EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Design and analysis of a general vector space model for data classification in Internet of Things

Jinguo Sang[1], Shanchen Pang[2*], Yang Zha[3] and Fan Yang[4*]

Abstract

The amount of information increases explosively in Internet of Things, because more and more data are sensed by large amount of sensors. The explosive growth of information makes it difficult to access information efficiently, so it is an effective method to decrease the amount of information to be transferred on network by text classification. This paper proposes a new text classification algorithm based on vector space model. This algorithm improves the feature selection and weighting methods by introducing synonym replacement to traditional text classification algorithms. The experimental results show that the proposed classification algorithm has considerably improved the precision and recall of classification.

Keywords: Internet of Things, Edge computing, Text classification, Feature selection, Feature weighting, Classification algorithm, Vector space model

1 Introduction

In the twenty-first century, Internet and cloud platforms have become the main method to access information with their penetration and popularization rapidly. With the advances in science and technology, human society has gradually entered the Internet of Things era from the Internet era, it has brought great convenience to our study, work, and life. However, with the advent of the Internet of Things era, cloud platforms are facing the challenges of massive equipment access, massive data, insufficient bandwidth, and high power consumption [1–3]. On the one hand, people enjoy the convenience brought by the abundant information on the network; on the other hand, they are also suffering from the puzzle of information explosion. For example, if you are not good at filtering useless information when searching for the information you need from within a large amount of information, a considerable amount of time is spent in filtering information [4–7], which leads to a significant increase in retrieval time.

As a result, edge computing began to come into people's sight. Edge computing is an open platform that integrates network, computing, storage, and application core competencies on the edge of the network near the source of objects or data. It migrates computing tasks from cloud computing centers to edge devices that generate source data [8, 9]. For a large amount of text information in the network world, it is also the main object of edge computing [10, 11]. Text classification, as an important subject of identifying and storing useful information, is widely used in information filtering, spam identification, text database, information retrieval, and the other application areas. With the explosive growth of the amount of information, efficient access to information has become difficult [12, 13] and text classification is becoming increasingly important.

Currently, text classification has attracted considerable research attention. By reasonably organizing and storing texts according to different categories, we can construct the corresponding training set. Then, the corresponding classification rules are defined according to the different categories, and a text classifier is constructed. Then, through the crawler and other technologies, we can constantly look for new texts in the network and add them to the text library. The research on text classification technology began earlier in foreign countries than in ours.

* Correspondence: pangsc@upc.edu.cn; sally01_yang@163.com
[2]College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266580, Shandong, China
[4]School of Information and Security Engineering, Zhongnan University of Economics and Law, Wuhan 430073, Hubei, China
Full list of author information is available at the end of the article

In 1959, Luhn [14] of IBM Company put forward the concept of word frequency statistics. In the same year, Maron [15] published a paper on automatic text classification. In the recent years, an increasing number of research results in this field have been published. Goudjil et al. [16] selected samples by using the posterior probability provided by the support vector machine (SVM) classifier and used the selected samples to classify data. Kim et al. [17] used a convolution neural network (CNN) to extract advanced features from the local text with a window filter. Research on text classification technology in China began later than in foreign countries. Li et al. [18] used the maximum entropy model as the text representation model to explore the text classification technology in depth. Dong et al. [19] deeply studied the relevant techniques of the feature weighting method and attempted to adjust the weight of the feature items by measuring the importance of different sentences in the text. Huang et al. [20] classified the text spam messages by using the advantages of the K-nearest neighbor (KNN) method. Jiang et al. [21] estimated the conditional probability of Bayes by using the deep feature weighted frequency in the training data and improved the classification performance.

However, there are still some challenges in text classification, and the accuracy of classification needs to be improved. In order to achieve better classification effect, in this paper, a new text classification algorithm based on the vector space model is proposed. We introduced the concepts of the inter-class concentration degree, intra-class dispersity degree, and intra-class frequency and proposed the feature selection method based on the inter-class concentration degree, intra-class dispersity degree, and intra-class frequency, which overcame the shortcomings of the current methods. And we developed the improved TF-IDF feature weighting method (I-TF-IDF), which resolved some disadvantages of the term frequency-inverse document frequency method (TF-IDF) in the weighting of different categories. Then, using the center classification method, we formed the text classifier. The comparative experiments results showed that the proposed algorithm was more accurate than traditional classification methods, and it not only shed a light on existing challenges in text classification but also inspire new methods.

The rest of this paper is organized as follows: Section 2 briefly introduces the related works in the literature. Section 3 is the method of the paper. Section 4 discusses the text classification algorithm based on VSM. Section 5 presents experimental results and discussion. Section 6 is the conclusion.

2 Related work

Over the years, scholars and experts at home and abroad have studied and explored different methods for text categorization.

With the advent of high dimensionality, adequate identification of relevant features of the data has become indispensable in real-world scenarios. In this context, the importance of feature selection is beyond doubt and different methods have been developed. However, with such a vast body of algorithms available, choosing the adequate feature selection method is not an easy-to-solve question and it is necessary to check their effectiveness on different situations. Nevertheless, the assessment of relevant features is difficult in real datasets and so an interesting option is to use artificial data. In [22], several synthetic datasets are employed for this purpose, aiming at reviewing the performance of feature selection methods in the presence of a crescent number or irrelevant features, noise in the data, redundancy and interaction between attributes, as well as a small ratio between number of samples and number of features. Brown et al. [23] present a unifying framework for information theoretic feature selection, bringing almost two decades of research on heuristic filter criteria under a single theoretical interpretation. While many hand-designed heuristic criteria try to optimize a definition of feature "relevancy" and "redundancy," the approach leads to a probabilistic framework which naturally incorporates these concepts. As a result, we can unify the numerous criteria published over the last two decades and show them to be low-order approximations to the exact (but intractable) optimization problem. The primary contribution is to show that a large empirical study provides strong evidence to favor certain classes of criteria, in particular those that balance the relative size of the relevancy/redundancy terms.

