Multi-type Classifier Fusion for Text Classification: A Postprint

Authors: Li Huifu, Lu Guang

Date: 2018-05-18T00:00:00+00:00

Abstract

Traditional text classification methods predominantly employ a single classifier. However, different classifiers exhibit varying emphases on classification tasks, which imposes certain limitations on single-classifier approaches, while each feature extraction method considers feature words from distinct perspectives. To address these issues, we propose a text classification method based on multi-type classifier fusion. This model employs word2vec, Principal Component Analysis, Latent Semantic Indexing, and TFIDF as feature extraction methods for multi-type classifier fusion. Furthermore, to remedy the problem of neglecting category information in multi-type classifier weighted voting methods, we propose a category-weighted classifier weight calculation method. Experimental results demonstrate that the multi-type classifier fusion method achieves excellent performance on binary, multi-class, and specific corpora, and the category-weighted classifier weight calculation method improves classification performance by 1.19% compared to the multi-type classifier fusion method.

Full Text

Preamble

Journal Information: ChinaXiv Cooperative Journal *Application Research of Computers*

Title: Research on Text Classification Method Based on Multi-Type Classifier Fusion

Authors: Li Huifu, Lu Guang† (College of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China)

Abstract: Most traditional text classification methods employ a single classifier, yet different classifiers emphasize different aspects of classification tasks, which introduces inherent limitations to any single-method approach. Furthermore,

each feature extraction method considers feature words from distinct perspectives. To address these issues, this paper proposes a text classification method based on multi-type classifier fusion. The model utilizes word2vec, Principal Component Analysis, Latent Semantic Indexing, and TFIDF as feature extraction methods for the fusion framework. To remedy the problem that multi-type classifier weighted voting methods ignore category information, we propose a category-weighted classifier weight calculation approach. Experimental results demonstrate that the multi-type classifier fusion method achieves excellent performance on binary, multi-class, and domain-specific corpora. The category-weighted classifier weight calculation method improves classification performance by 1.19% compared to the standard multi-type classifier fusion approach.

Keywords: text classification; classifier fusion; principal component analysis; latent semantic indexing

0 Introduction

The maturation of the Internet and the rise of social networks such as Weibo have revolutionized information technology and profoundly transformed people' s lifestyles. Increasingly, users publish and evaluate information online, with thematic categories spanning various harmful content including pornography, cult propaganda, and drug-related information [1]. Consequently, effective information management is crucial for the healthy development of the Internet.

Text classification in machine learning represents a key technology for processing and managing document data [2]. Researchers have conducted extensive studies on text classification methods. For instance, the K-Nearest Neighbors (KNN) method has been leveraged for its simplicity and non-parametric nature in spam SMS classification [3]. Goudjil et al. [4] utilized posterior probabilities provided by SVM classifiers for sample selection. In [5], feature weighting frequencies were deeply computed from training data to estimate Bayesian conditional probabilities, thereby improving classification performance. These studies have achieved excellent results, yet limitations remain: the K-value in KNN is manually set and highly subjective, determining kernel functions in high-dimensional space for SVM remains challenging, and Bayesian classification assumes feature independence despite real-world feature interdependencies. All these methods employ single classifiers, which cannot adequately cover the vast domains spanned by text data. Therefore, this paper introduces a multi-classifier fusion approach for text classification.

Text classification assigns documents with similar content to one or more predefined categories, where feature extraction methods play a vital role in improving classifier performance [6]. For example, [7] applied kinetic energy theory with TFIDF feature extraction for Weibo topic detection. [8] utilized word2vec as an automatic feature extraction tool, subsequently employing sentence vectors for

classification. Santosh et al. [9] leveraged a feature ontology tree and LDA for online product review analysis to better identify opinion words. Uysal et al. [10] combined genetic algorithms with LSI to obtain better document feature vectors for classification tasks.

