *Weinan Vocational and Technical College, Weinan 714000, China*

Abstract

Massive relative information about the locations and distributions of feature words on web page is ignored in traditional web page classification algorithms. This paper gives a new feature selection method integrating with intra-class distribution ratio and inter-class deviation. Based on this, it proposes a new web page classification algorithm. The experimental results show it can improve the accuracy of classification greatly compared with traditional $TFIDF$ and GA algorithm.

Keywords: Web Page Classification; Location Feature; Feature Selection; Inter-class Deviation; Intra-class Distribution Ratio

1 Introduction

With the development of Internet, the quantity of web pages has grown crazily. But owing to the variety and complexity of web pages, it is difficult for users to obtain their required information exactly and timely. So web page classification arises, and it becomes a hot research topic in the field of machine learning after text classification [1].

At present, most researches relative to web page classification are mainly based on text classification, and the latter is based on text vectorization [2]. In the process of vectorization, the weight of a feature word is used for expressing its effect on describing text contents. Therefore, the weight calculation method of the feature words is an important factor of text classification. But the feature words selected by traditional methods sometimes have little effect on text classification. Furthermore, they may influence the accuracy of classification. Meanwhile, the dimensions of relative characteristics are quite high, and the required time training these samples

⋆Project supported by the National Science Foundation of China (No. 61402363, 61272284), Natural Science Foundation Project of Shaanxi Province (No. 2014JQ8361), Xi'an Science Program Project (No. CXY1339(5)), Beilin District of Xi'an Science and Technology Project (No. GX1405), and Program Project in Xi'an University of Technology (No. 116-211302).

∗Corresponding author.
Email address: zhouhf@xaut.edu.cn; 99582540@qq.com (Hongfang Zhou).

DOI: 10.12733/jics20105546

is too long. As a result, it is necessary to introduce an effective feature extraction method [3]. In recent years, a lot of researchers have studied the feature extraction techniques. Gong and Li optimized the traditional $TFIDF$ algorithm by means of semantics, word length and some other information [4, 5]. In virtue of the global searching capability of GA (Genetic Algorithm), Liu calculated the weight factor of the feature words in every classification [6]. Since the distribution of the feature words in every class is ignored, Zhang integrated the inter-class distribution of the feature words [7]. To some extent, these algorithms improve the rationality and validity of the feature words weight. But there are still two deficiencies. (1) The web page characteristics and the locations of the feature words in different tags are ignored when calculating the weight. (2) The situation of the feature words distributing uneven in a fixed classification is not considered. So this paper proposes a new web page classification algorithm $WCAFS$ (Web page Classification Algorithm based on Feature Selection).

2 Related Work

The main idea of $TFIDF$ algorithm [8] is explained as follows. The higher frequency a word has in a document, the stronger ability it will have to distinguish the contents. The broader scope a word has in document set, the lower ability it will have to distinguish the corresponding contents. Now, we give some definitions of related terms used in this paper.

Definition 1 *(Term Frequency) Term Frequency* (TF) *is the times that a feature word* t_k *appears in a document* d_i*, and is expressed by* $tf_{ik}(d_i)$*.*

On the premise of excluding stop words and some high-frequent words, the more times a feature word t_k appears in a document d_i, the stronger ability it will have to express the document d_i.

Definition 2 *(Document Frequency) Document Frequency* (DF) *is the number of documents which contain feature word* t_k *in the collection* D*, and is expressed by* $N(t_k, D)$*.*

Obviously, the larger value of $N(t_k, D)$ a feature word t_k has, the weaker ability it will have to represent the document d_i.

