Computing and Informatics, Vol. 37, 2018, 992–1010, doi: 10.4149/cai_2018_4_992

A NOVEL KERNEL FOR TEXT CLASSIFICATION BASED ON SEMANTIC AND STATISTICAL INFORMATION

Haipeng Yao, Bo Zhang, Peiying Zhang

School of Information and Communication Engineering
Beijing University of Posts and Telecommunications
No. 10 Xitucheng Road
Haidian District, Beijing, China
e-mail: yaohaipeng@bupt.edu.cn, zhangbobupt@qq.com,
zhangpeiying@upc.edu.cn

Maozhen Li

Department of Electronic and Computer Engineering
Brunel University London
Uxbridge, UB8 3PH, UK
e-mail: Maozhen.Li@brunel.ac.uk

Abstract. In text categorization, a document is usually represented by a vector space model which can accomplish the classification task, but the model cannot deal with Chinese synonyms and polysemy phenomenon. This paper presents a novel approach which takes into account both the semantic and statistical information to improve the accuracy of text classification. The proposed approach computes semantic information based on HowNet and statistical information based on a kernel function with class-based weighting. According to our experimental results, the proposed approach could achieve state-of-the-art or competitive results as compared with traditional approaches such as the k-Nearest Neighbor (KNN), the Naive Bayes and deep learning models like convolutional networks.

Keywords: Text categorization, semantic information, statistical information, support vector machine

Mathematics Subject Classification 2010: 68T50

1 INTRODUCTION

In recent years, with the increasing volume of text information on the Internet and social media, text categorization has become a key technique to process these textual data. In text categorization, a Bag of Words (BOW) is usually used to represent a document. The weight in BOW is usually obtained by computing word frequency or the widely accepted TF-IDF formula. However, the BOW representation has several limitations:

1. The TF-IDF formula does not consider the calculation of class-based weights. As a result, the same word has the same weight in all categories.
2. It cannot deal with synonyms and polysemy.

In the absence of knowledge-based word similarity and statistic-based word similarity, automatic text categorization using BOW only as a document representation model [1] has not yet achieved the best performance and cannot meet the needs of all scenes in the real life. There are two ways to address this problem. First, we could use language model based on deep learning models such as word2vec [2] and Glove [3] to learn the vector representations of words. However, such new approaches do not have to be necessarily better when the corpora is not particularly large. And it takes considerable time and effort to train word vectors. The second method is to collect semantic and syntactic informations as much as possible. We mainly adopt the second method and develop a new semantic smoothing kernel function based on knowledge-based word similarity and statistic-based word similarity to increase the capability of feature vectors to represent a document.

This paper presents a novel approach for text classification. In this approach, word similarity based on HowNet is embedded into semantic information. This method promisingly improves the accuracy of text classification via using ontology knowledges. Moreover, the proposed approach takes advantage of the class-based term weighting by giving more weights on core words in each class during the transformation phase of SVM from the input space to the feature space. A term has a more discriminative power on a class if it has a higher weight for that class. The heuristic idea combining sematic and statistical information finally improves the classification accuracy.

The rest of the paper is organized as follows: In Section 2, we briefly introduce the Support Vector Machine (SVM) and discuss the related work in the field of semantic smoothing kernels, with an emphasis on the task of text classification. Section 3 describes and analyzes the proposed kernel for text classification. Experimental setup and corresponding experiment results are given in Section 4. Finally, we conclude this paper in Section 5 with a discussion on future work.

2 RELATED WORK

2.1 Support Vector Machines for Classification

Support vector machine (SVM) is a very effective machine learning algorithm developed from statistical learning theory. This algorithm was proposed by Vapnik, Guyon and Boser [4] and further analyzed in [5]. The core goal of SVM is to find the optimal segmentation hyperplane by the maximum spacing between classes. The author of [6] proposed that SVM has many advantages, such as finding the global optimal solution and having a good robustness.