Chen et al. [24] introduce a wrapper method, namely cosine similarity measure support vector machines (CSMSVM), to eliminate irrelevant or redundant features during classifier construction by introducing the cosine distance into support vector machines (SVM). Traditionally, feature selection approaches typically extract features and learn SVM parameters independently or in the attribute space, which might result in a loss of information related to classification process or lead to the increase of classification error when introduce the kernel SVM. Comparing the novel method with well-known feature selection techniques with experiments, CSMSVM outperformed the other methodologies in improving the pattern recognition accuracy with fewer features.

Fung et al. [25] conduct an in-depth empirical analysis and argue that simply selecting the features with the highest scores may not be the best strategy. A highest scores approach will turn many documents into zero length, so that they cannot contribute to the training process. Accordingly, we formulate the feature selection process as a dual objective optimization problem and identify the best number of features for each document automatically. Extensive experiments are conducted to verify our claims. The encouraging results indicate our proposed framework is effective [26–28]. Mladenic et al. [29] describe an approach to feature subset selection that takes into account problem specifics and learning algorithm characteristics. It is developed for the Naive Bayesian classifier applied on text data, since it combines well with the addressed learning problems. They focus on domains with many features that also have a highly unbalanced class distribution and asymmetric misclassification costs given only implicitly in the problem. By asymmetric misclassification costs, we mean that one of the class values is the target class value for which we want to get predictions and we prefer false positive over false negative. The example problem is automatic document categorization using machine learning, where they want to identify documents relevant for the selected category.

Text categorization is a significant tool to manage and organize the surging text data. Many text categorization algorithms have been explored in previous literatures [30–32], such as KNN, Naive Bayes, and support vector machine. KNN text categorization is an effective but less efficient classification method. Jiang et al. [33] propose an improved KNN algorithm for text categorization, which builds the classification model by combining constrained one pass clustering algorithm and KNN text categorization. Empirical results on three benchmark corpora show that their algorithm can reduce the text similarity computation substantially and outperform the-state-of-the-art KNN, Naive Bayes, and support vector machine classifiers. Nielsen et al. [34] present a survey of the basic theory of the backpropagation neural network architecture covering architectural design, performance measurement, function approximation capability, and learning. The survey includes previously known material, as well as some new results, namely, a formulation of the backpropagation neural network architecture to make it a valid neural network (past formulations violated the locality of processing restriction) and a proof that the backpropagation mean-squared-error function exists and is differentiable. They present a speculative neurophysiological model illustrating how the backpropagation neural network architecture might plausibly be implemented in the mammalian brain for corticocortical learning between nearby regions of the cerebral cortex.

The idea behind the support vector network was previously implemented for the restricted case where the training data can be separated without errors. Cortes et al. [35] extend the result to non-separable training data. High generalization ability of support vector networks utilizing polynomial input transformations is demonstrated. They also compare the performance of the support vector network to various classical learning algorithms that all took part in a benchmark study of optical character recognition. Joachims [36] explores the use of support vector machines (SVMs) for learning text classifiers from examples. It analyzes the particular properties of learning with text data and identifies why SVMs are appropriate for this task. Empirical results support the theoretical findings. SVMs achieve substantial improvements over the currently best performing methods and behave robustly over a variety of different learning tasks. Furthermore, they are fully automatic, eliminating the need for manual parameter tuning. LIBSVM is a library for SVMs. The goal is to help users to easily apply SVM to their applications. LIBSVM has gained wide popularity in machine learning and many other areas. Chang et al. [37] present all implementation details of LIBSVM. Issues such as solving SVM optimization problems, theoretical convergence, multiclass classification, probability estimates, and parameter selection are discussed in detail.

The choice of the kernel function is crucial to most applications of SVMs. However, Leopold et al. [38] show that, in the case of text classification, term-frequency transformations have a larger impact on the performance of the SVM than the kernel itself. They discuss the role of importance weights (e.g., document frequency and redundancy), which is not yet fully understood in the light of model complexity and calculation cost, and they show that time-consuming lemmatization or stemming can be avoided even when classifying a highly inflectional language like German.

3 Method

3.1 Text preprocessing

The text belongs to unstructured data, which cannot be directly processed by a computer. It must go through a series of processing steps to transform the text into information that can be recognized and processed by the computer. In view of the particularity of Chinese, text preprocessing includes three steps: Chinese word segmentation, stop word processing, and synonym replacement.

English words are separated by spaces. Compared with English, the Chinese structure is more complex without any obvious separation mark between words. Therefore, the first step in text preprocessing is to carry out Chinese word segmentation by separating sentences in the text by words. Some of the common segmentation algorithms [39] are word segmentation based on semantic understanding, word segmentation based on probability statistics, and word segmentation based on specific string matching. A stop word is defined as a word that often appears in text but does not have a practical meaning. These words only serve as an auxiliary expression and are not distinguished strongly from one other; therefore, they need to be deleted before classification. Sebastiani et al. [40] pointed out that if a word in the current dataset exceeds the threshold set by the system, it is considered to be a stop word and deleted. The expression of Chinese is abundant. To increase the readability of the article, the author generally expresses the same meaning in a variety of ways. There are many synonyms in articles, which increase the diversity of words and spread the words out. Therefore, synonym replacement can make words more concentrated and improve the accuracy of text classification.