While these feature extraction methods have proven effective in their respective tasks, each has limitations: TFIDF only considers statistical metrics without semantic knowledge; LSI focuses solely on semantic relationships between feature words; word2vec neglects statistical features; and LDA fails to incorporate category information into its topic model. As these are single feature extraction methods with different emphases, they exhibit certain constraints. To better represent text features, this paper employs a fusion of feature extraction methods. In summary, to address the limitations of single classifiers and single feature extraction methods, we propose a multi-type classifier fusion method for text classification. Additionally, to remedy the issue that classifier weighting in voting decisions overlooks category-specific contributions, we introduce a category-weighted classifier weight calculation method.

---

1 Multi-Type Classifier Fusion for Text Classification

The multi-type classifier fusion approach enriches feature words in the feature space vector by combining different feature extraction methods, creating richer text representations before applying classifiers for text categorization. Our method incorporates the following feature extraction techniques: word2vec, TF-IDF (Term Frequency-Inverse Document Frequency), Latent Dirichlet Allocation (LDA), and Latent Semantic Indexing (LSI).

1.1 Feature Extraction Methods

1.1.1 Word2vec In 2013, Mikolov et al. proposed the open-source word2vec toolkit [11]. Word2vec converts words into vector representations through neural network approaches. During training, it first extracts words from the training dataset to generate a vocabulary, then uses either CBOW (Continuous Bag-of-Words) or Skip-Gram models to obtain word vectors for each term. The model architecture is illustrated in [Figure 1: see original paper].

As shown in [Figure 1: see original paper], CBOW and Skip-Gram represent inverse processes. CBOW predicts the current word using its surrounding t words before and after, while Skip-Gram uses the current word to predict surrounding words. Since this work employs the CBOW model, we elaborate on its details.

CBOW is a model that utilizes contextual information to predict the current word' s occurrence probability. This three-layer neural network consists of an input layer, hidden layer, and output layer. The input layer receives word vectors (initialized randomly and updated during training), the hidden layer accumulates word vectors, and the output layer produces word probabilities.

1.1.2 TF-IDF TF-IDF is a classical feature weighting method composed of TF (Term Frequency) and IDF (Inverse Document Frequency), calculated as:

$$\text{tfidf}(w) = \text{tf}(w) \times \text{idf}(w)$$

where tf(w) represents the frequency of word w in the text, and idf(w) represents the inverse document frequency of word w, computed as:

$$\text{idf}(w) = \log \frac{A}{B(w)}$$

where A denotes the total number of documents in the training set, and B(w) represents the number of documents containing word w.

1.1.3 LDA LDA is a topic model representing a three-layer Bayesian structure of words-documents-topics. The model trains on the dataset to obtain Dirichlet distributions over topics and multinomial distributions between topics and words. The process first determines a topic, then selects words from that topic until all words are traversed. The LDA model is shown in [Figure 2: see original paper].

In [Figure 2: see original paper], W represents words in the text; N represents the number of words; M represents the number of documents; L represents the multinomial distribution with parameter D; P2 represents the Dirichlet prior parameter indicating word probability; D represents the topic distribution with Dirichlet parameter P1; and P1 is the parameter for D.

1.1.4 LSI LSI is an unsupervised data mining technique effective for semantic issues such as polysemy. LSI employs Singular Value Decomposition (SVD) to decompose the feature vector space for dimensionality reduction. The algorithm model is illustrated in [Figure 3: see original paper].

1.2 Multi-Type Classifier Fusion

Section 1.1' s feature extraction methods produce four distinct feature vector spaces: (1) word2vec vector space from CBOW, (2) TFIDF vector space, (3) LSI semantic vector space, and (4) LDA vector space from topic modeling. Multi-type classifier fusion leverages the complementarity among these vector spaces. The fusion model is shown in [Figure 4: see original paper], where numbers in triangles represent classifier weights.