SVM kernel function can be regarded as similarity function, because it calculates the similarity values of data sets. It is proposed to define a suitable kernel function [7], which has a direct influence on finding the optimal hyperplane. The commonly used kernel functions for the document vectors are given below:

$$LinearKernel : k(d_p, d_q) = d_p d_q, \tag{1}$$

$$PolynomialKernel : k(d_p, d_q) = (d_p d_q)^b, b = 1, 2, \ldots, \tag{2}$$

$$RBFKernel : (d_p, d_q) = \exp(\gamma||d_p - d_q||)^2. \tag{3}$$

In current works, the authors of [8] proposed to develop a kernel function based on the similarity of the knowledge system, and used the Omiotis library function to measure the similarity of English words and improve the accuracy of the classification. The authors of [9] proposed to develop a kernel function based on the weights of the class which improved the accuracy of the classifier. Based on these research efforts, this paper optimizes the similarity of Chinese text words and combines the statistical methods with the knowledge-based methods to construct a kernel functions to improve the accuracy of text classification.

2.2 Knowledge-Based Word Similarity

Knowledge-based systems use ontology or thesaurus to capture the concepts in the documents and incorporate the domain knowledge into the words for the representation of textual data. These systems enhance the representation of terms by taking advantages of semantic relatedness among terms.

The similarity calculation of words is added to the text classification [10], which is used to modify the weights of the text feature vectors. Mavroeidis et al. [11] proposed a semantic kernel function based on WordNet [12] to improve the accuracy of English text classification. Based on the Chinese semantic knowledge system of HowNet [13], Zhang embeded the semantic similarity into the kernel function of Chinese text classification [14], and improved the performance of Chinese text classification.

In this paper, we use the Chinese dictionary HowNet to calculate the semantic similarity of words. HowNet is a very detailed dictionary of semantic knowledge.

Unlike CiLin [15] and WordNet, every word in HowNet has multidimensional knowledge representations. The structure of HowNet is described in detail below.

HowNet mainly includes concepts and primitives. Each term is described by a number of concepts, each of which is described by a sequence of primitives, so primitive is the smallest expression unit in HowNet. HowNet contains 1 500 primitives, which can be divided into three categories: basic semantics (describing the semantic features of concepts), grammatical semantics (describing the grammatical features of words) and relational semantics (describing the relationship between concepts). When we calculate the word similarity, we can define it in the following way. Word similarity calculation consists of four parts in Equation (4).

$$Sim(S_1, S_2) = \sum_{i=1}^{4} \beta_i \prod_{j=1}^{i} Sim_j(S_1, S_2). \tag{4}$$

$Sim_1(S_1, S_2)$ is the similarity of first basic primitives of these two words. The similarity $Sim_1(S_1, S_2)$ between S_1 and S_2 can be calculated using Equation (5). $Sim_2(S_1, S_2)$ is the similarity of the rest basic primitives, that is the arithmetic mean of the similarity of all pairs of elements. $Sim_3(S_1, S_2)$ is the similarity of two grammatical semantics, which can be transformed into the basic semantic meaning in the grammatical semantics. $Sim_4(S_1, S_2)$ is the similarity of two relational semantics, but the elements in the relational semantics are sets, which are basic primitives or concrete words.

There is a close relationship between word similarity and word distance. In fact, word similarity and word distance are different forms of the same feature of a pair of words. Word similarity is defined as a real number between 0 and 1.

$$Sim_1(S_1, S_2) = \frac{\alpha}{d + \alpha} \tag{5}$$

where S_1 and S_2 represent two of the words respectively, d is the distance between S_1 and S_2 in the original path hierarchy in HowNet; α is an adjustable parameter. When the distance in HowNet between words is particularly large, $Sim(S_1, S_2)$ approaches 0; when the distance in HowNet between words is particularly small, $Sim(S_1, S_2)$ approaches 1.

β_i in Equation (4) is an adjustable parameter and satisfies the Equation (6). The latter part of the Equation (6) represents the descending importance of $Sim_1(S_1, S_2)$ to $Sim_4(S_1, S_2)$.

$$\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1, \quad \beta_1 \geq \beta_2 \geq \beta_3 \geq \beta_4. \tag{6}$$

2.3 Statistic-Based Word Similarity

In the absence of semantic knowledge, a statistical-based approach such as the one presented in [16] can be applied to the text categorization to solve synonymic problems. Statistical similarity calculation is based on the correlation of training words, so the statistical similarity calculation method is very sensitive to the training data sets.