Definition 3 *(Inverse Document Frequency) Inverse Document Frequency* (IDF) *is a measurement reflecting how often a feature word* t_k *appears in the collection* D*, and is expressed by* IDF_k*.*

$$IDF_k = \log(\frac{N(D)}{N(t_k, D)}) \tag{1}$$

where $N(D)$ *is the number of documents in the collection* D*, and* $N(t_k, D)$ *is the document frequency of feature word* t_k *in the collection* D*.*

Clearly, IDF_k decreases when $N(t_k, D)$ increases. Concretely, the smaller value of $N(t_k, D)$ a feature word t_k has, the stronger ability it will have to represent the document d_i which is in the collection D.

Definition 4 *(Normalization) To reduce the restraint of the high frequency feature words to the low frequency feature words, the components should be normalized. After normalized, the calculation formula of $TFIDF$ is as follows [9].*

$$W_{ik}(d_i) = \frac{tf_{ik}(d_i) \times \log(\frac{N(D)}{N(t_k,D)} + L)}{\sqrt{\sum\limits_{k=1}^{n}(tf_{ik}(d_i))^2 \times (\log(\frac{N(D)}{N(t_k,D)} + L))^2}} \tag{2}$$

where L is an empirical value which is often assigned with 0.01, $tf_{ik}(d_i)$ is the term Frequency of feature word t_k in document d_i, $N(D)$ is the number of documents in the collection D, $N(t_k, D)$ is the document frequency of feature word t_k in the collection D, and n is the total number of feature words in document d_i.

3 WCAFS Algorithm and Related Concepts

Among the existing web page classification algorithms, the location of the feature word t_k, the distribution of the document containing the feature word t_k in each class, and the distribution of the feature word t_k in every document of a class are ignored. Especially, when intra-class distribution of the feature word t_k is uneven, the above algorithms will have considerable errors in calculating the weight of the feature words. So this paper proposes a new web page classification algorithm based on $TFIDF$. The algorithm calculates the weight of a feature word integrating with the relevant location, inter-class deviation and intra-class distribution ratio to avoid the situation that a feature word with no contribution to classification is assigned a larger weight. Now, we give some related knowledge of $WCAFS$ algorithm.

3.1 Weight Integrating with Location of the Feature Word

Web page is different from plain text file. It is a kind of semi-structured file containing a large number of links and tags [10]. The information in tags has different abilities to express web contents and different roles in web page classification. According to the practical experience and the earlier researchers [11], we think that information in location category-one has the strongest ability to express page contents, and it should be given the highest weight. Similarly, information in location category-two has a stronger ability to express page contents, so it should be given a higher weight. And the ability to express page contents in location-three is inferior to the formers, so it should be given a low weight. So we can get Eq. (3).

$$weight(p = 1) > weight(p = 2) > weight(p = 3) \tag{3}$$

where p is the location category.

Definition 5 *(Web Page Representation by VSM) The representation of web page d is*

$$V(d) = (t_1, w_1(d); \cdots; t_k, w_k(d); \cdots; t_n, w_n(d))$$

Here, t_k is a feature word on web page d, and $w_k(d)$ is the term frequency of t_k.

This paper improves the term frequency according to the location of the feature word on web page. The specific method to get a new term frequency $w_k(d)$ is multiplying the old term frequency by the corresponding weight based on the location category that the feature word belongs to.

3.2 Weight Integrating with Intra-class Distribution Ratio and Inter-class Deviation of the Feature Word

Definition 6 *(Inter-class Deviation) Inter-class Deviation* (ID) *means a feature word may appear in some classes and not appear in other classes. It is an uncertain measurement between classes, and is expressed by* ID_{kj}.

$$ID_{kj} = \frac{N(t_k, C_j)}{\sum\limits_{x=1}^{m} N(t_k, C_x)} \tag{4}$$

Here, $N(t_k, C_j)$ *is the number of documents which contain feature word* t_k *in class* C_j, $\sum\limits_{x=1}^{m} N(t_k, C_x)$ *is the number of documents which contain feature word* t_k *in all the classes, and* m *is the number of classes in the collection.*

As you can see, the larger ID_{kj} is, the stronger ability the feature word t_k will have to express class C_j.