3.2 Feature selection

After preprocessing, the text will retain a number of feature items, which will lead to excessive dimensions, cause noise, and affect the accuracy of the calculation. Therefore, feature selection is important, which is the selection of feature items with a strong distinction between text categories from a large number of feature items. The existing feature selection methods have document frequency (DF), information gain (IG), mutual information (MI), chi-squared test, expected cross entropy (ECE), weight of evidence for text (WET), and so on. However, these commonly used feature selection methods have some shortcomings. It is reasonable for the information gain method to consider the situation that the feature items do not appear, but too much consideration of the negative correlation between feature items and the text will interfere with the classification. Moreover, the information gain method takes the training set as a whole and does not distinguish among different categories. The mutual information method ignores the frequency of feature items when calculating the number of texts in which the feature item appears, and it is easy to select some rare words. To overcome the shortcomings of these methods, in this paper, we introduce the concepts of inter-class concentration degree, intra-class dispersity degree, and intra-class frequency and propose a new feature selection method.

Definition 1. The inter-class concentration degree refers to the concentration of the feature items' distribution among different categories.

Definition 2. The intra-class dispersity degree refers to the uniformity of the feature items' distribution within a certain category.

Definition 3. The intra-class frequency refers to the frequency of feature items within a certain category, and the frequency is for all the texts in all the categories.

The representativeness of feature items to categories is determined by the inter-class concentration degree, intra-class dispersity degree, and intra-class frequency. The evaluation method of the inter-class concentration degree is that the feature items distributed in one or several categories have more classification information than those evenly distributed in most categories. The evaluation method of the intra-class dispersity degree is that the more evenly distributed the intra-class feature items in different texts are, the more representative they are of the categories. The evaluation method of the intra-class frequency is that the higher the feature item's frequency is in the categories, the more representative the feature item is of the categories. This does not mean frequency in a single text, because if a feature item appears more than once in a text of a certain category, it is only strongly related to this text but not the entire category. Therefore, the proposed algorithm based on the inter-class concentration degree, intra-class dispersity degree, and intra-class frequency (CDF) can be expressed as follows:

$$\begin{aligned}\mathrm{CDF}(t, C_i) &= \frac{N_{(t,C_i)}}{N_t} * \frac{N_{(t,C_i)}}{N_{C_i}} * \frac{n_{(t,C_i)}}{N_{C_i}} \\ &= \frac{{N_{(t,C_i)}}^2 * n_{(t,C_i)}}{N_t * {N_{C_i}}^2},\end{aligned} \tag{1}$$

where $N(t, Ci)$ refers to the number of texts containing the feature item t in category Ci. Nt refers to the number of texts containing the feature item t. NCi refers to the number of texts in category Ci. $n(\mathrm{t, Ci})$ refers to the frequency of feature item t in category Ci. Compared with the other feature selection methods, the CDF method is simple and intuitive, and the inter-class concentration degree, intra-class dispersity degree, and intra-class frequency are considered synthetically, which can obtain a better feature selection effect.

3.3 Feature weighting

After feature selection, the reserved feature items need to be weighted. The commonly used feature weighting methods include Boolean weighting, term frequency (TF), inverse document frequency (IDF), and TF-IDF. Compared with the other algorithms, the TF-IDF algorithm takes the TF and IDF methods into consideration, which is more reasonable. In the practical applications, its effect is obviously better than that of the Boolean weighting, TF, and IDF methods, and it is one of the most popular weighting algorithms at present. However, the TF-IDF method has some shortcomings, which can be improved to obtain a better feature weighting effect. The unreasonable part of the TF-IDF method is in the IDF module. In this module, only the distribution of feature items in the whole text set is counted, but the distribution of feature items in different text categories and in different texts within the same category is ignored.

According to the defects of the TF-IDF method, in this paper, we propose a new feature weighting method, which preserves the rationality of TF-IDF and combines the concepts of the inter-class concentration degree and the intra-class dispersity degree. The improved formula is as follows:

$$W_{it} = \frac{TF_{it}}{N_{C_i}} * \frac{N_{(t,C_i)}}{N_{C_i}} * \log_2\left(\frac{(N - N_t) * N_{(t,C_i)}}{N_t * \left(N_t - N_{(t,C_i)}\right) + \beta}\right), \tag{2}$$

where W_{it} refers to the weight of feature item t in text i. TF_{it} refers to the word frequency of feature item t in text i. $N(\mathrm{t}, Ci)$ refers to the number of texts containing the feature item t in category Ci. N_t refers to the number of texts containing the feature item t. NCi refers to the number of texts in category Ci. N refers to the total number of texts. β is set at 1.

3.4 Text classifier

The main function of the text classifier is to classify the text that needs to be classified into predefined different text categories according to the matching rules. At present, some commonly used text classification methods for the classifier are Bayesian algorithm, K-nearest neighbor algorithm, artificial neural network, support vector machine, and class center vector algorithm. These methods can form different classifiers combined with their own training set, and the classifiers will implement text classification after training. In this study, the class center vector method was used to design the classifier. The basic ideas of this method were as follows. In the training stage, feature selection and feature weighting were carried out for each text in the training set to obtain the text feature vectors. Then, the class center vector was calculated according to the feature vector of the text in each category. In the text classification stage, the text to be classified was also represented as a text feature vector and compared with the class center vector of each category. Then, the similarity between them was obtained, and the text to be classified was reduced to the category with the highest similarity. If a text could correspond to more than one category, then a threshold was set to attribute the text to a category whose similarity was greater than the threshold. The advantages of the class center vector method are its simple principle, easy implementation, and little calculation.

