Postprint: Chinese Very Short Text Classification Model Based on Support Vector Machines

Authors: Wang Yang, Xu Shanshan, Li Chang, Ai Shicheng, Zhao Chuanxin, Zhen Lei, Huang Shaofen

Date: 2018-12-13T00:00:00+00:00

Abstract

With the continuous proliferation of intelligent terminal devices, information rich in extremely short text data, such as WeChat messages, instant online news, and e-commerce customer product reviews, has experienced explosive growth. To effectively extract key feature information from extremely short texts, this paper proposes an extremely short text classification model based on Support Vector Machine (SVM). First, the original data undergoes data cleaning and is processed using Jieba segmentation; the processed data is then stored in a database, and text features are extracted via TF-IDF; simultaneously, Support Vector Machine is utilized to classify the extremely short texts. The effectiveness of the model is verified through (1-0) testing. The experiment employs 9,906 extremely short text data entries from the Wuhu City Social Management Platform as samples for algorithm testing and analysis. Results demonstrate that in terms of classification accuracy, this method achieves effective improvement compared to traditional methods such as Naive Bayes, Logistic Regression, and Decision Trees; furthermore, the matching results exhibit greater balance between misclassification rate and precision metrics.

Full Text

Preamble

Classification Model Based on Support Vector Machine for Chinese Extremely Short Text

Wang Yang†, Xu Shanshan, Li Chang, Ai Shicheng, Zhang Weidong, Zhen Lei, Meng Dan
(School of Information & Computer Science, Anhui Normal University, Wuhu, Anhui 241000, China)

Abstract: With the increasing popularization of intelligent terminal devices, information containing abundant extremely short text data—such as WeChat messages, online instant news, and e-commerce customer product reviews—has been experiencing explosive growth. To effectively extract key feature information from extremely short texts, this paper proposes an extremely short text classification model based on Support Vector Machine (SVM). The model first performs data cleansing on the raw data and processes it using Jieba segmentation; the processed data is then stored in a database, and text features are extracted through TF-IDF. Simultaneously, SVM is employed to classify the extremely short texts. The effectiveness of the model is verified through (1-0) testing. Experiments conducted on 9,906 extremely short text samples from the Wuhu City Community Management Platform demonstrate that the proposed method effectively improves classification accuracy compared to traditional methods such as Naive Bayes, Logistic Regression, and Decision Tree. Moreover, the results show more balanced performance in terms of misclassification degree and precision metrics.

Key words: support vector machine; Jieba segmentation; extremely short text classification; TF-IDF

0 Introduction

The widespread adoption of various intelligent terminals and social software has enabled users to express their views on social hotspots and government actions through increasingly diverse, convenient, and extensive channels. Many of these evaluations are expressed in the form of incomplete, extremely short texts. Extracting valuable information quickly from such incomplete texts is critically important for decision-makers. In today' s era of big data and flourishing self-media, people have grown accustomed to conveying emotions and expressing demands through brief, concise social media posts and succinct problem feedback on platforms like Twitter, Facebook, Weibo, and WeChat. This textual form exhibits fragmented and instantaneous characteristics, making it difficult for traditional text classification methods to rapidly extract information from such content. This paper proposes an extremely short text classification model based on Support Vector Machine.

Existing text classification methods primarily include two approaches: (a) Clustering word embedding method, which applies a k-means algorithm to word vectors of documents to obtain a fixed-size cluster set. Each text is represented as a bag of super-word embeddings, and text classification is derived by calculating the frequency of each super-word embedding in the respective text [**?**]. (b) Frequency-weighted method, which accounts for missing terms, calculates weights for existing terms, and combines them with an SVM classifier to achieve optimal classification performance [**?**].

In text classification research, Support Vector Machine has been widely applied. Current SVM-based text classification techniques mainly include the following:

(a) Improved hybrid kernel function classification method, which enhances classification effectiveness by recombining kernel functions with strong learning capabilities and those with good generalization ability into a hybrid kernel function [**?**]; (b) SVM classification method based on incremental learning, which fully considers the impact of new samples on initial samples, introduces boundary support vectors, and proposes an incremental learning algorithm based on boundary support vectors, achieving certain improvements in training speed and accuracy [**?**]; (c) Feature selection classification model, which addresses the limitations of traditional chi-square feature selection methods by proposing a new intra-class information optimization chi-square statistical feature selection method, effectively improving the model' s feature selection capability [**?**].