The similarity calculation using statistics in text categorization contains calculation based on classes [9], high-order paths [17, 18, 19] and mean value calculations [20]. In this paper, we use the method proposed in [9] which considers the weight of a certain word in a training set depending on the relevance of terms and categories. After the text feature vector is smoothed through the semantic kernel, it can improve the weights of important words in the category and reduce the weight of common words in the category. By modifying weights of the text feature vectors, the representation capability of the feature vectors is increased.

2.4 Weight of Feature Words

In the classification system, the most commonly used method of calculating word weights is the TF-IDF formula mentioned in [21, 22] where TF denotes the term frequency and IDF denotes inverse document frequency. TF-IDF formula was first used in the field of information retrieval, because its calculation method is simple and practical. It is also widely used in the text automatic classification.

TF-IDF is a statistical method to evaluate the importance of a document in a corpus. In general, the importance of a word increases in proportion to its number of occurrences in the document and decreases inversely with its higher frequency of occurrences in the corpus.

IDF formula is given in Equation (7)

$$IDF(w) = \frac{|D|}{df_w} \tag{7}$$

where $|D|$ denotes the total number of documents in the corpus and df_w represents the number of documents which contains term w.

TF-IDF formula is given in Equation (8)

$$TFIDF(w, d_i) = tf_w * \log(IDF(w)) \tag{8}$$

where tf_w represents the term frequency which is the number of word w in document D.

TF-ICF is proposed as another method of calculating word weight in [23, 24], which is similar to TF-IDF. ICF denotes inverse class frequency. TF-ICF calculates the word weight in category level rather than in document level.

Equation (9) shows the ICF formula:

$$ICF(w) = \frac{|C|}{cf_w} \tag{9}$$

where $|C|$ denotes the total number of classes in the corpus and cf_w represents the number of classes which contain term w.

TF-ICF formula is shown in Equation (10),

$$TFICF(w, c_i) = \sum_{d \in c_j} tf_w * \log(ICF(w)). \tag{10}$$

Inspired by the IDF and ICF formulas, [25, 26] propose a new method for calculating weights:

$$W_{w,c} = \log(tfc_{w,c} + 1) * \frac{|D|}{df_w} \tag{11}$$

where $tfc_{w,c}$ represents the total number of feature term w of class c. $|D|$ denotes the total number of documents and df_w represents the number of documents which contain term w.

From the analysis above, we can see that W is a matrix which is determined by categories and feature terms. In fact, terms that are similar to the topic in the category are given a larger weight because of the W matrix. The authors of [25, 26] compare the weighting algorithm based on the category with other commonly used feature selection algorithms, and conclude that the former one can improve the classification performance significantly.

3 MERGED KERNEL FUNCTION

Both knowledge-based word similarity computation and statistical-based word similarity computation can improve the performance of text classifiers. A heuristic idea is to apply two computation methods to a kernel function which we call merged kernel function in this paper. The pseudocode for the merged kernel function is shown in Algorithm 1. This section will explain in detail how to combine the two kernel functions to improve the accuracy of classification.

3.1 Vector Space Model

Document representation is a basic problem in natural language processing. Computer cannot directly deal with document which is mainly consisted of unstructured data. The key challenge is how to map a document into a vector space model (VSM). First, a document d_i is represented as an n-dimensional vector composed of feature words as shown in Equation (12).

$$d_j = (w_{1j}, w_{2j}, \ldots, w_{nj}). \tag{12}$$

Then the weighting formula maps the document vector d_i to the word weight vector $\phi(d_j)$:

$$\phi(d_j) = [\mathit{tfidf}(t_1, d_j), \mathit{tfidf}(t_2, d_j), \ldots, \mathit{tfidf}(t_n, d_j)] \tag{13}$$