4 Text classification algorithm based on VSM

Text classification was divided into two stages. The first stage was the classifier construction phase. Combined with the corresponding classification algorithm, the classifier was trained by the texts classified in the training set, and the class center vectors were constructed. The second stage was the process of classifying the texts to be classified by using the classifier. The general flowchart of the text classification algorithm is shown in Fig. 1.

The text preprocessing modules of the two stages were the same. For the Chinese word segmentation, we adopted the ICTCLAS segmentation system of the Chinese Academy of Sciences after the experiment, which could be downloaded from Open Platform of Chinese NLP (www.nlp.org.cn). After word segmentation, the words in the texts were separated with spaces. For stop word processing, we adopted the stop word list of Harbin Institute of Technology. This stop word list was considerably sufficient for the processing of punctuation marks and special characters, took the whole text as a dimension to select feature items, and did not have to consider the division of paragraphs, sentences, and so on. The synonym replacement was carried out on the basis of HIT IR-Lab Tongyici Cilin (Extended), which counted only the synonymous words; it was not possible to replace the synonyms directly. For a set of synonyms, we used the first word as the base word and the others as the synonyms to construct a thesaurus. In the process of synonym replacement, if a synonym appeared, we replaced it with the base word.

The differences between the two stages were the feature selection module and the feature weighting module. In the classifier construction phase, the processing object were the texts that had been classified, and the CDF feature selection method and the I-TF-IDF feature weighting method could be used. In the text classification phase, the processing objects were the texts to be classified, and the frequency-based feature selection method and the TF weighting algorithm were used.

4.1 Classifier module design

The classifier module was the core of the whole algorithm, which was formed by combining the corresponding classification algorithm and the training text set after training, and the texts to be classified were also classified in the classifier module. The classification algorithm selected in this study was the class center vector algorithm, and the training set was the text classification corpus of Fudan University, which was divided into the test corpus and the training corpus and could be downloaded from Open Platform of Chinese NLP (www.nlp.org.cn). The test corpus contained 9833 documents, while the training corpus contained 9804 documents, and they were basically collected according to the ratio of 1:1. The training corpus was used to train the classifiers during the training process. The test corpus was used to carry out a classification test to evaluate the classifier effect after the completion of the classifier training. We took all the training corpus texts from training set for the Chinese word segmentation, stop word processing, and synonym replacement and marked the category of each text to form an initial text feature array. And the detailed algorithm for the classifier training was as follows:

Input: initial text feature array $D = (D1, C1; D2, C2; ...; Di, Ci; \ldots; DMCM)$, $Di = (T1, T2, ..., Tj, \ldots, T\mathrm{ki})$

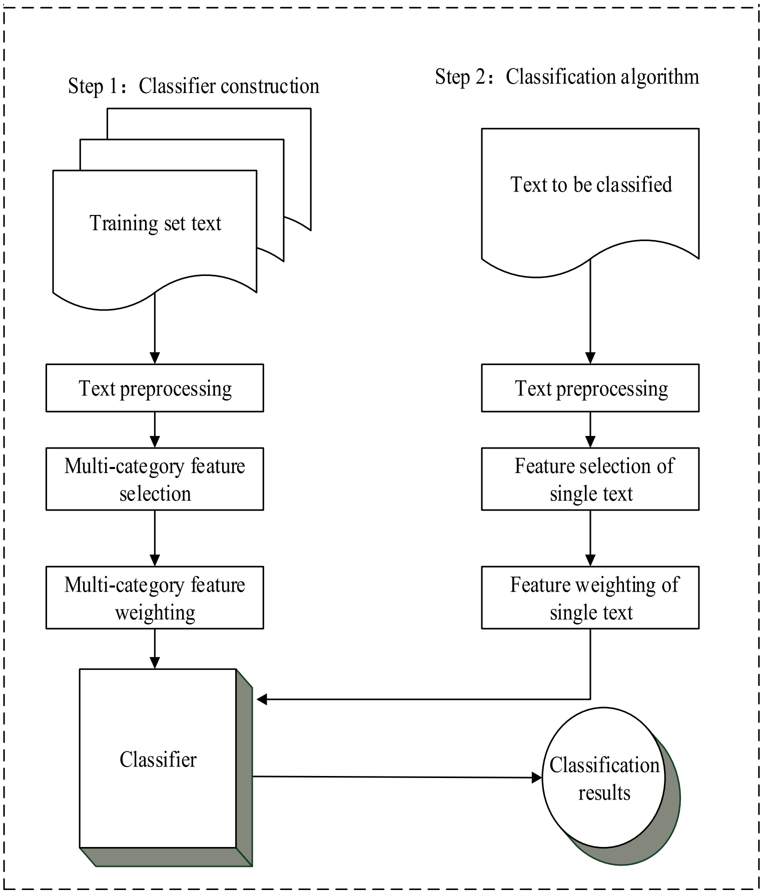
Step 1：Classifier construction
Step 2：Classification algorithm
Training set text
Text to be classified
Text preprocessing
Text preprocessing
Multi-category feature selection
Feature selection of single text
Multi-category feature weighting
Feature weighting of single text
Classifier
Classification results