1.1 Extremely Short Text

In the narrow sense, text refers to the manifestation of written language, which from a literary perspective typically comprises a sentence or combination of sentences with complete and systematic meaning. In the broad sense, text refers to any discourse fixed by writing. Building upon the narrow definition, text with a length not exceeding 160 characters is referred to as short text [**?**], such as content from Weibo posts, NetEase Cloud comments, Chinese spam messages, and spam emails—the primary subjects of current text classification research. With the development of information technology and the acceleration of life pace, a new type of text has emerged that uses even more concise language to describe things: extremely short text (EST). We provide the following definition:

Definition 1. Extremely short text refers to the manifestation of written language that may contain objective statements or evaluative suggestions, does not necessarily have complete or systematic meaning, consists of several words or phrases, and generally has a sentence length not exceeding 15 characters.

Extremely short texts primarily originate from the Internet and are characterized by large volume, strong noise, and extremely sparse content features [**?**]. Examples in daily life include error reports for shared bicycles, brief product reviews on Taobao, and case reports submitted through community management platforms. Effectively identifying and classifying extremely short texts to achieve rapid processing of their content holds significant importance for data applications, corporate management, and government decision-making.

1.2 Enhanced Feature Vector

In analyzing extremely short texts, word segmentation and feature word selection are particularly crucial for subsequent research. Due to the brevity of the text, generally only 3–4 keywords can be extracted from the known content. Obviously, if a model is built based solely on these feature words, the information would be insufficient to guarantee result accuracy. Therefore, this paper proposes a feature word augmentation model.

Using social garbage information management as an example to illustrate the expansion of feature words in this model: First, analyze the extremely short texts from social management case reports and extract features to form a feature vector $B = (B_1, B_2, \cdots, B_u)$, where the value of u is small, generally not exceeding 4. Second, further analysis of the text reveals that words such as "water floating," "green belt," and "road surface" describe the location information of garbage, which can be summarized as a new feature word, denoted as B_{u+1}. By analogy, when B_{u+m} is obtained for $m \geq 5$, the feature vector becomes an enhanced feature vector $B = (B_1, B_2, \cdots, B_u, B_{u+1}, \cdots, B_{u+m})$. When the value of m is greater than or equal to 5, the feature vector possesses strong representational capability.

1.3 Text Preprocessing

As shown in [Figure 1: see original paper], text preprocessing proceeds in three steps: (a) Load raw data and separate the mixed text from its corresponding category labels; (b) Filter stop words—considering that the raw data contains severe colloquialism and numerous meaningless stop words, stop word removal is necessary, using the "Harbin Institute of Technology Stop Word List" [**?**]; (c) Use the Jieba segmentation tool to perform word segmentation on the cleaned text.

1.4 TF-IDF Feature Extraction

After preprocessing the text, the TF-IDF feature extraction method is employed to extract keywords from the resulting text for modeling. TF-IDF evaluates the importance of a word or term for a document within a file set or corpus. For a feature word w, its feature extraction function is:

$$f(w) = TF(w) \times IDF(w) \tag{1}$$

where TF (term frequency) refers to the ratio of the occurrence count of a feature term (which can be a character or word) in a text to the total occurrence count of all feature terms in that text. If a feature term appears frequently in a text, it indicates that the term may well describe the main information of the text and is suitable for classification. The main idea behind selecting IDF (inverse document frequency) as another factor is that feature words contained in only a few texts are more important and more conducive to distinguishing text categories than those contained in many texts. The fewer texts in the corpus that contain feature word w, the better w can differentiate categories. IDF can reduce the importance of feature words that appear in many texts while enhancing the importance of those contained in only a few texts.

Therefore, term frequency TF and inverse document frequency IDF are often used in combination. IDF is commonly calculated using Equation (2):

$$IDF(w) = \log\left[\frac{N}{n(w)} + 1\right] \tag{2}$$

where N is the total number of texts, and $n(w)$ is the number of texts containing w.

The TF-IDF feature extraction method calculates the TF-IDF weight value for each feature word in the text using Equation (1), sorts them in descending order, and then selects the top n feature words that meet predetermined screening conditions, thereby achieving dimensionality reduction of the original feature space.

1.5 Support Vector Machine

Through the aforementioned text classification method, several feature words for extremely short texts are determined. In SVM, for samples to be classified, we seek a so-called optimal hyperplane that maximizes the margin between samples, which greatly helps improve the generalization ability of the SVM classifier and enhance its prediction accuracy for unknown samples [**?**].