where $\mathit{tfidf}(t_i, d_j)$ denotes the TF-IDF value of the feature word t_i in the document d_j.

```
Algorithm 1 Calculation of the combined semantic smoothing matrix C
Require: Training set D
Ensure: Semantic Smoothing Matrix C
Local variables :
        $tfc_{w,k}$ : total term frequency of word $w$ in the documents of class k
        $tf_{w,d}$ : total term frequency of word $w$ in the document d
        $N$ : total number of documents in the training set
        $N_w$ : shows total number of documents in the training set those contain
    word w
        $Z$ : matrix is constructed according to the list of feature word by the
    introduction of HowNet
 1: for each word w do
 2:     for each document d contains word $w$ in the train set do
 3:         $N_w = N_w + 1$
 4:     end for
 5: end for
 6:
 7: for each word w do
 8:     for each document d contains word $w$ in class k do
 9:         $tfc_{w,k} = tfc_{w,k} + tf_{w,d}$
10:     end for
11:     $W_{w,k} = (log(tfc_{w,k}) + 1) * log(N/N_w)$
12: end for
13:
14: $S = W * W^T$
15:
16: $Z^2 = Z * Z^T$
17:
18: for  i in columns of S do
19:     for for j in rows of S do
20:         $C_{i,j} = \lambda_1 * S_{ij} + \lambda_2 * Z^2_{ij}$
21:     end for
22: end for
```

3.2 Statistic-Based Similarity Matrix

The training phase for statistic-based similarity matrix is shown in Algorithm 1 from the first line to the fourteenth line. In order to embed the statistical information into the space vector model and increase the capability of representing a document of feature vectors, we construct a matrix S based on the class-based weight, which is called the statistical similarity matrix. The formula has been described in detail in Section 2.4. To make use of this formula, we define the statistical similarity matrix S as:

$$S = WW^T \tag{14}$$

where W is the class-based weighting formula mentioned in Section 2.4. The statistical similarity matrix S is a symmetric matrix, and the element S_{ij} in matrix S is the class-based statistical similarity of the feature words w_i and w_j.

The S matrix represents the statistical similarity of words. For example, the words "patient" and "sufferer" have similar meaning. When the weight of the "patient" is higher and the weight of the "sufferer" is lower in vector $\varphi(d_j)$, the weight of the word sufferer will be increased after multiplication with the S matrix.

3.3 Combined Semantic Smoothing Matrices

The knowledge-based similarity matrix Z can be constructed according to the list of feature words by the introduction of HowNet in Section 2.2. Z_{ij} denotes the knowledge-based similarity between the feature words w_i and w_j. The knowledge-based word similarity can be embedded into the space vector model by matrix Z. In order to ensure the validity of final merged kernel, a second-order rule similarity matrix is used here, that is $Z^2 = ZZ^T$.

After computing the statistic-based similarity matrix S and the second-order knowledge-based similarity matrix Z^2, the degree of weight modification of the two matrices to the space vector model cannot be determined, which should be determined according to the data set. In this paper, the matrix S and Z^2 are combined to generate a new semantic smoothing matrix C:

$$C_{ij} = \lambda_1 * S_{ij} + \lambda_2 * Z_{ij}^2 \tag{15}$$

where S_{ij} and Z_{ij}^2 are described in Sections 3.2 and 3.3, λ_1 and λ_2 adjust the normalization parameters of the weights in S and Z^2. Parameters λ_1 and λ_2 satisfy $\lambda_1 + \lambda_2 = 1$. We can adjust λ_1 and λ_2 to determine how matrices S and Z^2 affect the classification performance of the classifier.

3.4 Semantic Smoothing Kernel Function

By defining the mapping architecture matrix C, we can map a document vector to a new feature space vector by using Equation (16).

$$\overline{\phi}(d_j) = \phi(d_j)C. \tag{16}$$

The mapped vectors can be directly used in many classification methods. If the high-dimensional sparse matrix appears in the text classification, there will be a high-dimensional disaster in computation. Defining a kernel function can reduce the influence of the high-dimensional sparse matrix. The inner product between documents p and q in the feature space is computed by the kernel function using Equation (17).