Fig. 1 General flow chart of text classification algorithm. Text classification was divided into two stages. Step 1 represents the construction process of the classifier (including training set text, text preprocessing, multi-category feature selection, multi-category feature weighting). Step 2 represents the process of text categorization based on the classifier (including text to be classified, text preprocessing, feature selection of single text, feature weighting of single text) and gets the final classification results

```
Output :class center vector $C_t(T_1, W_1; T_2, W_2; \cdots; T_k, W_k)$
Parameter :M (the number of texts); $N$ (the number of categories); $n$ (the number of texts in category $C_t$); $k_i$ (the number of feature items in text $D_t$).
For $i$=1:$M$
For $j$=1:$k_i$
CDF$_j$( $T_j$, $C_i$)
End
select $k$ feature items with the largest CDF value: $dp$ = ($T$1, $T$2, ..., $Tq$, ..., $T$k)
End
text feature array: $d$ = ($d$1, $C$1; $d$2, $C$2; ...; $dp$, $Cp$; …; $dMCM$)
For p=1:M
For q=1:k
$Wq = WdpTq$
End
text feature vector $dp$ = ($T$1, $W$1; $T$2, $W$2; ...; $Tk$, $W$k)
End
For t=1:N
class center vector: $C_t(T_1, W_1; T_2, W_2; \cdots; T_k, W_k) = \frac{1}{n}\sum_{i=1}^{n} d_i(T_1, W_1; T_2, W_2; \cdots; T_k, W_k)$ (3)
End
```

Note that because the feature items of different texts were different, we had to synthesize the feature items in all of the texts of a certain category for averaging the feature vectors. Finally, we selected k feature items with the largest value as the class center vector. The class center vectors for all the categories were formed, which marked the end of the training of the text classifier to be used for the text classification.

4.2 Classification algorithm

We take the text to be classified, and implement Chinese word segmentation, stop word processing, and synonym replacement to form the initial text features. And the classification algorithm is as follows:

```
Input:initial text feature $D' = (T1, T2, ..., Tj, \ldots, Tm)$
Output :the category of the text to be classified$C'$
Parameter :m (the number of feature items in the text to be classified); N (the number of categories).
For i=1:m
    frequency-based method ($T_j$)
End
select feature items with the largest value: text feature $d' = (T1, T2, ..., Tj, ..., T\mathrm{k})$
For j=1:k
    $Wj = TFd'Tj$
End
text feature vector $d' = (T1, W1; T2, W2; ...; Tk, W\mathrm{k})$
For t=1:N
    $simt(d', Ct)$
End
the category of the text to be classified$C'$: withmax($simt$).
```

And the formula of the included angle cosine method is as follows:

$$Sim(\mathrm{d}_1, d_2) = \cos(d_1, d_2) = \frac{\sum_{i=1}^{k}(W_{1i} * W_{2i})}{\sqrt{\sum_{i=1}^{k} W_{1i}{}^2} * \sqrt{\sum_{i=1}^{k} W_{2i}{}^2}}, \tag{4}$$

where $W1$i indicates the weight of the ith feature item in text $d1$ and $W2$i indicates the weight of the ith feature item in text $d2$. After all these steps, the entire text classification algorithm flow was completed.

5 Results and discussion

We selected nine categories, namely art, history, aviation, computer, environment, agriculture, economy, politics, and sports, as the experimental training set and test set from the training and test texts of the text classification corpus of Fudan University, because the texts of these nine categories in the corpus were abundant and occupied a large proportion of the corpus, which was advantageous to form more accurate class center vectors. The experimental results reflected the real situation completely. The numbers of texts for the nine categories are presented in Table 1.

In this study, macro precision was taken as the measurement parameter and the feature items of different dimensions were selected for the contrast experiment. Different feature selection methods were used to measure the representativeness of the feature items in different ways, and the selection results were different. Some better feature selection methods might be able to select feature items that completely represent a text in a smaller dimension, while the poorer feature selection methods could only achieve the same at a higher dimension. When comparing different feature selection methods, we took the macro precision under the optimal feature dimension as the reference standard. In this study, we investigated the macro precision of four feature selection methods (IG, MI, ECE, and CDF) with different feature dimensions for the two feature weighting methods (TF-IDF and I-TF-IDF). With an increase in the feature dimensions, the trend of the macro precision was as follows.

Figure 2 shows that feature dimensions had a significant effect on the classification accuracy. With an increase in the feature dimensions, the macro precision of the classification algorithm also increased, but after reaching a certain threshold, the macro precision decreased instead. The best dimensions for the different feature selection methods were different. The best dimension for the IG method and the MI method was 450, that for ECE was 400, and that for CDF was 350. To achieve the best classification effect, we had to select the best feature dimension for the classification. And we found that for the same weighting method, the macro precision of the new CDF feature selection method was obviously higher than that of the other traditional feature selection methods. The experimental results showed that the proposed text classification method improved the macro precision as compared to the other methods.