Definition 7 *(Intra-class Distribution Ratio) Intra-class Distribution Ratio* (IDR) *is the probability that the feature word appears in all the documents in a class and is expressed by* IDR_{kj}.

$$IDR_{kj} = \frac{M(t_k, C_j)}{M(C_j)} \tag{5}$$

Here, $M(t_k, C_j)$ *is the total number that the feature word* t_k *appears in class* C_j, *and* $M(C_j)$ *is the total number that all the words appears in class* C_j.

As you can see, the larger IDR_{kj} is, the more homogeneous the distribution of feature word t_k appearing in class C_j will be and the stronger ability the feature word will have to express class C_j.

According to the above analysis, this paper proposes a feature weighting method integrating with location, intra-class distribution ratio and inter-class deviation of the feature word based on $TFIDF$. The formula is as follows.

$$W_{ik}(d_i) = \frac{tf_{ik}(d_i) \times \log(\frac{N(D)}{N(t_k,D)} + 0.01)}{\sqrt{\sum\limits_{k=1}^{n} (tf_{ik}(d_i))^2 \times (\log(\frac{N(D)}{N(t_k,D)} + 0.01))^2}} \times ID_{kj} \times IDR_{kj} \tag{6}$$

where $tf_{ik}(d_i)$ is the new modified term frequency according to the location of the feature word t_k on web page, $N(D)$ is the number of documents in the collection D, $N(t_k, D)$ is the document frequency of feature word t_k in the collection D, n is the total number of feature words in document

d_i, ID_{kj} is the inter-class deviation of feature word t_k, and IDR_{kj} is the intra-class distribution ratio of feature word t_k.

Under normal circumstance, the weight calculated by Eq. (6) can get better classification results. But when multiple classes contain the same feature word and the calculated weight is too large, the classification accuracy will be affected by certain. So this paper modifies the weight again after calculating by Eq. (6), and uses $W_{ik}^{'}(d_i)$ to express the modified weight. The modification method is counting up s, which is the sum of the feature words weight in each class, and then getting the weight, which is calculated as Eq. (6) and then divided by s. Through this, the impact on the classification results will be reduced. The modified formula is as follows.

$$W_{ik}^{'}(d_i) = \frac{W_{ik}(d_i)}{s} \tag{7}$$

The weight calculated by Eq. (7) reduces the impact on the classification results when the same feature word appears in different classes and the weight is large, meanwhile it doesn't influence the effect that the unique feature word in different classes has on classification.

3.3 Algorithm Description

Based on $TFIDF$, $WCAFS$ algorithm calculates the weight integrating with location, intra-class distribution ratio and inter-class deviation of the feature word. The pseudo codes is shown in Table 1.

Table 1: Pseudo codes of WCAFS

```
Algorithm: WCAFS
Input: DataSet, the name of classes
Output: the web page classification results
Step 1: Divide the DataSet into training set and testing set.
Step 2: Segment each document in the DataSet into words, and the word is expressed by $t$.
Step 3: for each word $t$ in training set do
            Compute the weight of the word $t$ according (6).
        end for
Step 4: for each word $t$ in training set do
            Modify the weight of the word $t$ according (7).
        end for
Step 5: Construct feature vector for each document in the training set.
Step 6: Classify the testing set with KNN classifier, and print the results.
```

4 Experiments and Analysis

The experiments are performed on a PC with operating system of Windows 7, an i3 CPU (2.40 GHz) and an 8G memory. The programming environment is JDK 1.6.

4.1 Experimental Data

The experimental data is from the Internet corpus SougouCS in sogou laboratory. In the experiments, as the number of web pages in some classes is too small, we only choose 12 classes including car, finance, IT, health, sports, tourism, education, culture, military, housing, entertainment and fashion. After analyzing and organizing, the dataset is divided into training set and testing set. The training set in each class has 500 web pages while the testing set has 300 web pages.