1.3 Category-Weighted Multi-Type Classifier Fusion

Multi-classifier fusion can utilize different classifiers for different tasks, avoiding narrow consideration. Since different classifiers exhibit varying capabilities on the same sample, each classifier contributes differently to each sample. Classifier

weighted voting serves as one decision method for ensemble classification [12]. Using classification performance (the correct recognition rate of trained samples on training data) as classifier weights offers advantages: different classifiers have different recognition rates for the same sample, and when the ensemble makes classification decisions, results tend toward better-performing classifiers, optimizing decision performance. Therefore, we use classification performance as classifier weights, calculated as:

$$\varepsilon = \frac{\text{errorNum}}{\text{textNum}}$$

where errorNum is the number of misclassified samples, textNum is the total number of samples in the dataset, and represents the classifier weight for the dataset, computed as:

$$\alpha = \ln\left(\frac{1-\varepsilon}{\varepsilon}\right)$$

[Figure 5: see original paper] illustrates a traditional binary classification sample scenario with 100 data points: 40 training points per class and 10 test points per class, where represents training data for class 1, × represents training data for class 2, represents test data for class 2, and represents test data for class 1. Using KNN and Multinomial Naive Bayes as classification algorithms with classification performance as weights, KNN achieves = 0.0375 and = 3.2452, while Multinomial Naive Bayes achieves = 0.9625 and = 2.5123. Thus, KNN' s classifier weight is 3.2452 and Multinomial Naive Bayes' s weight is 2.5123. KNN misclassifies 12 samples, while Multinomial Naive Bayes misclassifies 10. When test samples enter the ensemble, voting principles yield 12 misclassified test samples—equivalent to KNN' s error rate. This approach uses overall classification performance as classifier weights, ignoring category influences.

Therefore, we propose a category-weighted classifier weight calculation method. Category-weighted classifiers consider how category information affects classifier weights. In the ensemble described above, KNN' s classification performance includes 1 negative sample and 2 positive samples correctly classified, while Multinomial Naive Bayes includes 6 negative samples. This analysis reveals that Multinomial Naive Bayes has good recognition for positive samples, suggesting we should increase its weight for positive samples and decrease it for negative samples. Thus, we assign different classifier weights based on sample categories. The category-weighted classifier formula is:

$$\varepsilon_l = \frac{\text{error}_l\text{Num}}{\text{textNum}}$$

where x_i represents test samples and L_l represents classifier weights under category l. error_lNum is the classification error rate for category l, and is computed as in Equation (4).

From the data analysis, category-weighted classifier weighting better represents classifier contributions. Integrating this into the multi-type classifier fusion model yields an improved architecture shown in [Figure 6: see original paper].

1.4 Multi-Type Classifier Algorithm Steps

Input: Training sample set x_train, test sample set x_test, training labels y_train, test labels y_test, number of classifiers classNum

Output: Prediction result matrix predicted

1. Calculate word2vec for x_train as class1
2. Calculate TFIDF for x_train as class2
3. Calculate LSI for x_train as class3
4. Calculate LDA for x_train as class4
5. Train classifier according to x_train
6. For i {1, 2, ···, len(x_train)}:
   - Calculate error_word2vecNum, error_tfidfNum, error_lsiNum, error_ldaNum for each class according to y_train
7. For i {1, 2, ···, len(x_test)}:
   - Calculate w_li for error_word2vecNum, error_tfidfNum, error_lsiNum, error_ldaNum according to equations (4), (5), and (6)
8. For i {1, 2, ···, len(x_test)}:
   - For j {1, 2, ···, classNum}:
     - s[j] += class[j] * w[j][i]
   - predicted[i] = index of maximum value in s as class
9. Return predicted

---

2 Experiments

2.1 Experimental Data

To validate the performance of our multi-type classifier fusion method, we conducted experiments using the movie_reviews corpus from NLTK [13], the Sogou corpus for general text classification, and the 20news corpus [14]. The Sogou and 20news corpora are standard benchmarks for evaluating algorithm performance, with 20news being a relatively balanced dataset. The movie_reviews corpus is a sentiment analysis dataset for film reviews, enabling better validation of our algorithm through domain-specific classification tasks. Dataset distributions are shown in .