To further study the performances of the CDF feature selection method and the I-TF-IDF feature weighting method, and to understand the performance of each text classification algorithm in the different categories, we conducted comparative experiments for the different feature selection methods and feature weighting methods in

Table 1 Numbers of training texts and test texts for different categories

Category	Number of training texts	Number of test texts	Category	Number of training texts	Number of test texts
Art	740	742	Agriculture	1021	1022
History	466	468	Economy	1600	1601
Aviation	640	642	Politics	1024	1026
Computer	1357	1358	Sports	1253	1254
Environment	1217	1218			

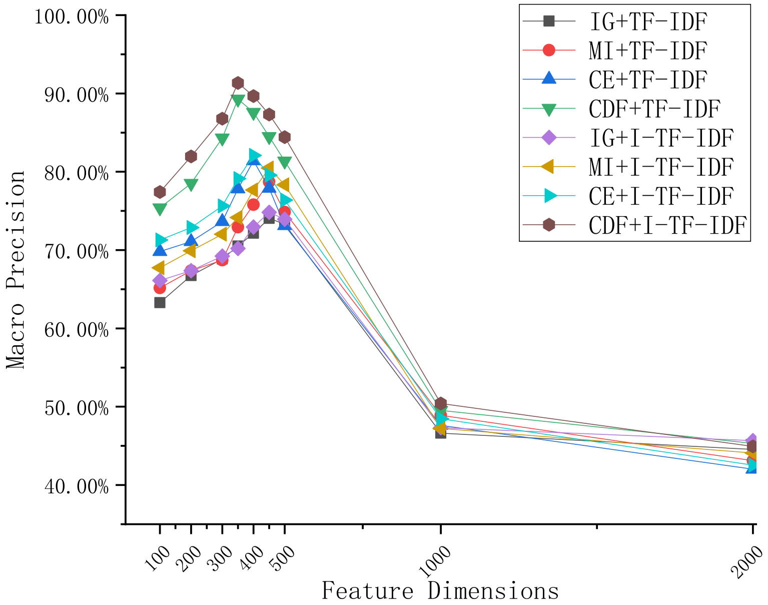

Fig. 2 Macro precision for different combinations of methods [41]. The best dimensions for the different feature selection methods were different (including IG, MI, ECE, CDF). The figure shows that feature dimensions had a significant effect on the classification accuracy. With an increase in the feature dimensions, the macro precision of the classification algorithm also increased, but after reaching a certain threshold, the macro precision decreased instead. The best dimension for the IG method and the MI method was 450, that for ECE was 400, and that for CDF was 350

this study. The precision, recall, and F1 values of each method in the different categories for the optimal feature dimension are presented in Tables 2 and 3.

Tables 2 and 3 show that the classification effects of the same feature selection method and the feature weighting method in the different categories were different. The classification effects of sports and computer were better, while the classification effects of economy and environment were relatively poor. By comparing the three feature selection methods, we found that the IG method was not effective, the MI method was slightly better, and the most effective method was CDF, which increased the precision and recall considerably. There were three groups of the contrast experiments of the TF-IDF and I-TF-IDF weighting methods. Each feature selection method implemented the classification experiments according to the TF-IDF and I-TF-IDF weighting methods. The experimental results showed that compared with the TF-IDF method, the I-TF-IDF weighting method improved the accuracy by approximately 2 percentage points.

6 Conclusion and future work

The requirement of text classification has become more and more important in the era of Internet of Things.

Table 2 Precision, recall, and F1 values for different combinations of methods (IG + TF-IDF, MI + TF-IDF, and CDF + TF-IDF) in the different categories

	IG + TF-IDF			MI + TF-IDF			CDF + TF-IDF		
	Precision (%)	Recall (%)	F1 value (%)	Precision (%)	Recall (%)	F1 value (%)	Precision (%)	Recall (%)	F1 value (%)
Art	70.56	73.69	72.09	76.34	79.01	77.65	87.28	88.01	87.64
History	72.80	76.87	74.78	71.28	80.46	75.59	88.23	87.27	87.75
Aviation	71.19	74.20	72.66	75.79	76.39	76.09	89.74	90.22	89.98
Computer	86.65	81.32	83.90	91.82	89.22	90.50	96.63	95.29	95.96
Environment	62.31	53.68	57.67	68.67	72.43	70.50	83.91	85.82	84.85
Agriculture	71.22	69.06	70.12	79.90	77.75	78.81	89.21	87.87	88.53
Economy	64.96	73.49	68.96	70.32	79.02	74.42	82.90	84.94	83.91
Politics	77.58	79.38	78.47	82.13	79.84	80.97	88.07	89.71	88.88
Sports	89.13	82.17	85.51	92.21	89.83	91.00	97.46	96.63	97.04

Table 3 Precision, recall, and F1 values for different combinations of methods (IG + I-TF-IDF, MI + I-TF-IDF, and CDF + I-TF-IDF) in the different categories

	IG + I-TF-IDF			MI + I-TF-IDF			CDF + I-TF-IDF		
	Precision (%)	Recall (%)	F1 value (%)	Precision (%)	Recall (%)	F1 value (%)	Precision (%)	Recall (%)	F1 value (%)
Art	71.32	75.63	73.41	77.92	80.12	79.00	88.76	87.23	87.99
History	72.91	73.28	73.09	73.01	74.28	73.64	90.37	89.84	90.10
Aviation	71.83	76.98	74.32	78.24	76.41	77.31	91.02	92.19	91.60
Computer	87.72	83.10	85.35	93.27	92.10	92.68	98.74	97.98	98.36
Environment	63.26	69.56	66.26	70.41	74.09	72.20	86.39	89.01	87.68
Agriculture	72.35	76.87	74.54	81.28	82.16	81.72	92.28	94.42	93.34
Economy	66.50	70.94	68.65	72.19	76.83	74.44	85.25	87.67	86.44
Politics	78.09	77.02	77.55	83.94	79.38	81.60	90.13	91.96	91.03
Sports	89.45	86.73	88.06	93.97	92.67	93.32	99.04	98.73	98.88

This paper proposes a new text classification algorithm to decrease the amount of information to be transferred on network. This algorithm introduces synonym replacement during text preprocessing phase and develops CDF method on the basis of existing feature selection. Further, the algorithm gives I-TF-IDF method by solving the disadvantages of TF-IDF feature weighting method.