4.2 Classifier Selection

In the experiments, weighted nearest neighbor classifier (KNN) is used as basic classifier, where K is set to be 1. The similarity measure [12] we use for the classifier is as follows.

$$sim(d_i, C_j) = \frac{\sum\limits_{k=1}^{n} W_{ik} \times W_{jk}}{\sqrt{(\sum\limits_{k=1}^{n} W_{ik}^2)(\sum\limits_{k=1}^{n} W_{jk}^2)}} \tag{8}$$

where W_{ik} is the weight of the k_{th} feature word in document d_i, W_{jk} is the weight of the k_{th} feature word in class C_j, and n is the total number of feature words.

4.3 Performance Evaluation

Widespread metrics to assess performance classification are precision and recall. *Precision* can be defined as the probability that a retrieved instance is relevant to a given query and *Recall* as the probability to retrieve a relevant instance [13]. Given a class label τ_j they be respectively estimated as follows.

$$\widehat{p_j} = \frac{n_{++}^j}{n_{++}^j + n_{+-}^j} \tag{9}$$

$$\widehat{r_j} = \frac{n_{++}^j}{n_{++}^j + n_{-+}^j} \tag{10}$$

where n_{+-}^j is the number of documents for which $\tau_j \in f(x_i)$ but $\tau_j \notin T_i$ (false positives), n_{++}^j is the number of documents for which $\tau_j \in f(x_i)$ and actually $\tau_j \notin T_i$ (true positives), n_{-+}^j is the number of documents for which $\tau_j \in T_i$ but $\tau_j \notin f(x_i)$ (false negatives).

4.4 Experimental Results

To verify the accuracy of $WCAFS$ algorithm, we do experiments with $TFIDF$ algorithm, GA and $WCAFS$ algorithm. In the experiment of $WCAFS$ algorithm, when improving the term frequency according to the location category of the feature word, we think title is the direct description of web page subject and can express the main contents of the corresponding web pages, so it should be in location category-one and the corresponding weight is set to be 4. Description is a brief introduction to the web page and keywords are the essential words on web page. These two parts play the key roles in summarizing and emphasizing web page, so they should be in location category-two and the corresponding weight is set to be 2. Plain text is a general text whose ability to express the web page is inferior to the formers, so it should be in location category-three and the corresponding weight is set to be 1.

Fig. 1 and Fig. 2 show the precision and recall about the three algorithms.