Taking two-dimensional linearly separable data as an example to discuss the construction of an SVM classifier, as shown in [Figure 2: see original paper]. Assume we have P linearly separable samples $\{(X_1, d_1), (X_2, d_2), \cdots, (X_p, d_p)\}$ where $d_p \in \{-1, 1\}$. For an input sample X_p, we expect to output its classification result d_p.

The hyperplane equation is defined as:

$$W_0^T X + b_0 = 0 \tag{3}$$

where X is the input, W is the weight vector, and b is the bias. Then any training sample satisfies:

$$d_p(W^T X_p + b) \geq 1 \tag{4}$$

When equality holds, sample points are distributed near the hyperplane and are called support vectors. To find the maximum-margin hyperplane (optimal hyperplane), based on analytic geometry knowledge, we define the distance from any point X in the sample space to the optimal hyperplane as:

$$r = \frac{W^T X + b}{\|W\|} \tag{5}$$

From Equation (5), the algebraic distance from support vectors to the hyperplane is:

$$r = \frac{1}{\|W\|} \tag{6}$$

From the above, to find the optimal hyperplane, we only need to minimize $\frac{1}{2}\|W\|^2$. The optimization problem can then be transformed into finding:

$$\min \frac{1}{2}\|W\|^2 \tag{7}$$

subject to the constraints in Equation (4). Introducing the Lagrange function:

$$L(W, b, \alpha) = \frac{1}{2}\|W\|^2 - \sum_{p=1}^{P} \alpha_p [d_p (W^T X_p + b) - 1] \tag{8}$$

The problem now becomes finding the minimum of the Lagrange function. Taking partial derivatives with respect to W and b and setting them to zero:

$$\frac{\partial L}{\partial W} = 0 \Rightarrow W = \sum_{p=1}^{P} \alpha_p d_p X_p \tag{9}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{p=1}^{P} \alpha_p d_p = 0 \tag{10}$$

Combining Equations (8) and (9) yields:

$$L(\alpha) = \sum_{p=1}^{P} \alpha_p - \frac{1}{2} \sum_{p=1}^{P} \sum_{j=1}^{P} \alpha_p \alpha_j d_p d_j X_p^T X_j \tag{11}$$

According to Equations (8) and (10), we obtain:

$$\max Q(\alpha) = \sum_{p=1}^{P} \alpha_p - \frac{1}{2} \sum_{p=1}^{P} \sum_{j=1}^{P} \alpha_p \alpha_j d_p d_j X_p^T X_j \tag{12}$$

After determining the optimal value of α, combining Equations (3) and (9) yields W and b, and the optimal classification discriminant function can be obtained as:

$$f(X) = \operatorname{sgn}(W^T X + b) = \operatorname{sgn}\left(\sum_{p=1}^{P} \alpha_p d_p X_p^T X + b \right) \tag{13}$$

For linearly non-separable data, it is mapped to a high-dimensional feature vector space. With an appropriate mapping function and sufficiently high-dimensional feature space, most nonlinearly separable patterns can be converted to linearly separable patterns in the feature space.

1.3.1 Word Segmentation Processing

Jieba segmentation is a Chinese word segmentation tool developed using the Python language. It has three main characteristics: (a) It supports three segmentation modes: precise mode, full mode, and search engine mode; (b) It supports traditional Chinese segmentation; (c) It supports custom dictionaries.

The implementation of Jieba segmentation is based on three principles: (a) Efficient word graph scanning based on Trie tree structure to generate a directed acyclic graph (DAG) of all possible word formations from Chinese characters in a sentence; (b) Dynamic programming to find the maximum probability path and identify the word frequency-based optimal segmentation combination; (c) For unknown words, it employs the Viterbi algorithm and an HMM model based on the word-forming ability of Chinese characters.

This paper adopts the precise mode of Jieba segmentation. This mode is the most fundamental and natural mode in Jieba segmentation, attempting to segment sentences as accurately as possible, making it suitable for extremely short text analysis.

2.1 Algorithm Implementation

Combining the SVM model, the processing flow for extremely short texts is shown in [Figure 3: see original paper]. After basic processing, textual information still cannot be recognized by computers, and the contribution of each word to classification remains unclear. Therefore, a method must be selected for feature extraction to strengthen the influence of feature words while weakening the interference of non-feature words. TF-IDF is a typical text feature extraction algorithm that effectively marks the contribution of words to classification through a combination of term frequency and inverse document frequency. Before training the classifier, the data is randomly divided into training and test sets at a 70%:30% ratio. After training the classifier using the training set, the test set is input for validation.