The experimental results show that the proposed algorithm is more precise than existing algorithm, so it provides a better algorithm to decrease the amount of information. However, the text classification corpus of Fudan University used in this paper is not comprehensive.

In the future, the following areas need to be improved upon. Firstly, a fuller and more representative large-scale corpus needs to be built. Further, the setting of more categories and category levels needs to be studied further.

Abbreviations

CDF: Inter-class concentration degree, intra-class dispersity degree, and intra-class frequency; CNN: Convolution neural network; CSMSVM: Cosine similarity measure support vector machines; DF: Document frequency; ECE: Expected cross entropy; IBM: International Business Machines Corporation; IDF: Inverse document frequency; IG: Information gain; I-TF-IDF: Improved term frequency-inverse document frequency; KNN: K-nearest neighbor; LIBSVM: A library for SVMs; MI: Mutual information; SVM: Support vector machine; SVMs: Support vector machines; TF: Term frequency; TF-IDF: Term frequency-inverse document frequency; WET: Weight of evidence for text

Acknowledgements

Not applicable

Authors' contributions

JS conceived the framework design, performed the data analysis, and took charge of all the work of paper submission. SP reviewed and revised the manuscript. YZ and FY improved the work. All authors read and approved the final manuscript.

Funding

This work was supported by National Natural Science Foundation of China (61572522, 61402187, and 61502535) and Science and Technology Project of Huangdao District, Qingdao City, Shandong Province (2014-1-43).

Availability of data and materials

The datasets used or analyzed during the current study are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, Shandong, China. [2]College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266580, Shandong, China. [3]Beijing Baidu Netcom Science and Technology Co., Ltd., Beijing 100000, China. [4]School of Information and Security Engineering, Zhongnan University of Economics and Law, Wuhan 430073, Hubei, China.

Received: 20 August 2019 Accepted: 17 October 2019
Published online: 02 December 2019

References

1. N. Xiong, A.V. Vasilakos, J. Wu, Y.R. Yang, A. Rindos, Y. Zhou, W.Z. Song, *A self-tuning failure detection scheme for cloud computing service, IEEE 26th Parallel & Distributed Processing Symposium (IPDPS)* (2012). https://doi.org/10.1109/IPDPS.2012.126
2. Y. Zeng, C.J. Sreenan, N. Xiong, L.T. Yang, J.H. Park, Connectivity and coverage maintenance in wireless sensor networks. J Supercomput **52**(1), 23–46 (2010). https://doi.org/10.1007/s11227-009-0268-7
3. Z. Wang, T. Li, N. Xiong, Y. Pan, A novel dynamic network data replication scheme based on historical access record and proactive deletion. J Supercomput **62**(1), 227–250 (2012). https://doi.org/10.1007/s11227-011-0708-z
4. W. Guo, N. Xiong, A.V. Vasilakos, G. Chen, C. Yu, Distributed k–connected fault–tolerant topology control algorithms with PSO in future autonomic sensor systems. Int J Sens Netw **12**(1), 53–62 (2012). https://doi.org/10.1504/IJSNET.2012.047720
5. Y. Yang, N. Xiong, N.Y. Chong, X. Défago, A decentralized and adaptive flocking algorithm for autonomous mobile robots. Grid and Pervasive Computing Workshops (GPC) (2008). https://doi.org/10.1109/GPC.WORKSHOPS.2008.18
6. J. Yin, W. Lo, S. Deng, Y. Li, Z. Wu, N. Xiong, Colbar: a collaborative location-based regularization framework for QoS prediction. Inform Sciences. **265**, 68–84 (2014). https://doi.org/10.1016/j.ins.2013.12.007
7. J. Li, N. Xiong, J.H. Park, C. Liu, M.A. Shihua, S.E. Cho, Intelligent model design of cluster supply chain with horizontal cooperation. J Intell Manuf **23**(4), 917–931 (2012). https://doi.org/10.1007/s10845-009-0359-6
8. Y. Deng, H. Hu, N. Xiong, W. Xiong, L. Liu, A general hybrid model for chaos robust synchronization and degradation reduction. Inform Sciences. **305**, 146–164 (2015). https://doi.org/10.1016/j.ins.2015.01.028
9. C. Lin, Y.X. He, N. Xiong, *An energy-efficient dynamic power management in wireless sensor networks, Fifth International Symposium on Parallel and Distributed Computing* (2006). https://doi.org/10.1109/ISPDC.2006.8
10. H. Cheng, Z. Su, N. Xiong, Y. Xiao, Energy-efficient node scheduling algorithms for wireless sensor networks using Markov Random Field model. Inform Sciences **329**, 461–477 (2016). https://doi.org/10.1016/j.ins.2015.09.039