TABLE:1 Dataset Distribution

Dataset Name	Category Name	Training Set	Test Set
movie_reviews	-	-	-

Dataset Name	Category Name	Training Set	Test Set
20news	atheism, crypt, graphics	-	-
Sogou Corpus	-	-	-

2.2 Experimental Analysis

The experimental platform was Anaconda with Python, conducted on a computer with 4GB RAM and 1TB hard drive. Classification employed KNeighborsClassifier from Python' s sklearn library (k=10). Feature extraction methods included word2vec, LSI, LDA, and TFIDF, filtering out features appearing fewer than 30 times. Sample recognition rate served as the evaluation metric with 6-fold cross-validation [15]. We performed three experiments: (1) validating algorithm effectiveness, (2) examining feature dimension impact, and (3) verifying the category-weighted classifier weighting method.

2.2.1 Comparison of Multi-Type Classifier Fusion Methods This experiment used the 20news and movie_reviews datasets. 20news is a relatively balanced multi-class dataset, while movie_reviews is a balanced binary classification dataset for sentiment analysis. Feature dimension was set to 300. Results are shown in .

TABLE:2 Classification Results on 20news and movie_reviews (%)

Method	movie_reviews	20news
TFIDF	-	-
word2vec	-	-
LDA+TFIDF	-	-
LSI+TFIDF	-	-
LSI+LDA	-	-
word2vec+TFIDF	-	-
word2vec+LDA	-	-
word2vec+LSI	-	-
LSI+LDA+TFIDF	-	-
word2vec+TFIDF+LDA	-	-
word2vec+LSI+TFIDF	-	-
word2vec+LSI+LDA	-	-
word2vec+LSI+TFIDF+LDA	-	-
Our Algorithm (Min Recognition Rate)	-	-

The results show that 20news achieved an average recognition rate of 92.16%, higher than movie_reviews' 67.74%, because movie_reviews is a specialized sentiment analysis dataset while 20news is for general text classification. Our multi-type classifier fusion and its variants achieved strong performance, improving movie_reviews by at least 2.13% and 20news by at least 1.06%. Since

20news is relatively balanced, classifier recognition rates tend toward majority categories.

2.2.2 Impact of Feature Dimensions on Fusion Classifiers This section uses movie_reviews to evaluate performance across different dimensions (100, 300, 500, and 700). Results are shown in .

TABLE:3 movie_reviews with Different Feature Dimensions

Method	Dim 100	Dim 300	Dim 500	Dim 700
TFIDF	-	-	-	-
word2vec	-	-	-	-
LDA+TFIDF	-	-	-	-
LSI+TFIDF	-	-	-	-
LSI+LDA	-	-	-	-
word2vec+TFIDF	-	-	-	-
word2vec+LDA	-	-	-	-
word2vec+LSI	-	-	-	-
LSI+LDA+TFIDF	-	-	-	-
word2vec+TFIDF+LDA	-	-	-	-
word2vec+LSI+TFIDF	-	-	-	-
word2vec+LSI+LDA	-	-	-	-
word2vec+LSI+TFIDF+LDA	72.15	73.00	70.60	70.75
Our Algorithm (Min Recognition Rate)	-	-	-	-

The results indicate that performance peaks at 300 dimensions for movie_reviews. As dimensions increase, average recognition rates decline because higher dimensionality introduces sparsity (more zero values) in document feature vectors, degrading classifier effectiveness.

2.2.3 Comparison of Category-Weighted Classifier Weights This experiment uses the Sogou corpus to validate category-weighted classifier effectiveness. Results are shown in .