2.2 Kernel Function Selection

When constructing an SVM classification model, selecting an appropriate kernel function is crucial. For inner product kernel functions, the following four types are commonly used:

a) Linear kernel function (Linear):

$$K(X_i, X_j) = X_i \times X_j \tag{14}$$

b) Polynomial kernel function (Poly):

$$K(X_i, X_j) = [(X_i \cdot X_j) + 1]^d \tag{15}$$

where d is the degree of the highest term in the polynomial kernel function.

c) Radial Basis Function kernel (RBF):

$$K(X_i, X_j) = \exp(-\gamma \|X_i - X_j\|^2) \tag{16}$$

where γ is the parameter of the RBF kernel function.

d) Sigmoid kernel function (Sigmoid):

$$K(X_i, X_j) = \tanh[u(X_i \cdot X_j) - r] \tag{17}$$

where u and r are parameters of the sigmoid function.

The performance of these four kernel functions varies across different application scenarios. In this paper' s case where the number of features far exceeds the number of samples, the linear kernel function is typically selected.

3.1 Model Validation

To test model sensitivity, particularly prediction accuracy and misclassification conditions, this paper employs a validation method based on confusion matrices. The structure of a confusion matrix is as follows:

For an n-order confusion matrix, when $i = j$, t_{ij} represents the number of samples correctly classified into class C_i; when $i \neq j$, t_{ij} represents the number of samples that should belong to class C_i but are classified as class C_j.

Model validation requires analysis from two perspectives: prediction accuracy and stability in handling different predictions.

First, we examine model accuracy. When predicting a sample of size N, the ratio of correctly predicted samples to the total sample size is called accuracy, denoted as A_r, which is the ratio of the trace of the confusion matrix to the total number of samples:

$$A_r = \frac{\sum_{i=1}^{n} t_{ii}}{N} \tag{18}$$

Under the same environment, using a program to conduct multiple predictions on a selected sample yields a random accuracy each time. Under extensive experimentation, the distribution of accuracy is shown in [Figure 4: see original paper]. With a large number of training samples, the model achieves high accuracy. After 100 experiments, the model' s accuracy stabilizes at approximately 98.1%.

Second, we examine model stability. The fewer misclassified samples relative to the total predicted samples, the more stable the model. This paper introduces the concept of misclassification degree to characterize model stability.

Definition 2. Let the misclassification degree be denoted as E_r, then:

$$E_r = \frac{1}{N}\sum_{i=1}^{n}\left(\sum_{j=0,j\neq i}^{n} t_{ij} \times \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{t_{ij}}{PC_i}\right) \tag{19}$$

where PC_i represents the number of predictions in the i-th classification category, i.e., the column sum of the confusion matrix.

Regarding misclassification degree, evaluation is needed from both numerical and graphical perspectives. Numerically, when misclassification is extremely rare, the misclassification degree approaches the reciprocal of the predicted sample size. Simultaneously, if the feature vector is reasonable and the total predicted sample size is extremely large, the misclassification degree approaches 0, which aligns with practical scenarios.

As shown in [Figure 5: see original paper], when the test set gradually increases, the misclassification degree tends toward the reciprocal of the sample size. Meanwhile, as the sample size grows, the misclassification degree fluctuates along a horizontal line slightly above the reciprocal of the sample size. Therefore, the model performs optimally in large-scale data sample scenarios.

A model satisfying both high accuracy and low misclassification degree exhibits good stability. In summary, this paper refers to the joint examination of accuracy and misclassification degree as the (1-0) test model. When both satisfy their respective test conditions, ideal prediction results can be obtained. This model not only pursues high success rates but also considers applicability under complex conditions, demonstrating excellent performance in measuring classification quality.

3.2 Experiment Based on Community Management Platform Data

This paper further tests the model' s applicability using real data collected from Wuhu City' s "Quanmin Sheguan" (Community Participation in Governance) platform. "Quanmin Sheguan" enables citizens to report uncivilized phenomena, safety hazards, damaged public facilities, and other issues via mobile phones for government departments to handle, thereby achieving a social governance pattern of "co-construction, co-governance, and co-sharing." The model classifies each reported case to facilitate rapid processing. Through this software, 9,906 extremely short texts were collected, covering six categories: environmental sanitation, illegal advertising, construction waste, safety hazards, illegal occupation,

and public facilities. The texts were input into a trained classifier, with results shown in Table 1.

To test the model' s misclassification degree, experiments were conducted with 600, 1,000, and 2,000 data points from the test set without changing the training set, with results shown in [Figure 5: see original paper]. The figure demonstrates that as the test set increases, the misclassification degree tends toward the reciprocal of the sample size, while fluctuating along a horizontal line slightly above this value. Thus, the model performs well in large-scale data scenarios.