11. Y. Sang, H. Shen, Y. Tan, N. Xiong, Efficient protocols for privacy preserving matching against distributed datasets. International Conference on Information and Communications Security (ICICS) (2006). https://doi.org/10.1007/11935308_15
12. Y. Shi, R. Enami, J. Wensowitch, J. Camp, UABeam: UAV-based beamforming system analysis with in-field air-to-ground channels, 2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), (2018). DOI: https://doi.org/10.1109/SAHCN.2018.8397110
13. X. Liu, X. Zhang, M. Jia, L. Fan, W. Lu, X. Zhai, 5G-based green broadband communication system design with simultaneous wireless information and power transfer. Physical Communication **28**, 130–137 (2018). https://doi.org/10.1016/j.phycom.2018.03.015
14. H.P. Luhn, *Auto-encoding of documents for information retrieval systems* (Pergamon Press, London England, 1959)
15. M.E. Maron, J.L. Kuhns, On relevance, probabilistic indexing and information retrieval. J ACM **7**(3), 216–244 (1960)
16. M. Goudjil, M. Koudil, M. Bedda, A novel active learning method using SVM for text classification. Int. J. Autom. Comput. **15**(03), 44–52 (2018). https://doi.org/10.1007/s11633-015-0912-z
17. Y. Kim, Convolutional neural networks for sentence classification. (e-print arXiv). DOI:1408.5882, Accessed 3 Sep 2014.
18. R.L. Li, J.H. Wang, Z.Y. Chen, X.P. Tao, Y.F. Hu, Using maximum entropy model for Chinese text categorization. Journal of Computer Research and Development **42**(1), 94–101 (2005)
19. X.G. Dong, L.G. Gan, Feature weighting based on the importance of sentence. Computer & digital engineering. 34(8), 34-37,58(2006).DOI: https://doi.org/10.3969/j.issn.1672-9722.2006.08.011
20. W.M. Huang, Y. Mo, Chinese spam message filtering based on text weighted KNN algorithm. Computer engineering **43**(3), 193–199 (2017). https://doi.org/10.3969/j.issn.1000-3428.2017.03.033
21. L. Jiang, C. Li, S. Wang, Deep feature weighting for naive Bayes and its application to text classification. Eng Appl Artif Intel. **52**, 26–39 (2016). https://doi.org/10.1016/j.engappai.2016.02.002
22. V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, A review of feature selection methods on synthetic data. Knowl Inf Syst **34**(3), 483–519 (2013). https://doi.org/10.1007/s10115-012-0487-8
23. G. Brown, A. Pocock, M.J. Zhao, M. Luján, Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. Journal of machine learning research **13**(1), 27–66 (2012). https://doi.org/10.1080/00207179.2012.669851
24. G. Chen, J. Chen, A novel wrapper method for feature selection and its applications. Neurocomputing **159**, 219–226 (2015). https://doi.org/10.1016/j.neucom.2015.01.070
25. P.C.G. Fung, F. Morstatter, H. Liu, *Feature selection strategy in text classification, Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining* (2011). https://doi.org/10.1007/978-3-642-20841-6_3
26. L. Jian, J. Li, K. Shu, H. Liu, *Multi-label informed feature selection* (International Joint Conference on Artificial Intelligence AAAI Press, 2016)
27. L.S. Larkey, *Automatic essay grading using text categorization techniques, SIGIR '98 Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval.p90-95* (1998). https://doi.org/10.1145/290941.290965
28. A Learner-Independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization, in Text databases & document management ed. By M.F. Caropreso, S. Matwin, F. Sebastiani, (IGI Publishing Hershey, USA, 2001), 78-102.
29. D. Mladenic, M. Grobelnik, *Feature selection for unbalanced class distribution and Naive Bayes, Proceedings of the Sixteenth International Conference on Machine Learning* (1999), pp. 258–267
30. D.M. Zhang, *Research on key problems in text sentiment classification and opinion summarization* (Shandong University, 2012)
31. A. McCallum, K. Nigam, *A comparison of event models for Naive Bayes text classification, AAAI-98 workshop on learning for text categorization* (1998)
32. Y.X. Liao, *Research on text classification and its feature dimension reduction* (Zhejiang University, 2012)
33. S. Jiang, G. Pang, M. Wu, An improved K-nearest-neighbor algorithm for text categorization. EXPERT SYST APPL **39**(1), 1503–1509 (2012). https://doi.org/10.1016/j.eswa.2011.08.040
34. H. Nielsen, *Theory of the backpropagation neural network, International 1989 Joint Conference on Neural Networks* (2002). https://doi.org/10.1109/IJCNN.1989.118638
35. C. Cortes, V. Vapnik, Support-vector networks. Machine learning **20**(3), 273–297 (1995)
36. T. Joachims, Text categorization with support vector machines: learning with many relevant features. European Conference on Machine Learning, 137–142 (1998). https://doi.org/10.1007/BFb0026683
37. C.C. Chang, C. J Lin. LIBSVM: a library for support vector machines. Acm T Intel Syst Tec, 2(3), 27(2011). DOI:https://doi.org/10.1145/1961189.1961199
38. E. Leopold, J. Kindermann, Text categorization with support vector machines. How to represent texts in input space. Machine Learning **46**(1-3), 423–444 (2002). https://doi.org/10.1023/a:1012491419635
39. A. Wu, *Word segmentation in sentence analysis* (International Conference on Chinese Information Processing, 1998)
40. F. Sebastiani, Machine learning in automated text categorization. ACM computing surveys **34**(1), 1–47 (2002). https://doi.org/10.1145/505282.505283
41. I. Hmeidi, B. Hawashin, E. El-Qawasmeh, Performance of KNN and SVM classifiers on full word Arabic articles. Advanced Engineering Informatics **22**(1), 106–111 (2008). https://doi.org/10.1016/j.aei.2007.12.001

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.