TABLE:4 Results on Sogou Corpus

Method	Performance-Weighted	Category-Weighted
TFIDF	-	-
word2vec	-	-
LDA+TFIDF	-	-
LSI+TFIDF	-	-
LSI+LDA	-	-
word2vec+TFIDF	-	-
word2vec+LDA	-	-

Method	Performance-Weighted	Category-Weighted
word2vec+LSI	-	-
LSI+LDA+TFIDF	-	-
word2vec+TFIDF+LDA	-	-
word2vec+LSI+TFIDF	-	-
word2vec+LSI+LDA	-	-
word2vec+LSI+TFIDF+LDA	-	-
Our Algorithm (Min Recognition Rate)	-	-

Category-weighted methods demonstrate consistent improvements over performance-weighted approaches, with average accuracy increasing by 1.19%. The fusion method outperforms sub-methods by 0.82%, and all single feature extraction methods show improved performance except word2vec+TFIDF, word2vec+LSI, and LSI+LDA+TFIDF. This occurs because the corpus size limits word2vec' s effectiveness, and LDA, LSI, and word2vec achieve similar recognition rates, making fusion effects comparable.

3 Conclusion

This paper addresses the limited extensibility of single classifiers and single feature extraction methods by proposing a multi-type classifier fusion approach for text classification. We combine four different feature extraction methods to create a multi-type text classification framework. To remedy the issue of classifier weights ignoring category information, we introduce a category-weighted classifier weight calculation method. Experiments on binary and multi-class classification tasks validate our algorithm' s effectiveness. Future work will focus on parallelizing the method to reduce computational time.

References

[1] He L, Ding Z, Jia Y, et al. Candidate category search in large-scale hierarchical classification [J]. Chinese Journal of Computers, 2014, 37(1): 41-49.

[2] Li R, Wang J, Chen X, et al. Chinese text categorization using maximum entropy model [J]. Journal of Computer Research and Development, 2005, 42(1): 94-101.

[3] Huang W, Mo Y. Chinese spam filtering based on text-weighted KNN algorithm [J]. Computer Engineering, 2017, 43(3): 193-199.

[4] Goudjil M, Koudil M, Bedda M, et al. A novel active learning method using SVM for text classification [J]. International Journal of Automation and Computing, 2016: 1-9.

[5] Jiang L, Li C, Wang S, et al. Deep feature weighting for naive Bayes and its application to text classification [J]. Engineering Applications of Artificial Intelligence, 2016, 52(C): 26-39.

[6] Sangodiah A, Ahmad R, Wan F W A. A review in feature extraction approach in question classification using Support Vector Machine [C]// Proc of IEEE International Conference on Control System, Computing and Engineering.

[7] Chen S, Jin Z. Weibo topic detection based on improved TF-IDF algorithm [J]. Science and Technology Review, 2016, 34(2): 282-286.

[8] Wang Z, Ma L, Zhang Y. A hybrid document feature extraction method using latent dirichlet allocation and word2vec [C]// Proc of IEEE International Conference on Data Science in Cyberspace. 2016.

[9] Santosh D T, Babu K S, Prasad S D V, et al. Opinion mining of online product reviews from traditional lda topic clusters using feature ontology tree and Sentiwordnet [C]// Proc of International Conference on Social Computing and Social Media. 2016: 34-44.

[10] Uysal A K, Gunal S. Text classification using genetic algorithm oriented latent semantic features [J]. Expert Systems with Applications, 2014, 41(13): 5932-5937.

[11] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [C]// Proc of International Conference on Learning Representations. 2013.

[12] Wang X, Li R, Xue A, et al. Adaptive weighted voting HRRP fusion recognition method based on entropy [J]. Systems Engineering and Electronics, 2017, 39(4): 707-713.

[13] Jongeling R, Sarkar P, Datta S, et al. On negative results when using sentiment analysis tools for software engineering research [J]. Empirical Software Engineering, 2017, 22(5): 2543-2584.

[14] Wei Y, Hu D, Hao C, et al. Geopolitical thematic crawler design based on classified keyword frequency model [J]. Computer Engineering, 2016, 42(2): 45-50.

[15] Duan H, Zhang Q, Zhang M. FCBF feature selection algorithm based on normalized mutual information [J]. Journal of Huazhong University of Science and Technology: Natural Science Edition, 2017, 45(1): 52-56.

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