To validate the model' s effectiveness, comparative experiments were conducted using Python for data analysis. The data were randomly divided into 70% training and 30% test sets, and five experiments were performed, with results shown in Table 1. The evaluation metrics selected were precision, recall, and F1-score. Precision reflects the model's ability to distinguish positive samples, recall reflects its ability to distinguish negative samples, and F1-score is their harmonic mean. Experimental data indicate that SVM achieves higher accuracy than other algorithms and demonstrates superior performance in sample recognition.

The dataset includes six categories: environmental sanitation, illegal advertising, construction waste, safety hazards, illegal occupation, and public facilities. Inputting the texts into a trained classifier yields the correctly classified quantities shown in Table 2. Table 2 demonstrates that SVM' s classification effect is significantly better than the other three algorithms, particularly excelling when sample sizes are small.

In SVM applications, selecting an appropriate kernel function is crucial. Current selection schemes primarily include: (1) relying on prior experience, and (2) conducting comparative experiments. This paper determined the linear kernel function through comparative experiments, with results (classification accuracy) shown in Table 3. The comparison reveals that the linear kernel function offers significant advantages over other kernel functions, maintaining over 98% classification accuracy while other kernel functions achieve only approximately 45%.

4 Conclusion

Based on the practical needs of intelligent social management platforms, this paper proposes an extremely short text classification model founded on Bayesian decision theory. The model ensures keyword rationality through feature word extraction, combines classification probabilities with a Bayesian classifier, and undergoes rigorous testing. Experiments demonstrate that the model exhibits excellent performance in extremely short text classification. However, issues such as misclassification degree stability and word weight rationality require further investigation.

References

[1] Butnaru A M, Ionescu R T. From image to text classification: a novel approach based on clustering word embeddings[J]. Procedia Computer Science, 2017: 112-120.

[2] Sabbah T, Selamat A, Selamat M H, et al. Modified frequency-based term weighting schemes for text classification[J]. Applied Soft Computing, 2017, 58(9): 193-206.

[3] Liu Zhikang. An improved mixed kernel function support vector machine text classification method[J]. Industrial Control Computer, 2016, 29(6): 113-114.

[4] Li Cunhe, Ma Minmin. Optimization of kernel function of incremental support vector machine[J]. Computer Systems & Applications, 2017, 26(8): 284-287.

[5] Zheng Lizhou. Research on Several Techniques of Short Text Information Extraction[D]. Hefei: University of Science & Technology of China, 2016.

[6] Ren Di, Wan Jian, Yin Yuyu, et al. Research on Web service quality prediction method based on Bayesian classification[J]. Journal of Zhejiang University: Engineering Science, 2017, 51(6): 1242-1251.

[7] Chi Yunxian, Zhao Shuliang, Luo Yan, et al. Text data preprocessing method based on word frequency statistics[J]. Computer Science, 2017, 44(10): 276-282.

[8] Data Hall. Stop word collection[DB/OL]. http://www.datatang.com/data/19300/.

[9] Meng Dan. Research on image classification based on deep learning[D]. Shanghai: East China Normal University, 2017.

[10] Mao X, Zhao G, Sun R. Naive Bayesian algorithm classification model with local attribute weighted based on KNN[C]//Proc of IEEE Information Technology, Networking, Electronic and Automation Control Conference. IEEE, 2017: 904-908.

[11] Mirzaei A, Mohsenzadeh Y, Sheikhzadeh H. Variational relevant sample-feature machine: a fully Bayesian approach for embedded feature selection[J]. Neurocomputing, 2017, 241: 181-190.

[12] Yi Shunming, Yi Hao, Zhou Guodong. Study on Twitter sentiment classification method based on emotional feature vector[J]. Mini-Micro Systems, 2016, 37(11): 2454-2458.

[13] Yang Sichun, Dai Xinyu, Chen Jiajun. Research progress of problem classification technology for open domain question and answer[J]. Acta Electronica Sinica, 2015, 43(8): 1627-1636.

[14] Wei Fangfang, Duan Qingling, Xiao Xiaoyan, et al. Study on Chinese agricultural text classification technology based on support vector machine[J]. Trans-

actions of the Chinese Society of Agricultural Machinery, 2015, 46(S1): 174-179.

[15] Wu Jiajing, Wang Yang, Yan Xiaojing, et al. A social network stylistic classification method based on multi-agent theory[J]. Journal of Computer Systems, 2014, 23(11): 122-126.

Note: Figure translations are in progress. See original paper for figures.