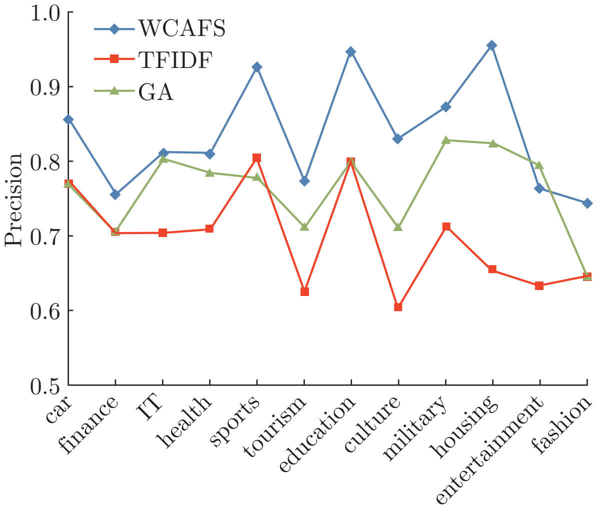

Fig. 1: Comparison of precision

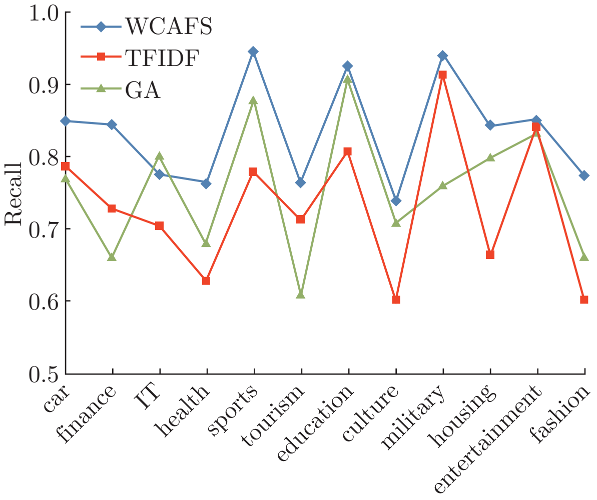

Fig. 2: Comparison of recall

From these two figures, we know the classification results using the improved feature selection algorithm are better than that using $TFIDF$ algorithm and GA. The precision and recall of classification in most classes have been improved. This suggests that intra-class distribution ratio and inter-class deviation of a feature word have both an effect on weight. So considering these two factors can effectively improve precision and recall of classification. At the same time, it also suggests that calculating weight considering the location of a feature word on web page can improve the accuracy of classification.

5 Conclusions

In view of the locations of the feature word are ignored in the process of classifying web pages, this paper proposes a web page classification algorithm integrating with its location. And at the same time, this paper also proposes a weight calculation method integrating with intra-class distribution ratio and inter-class deviation of the feature word based on $TFIDF$ algorithm. The experimental results show that the method can calculate the weight of the feature word effectively and improve the precision and recall of web page classification. All these verify that it is an effective method to calculate weight.

References

[1] Jun Lan, Huaji Shi, Xingyi Li, Min Xu, Associative web document classification based on word mixed weight, Computer Science, 38(3), 2011, 187-190

[2] Mingzhu Sun, Haiping Wei, Shaokun Dun, Juzhu Wang, A new feature selection method in SVM web page classification, Science Technology and Engineering, 11(6), 2011, 1359-1363

[3] Alper Kursat Uysal, Serkan Gunal, A novel probabilistic feature selection method for text classification, Knowledge-Based Systems, 36, 2012, 226-235

[4] Jing Gong, Jingye Zhou, A method for computing weight of text characteristic item based on multiple factors weighting, Computing Technology and Automation, 26(1), 2007, 81-83

[5] Yuanyuan Li, Yongqiang Ma, Text term weighting approach based on latent semantic indexing, Computer Applications, 28(6), 2009, 1460-1466

[6] Yanan Liu, Research of Feature Extraction Technology in KNN Text Classification Based on the Genetic Algorithm, Master Degree Thesis, China University of Petroleum, 2011

[7] Yufang Zhang, Xiaoli Chen, Zhongyang Xiong, Improved approach to weighting terms using information gain, Computer Engineering and Applications, 43 (35), 2007, 159-161

[8] G. Salton, T. Y. Clement, On the construction of effective vocabularies for information retrieval, Proc. of 1973 Meeting on Programming Languages and Information Retrieval, New York, USA: ACM Press, 1973

[9] Song Lu, Xiaoli Li, Shuo Bai, An improved approach to weighting terms in text, Journal of Chinese Information Processing, 14(6), 2000, 8-20

[10] Aixin Sun, Ying Liu, Ee-Peng Lim, Web classification of conceptual entities using co-training, Expert Systems with Applications, 38(12), 2011, 14367-14375

[11] M. Cutler, Yungming Shi, Weiyi Meng, Using the structure of HTML documents to improve retrieval, Proceeding of the USENIX Symposium on Internet Technologies and Systems Monterey, California, 1997, 22-23

[12] Deyi Tai, Jun Wang, Improved feature weighting algorithm for text categorization, Computer Engineering, 36(9), 2010, 0197-0200

[13] Yufang Zhang, Binhou Wan, Zhongyang Xiong, Research on feature dimension reduction in text classification, Application Research of Computers, 29(7), 2012, 2541-2543