$$K_{CK}(d_p, d_q) =< \overline{\phi}(d_p), \overline{\phi}(d_q) >= \phi(d_p)CC^T\phi(d_q)^T \tag{17}$$

where $K_{CK}(d_p, d_q)$ denotes the similarity of document d_p and document d_q. $\overline{\phi}(d_p)$ and $\overline{\phi}(d_q)$ are the new feature space vectors of document d_p and document d_q after transformed by the semantic smoothing matrix proposed in Equation (16). The kernel function information is stored in the matrix G:

$$G_{p,q} = K_{CK}(d_p, d_q). \tag{18}$$

Then we prove the validity of the semantic smooth kernel proposed in this section. According to Mercer's theorem [27], any semi-definite function can be used as a kernel function. The semantic smoothing matrix C proposed in this paper is composed of the statistical similarity matrix S and the second-order knowledge-based similarity matrix Z^2, so the matrix C is also a symmetric matrix. The matrix S and the matrix Z^2 are the product of a matrix and its transpositions, so the matrix S and Z^2 are both semi-definite matrix, which is proved in [28]. In linear algebra, the sum of two positive semi-definite matrices is also a semi-definite matrix. Therefore, the matrix C is a semi-definite matrix, satisfying the conditions required by Mercer's theorem. The kernel function can be constructed by the semantic smoothing matrix C.

4 EXPERIMENTS AND RESULTS

4.1 Corpora

This paper selects the corpus provided by the Sogou Company[1] and the Fudan University[2]. The Sogou corpus consists of SogouCA and SogouCS news corpora containing various categories of 2 909 551 news articles, of which about 2 644 110 articles contain both a title and relevant content. We manually categorize articles by using the channel in URL, then we get a large Chinese corpus with the article contents and categories. However, there are some categories that contain few articles. So 5 categories with largest number – "sports", "finance", "entertainment", "automobile" and "technology" are finally selected for our text classification experiments. The details of the Sogou corpus are presented in Table 1. During the training process, many parameters in the model are involved. In order to determine the optimal values of parameters in this model, we use the validation set. We partition training set, validation set and test set in 8:1:1 proportions in Sogou corpora after we shuffled the corpora. So the corpora is randomly divided into training set, validation set and test set. These parameters are described in detail in Section 4.4.

To validate the combine kernel's effect on a small corpora, we also use the corpus provided by the Fudan University. The corpora contains 9 804 articles that have been already divided into 20 categories. We choose 5 categories – "economy", "sports", "environment", "politics" and "agriculture". The details of the Fudan training set

[1] `http://www.sogou.com/labs/resource/list_news.php`

[2] `http://www.nlpir.org/download/tc-corpus-answer.rar`

Category	Total	Train	Validation	Test
Sports	645 931	80 000	10 000	10 000
Finance	315 551	80 000	10 000	10 000
Entertainment	160 409	80 000	10 000	10 000
Automobile	167 647	80 000	10 000	10 000
Technology	188 111	80 000	10 000	10 000

Table 1. Sogou News corpora

are presented in Table 2. We partition training set, validation set and test set in 7:1:1 proportions in Fudan corpus after we shuffled the corpora.

Category	Total	Train	Validation	Test
Economy	1 589	700	100	100
Sports	1 188	700	100	100
Environment	1 022	700	100	100
Politics	1 013	700	100	100
Agriculture	992	700	100	100

Table 2. Fudan University corpora

4.2 Word Segmentation and Stop Words

The current English word segmentation tool has been well developed, while the Chinese word segmentation technology is still evolving. For Python language, NLTK [29] nltk.tokenize module can be used for word segmentation in English, and jieba tool can be used for word segmentation in Chinese.

In order to save storage and improve the efficiency of classification, the classification system will ignore certain words after word classification, which are called stop words[3]. There are two kinds of stop words: the first one can be found everywhere in all kinds of documents with which the classification system cannot guarantee the true classification result. The second kind of stop words includes the modal particle, adverb, preposition, conjunction and so on.

4.3 Feature Word Extraction

In the problem of text categorization, a certain feature word and its class obey the CHI square distribution. The larger the CHI value is, the more the CHI value can be used to identify the category. The CHI formula is given:

$$\chi^2(t,c) = \frac{N(AD-BC)^2}{(A+C)(A+B)(B+D)(C+D)} \tag{19}$$

[3] `https://github.com/Irvinglove/Chinese_stop_words/blob/master/stopwords.txt`

where N is the number of texts in the training set, A is the number of documents belonging to class c and containing the word w; B is the number of documents that do not belong to class c but contain word w; C is the number of documents belonging to c class, but not containing the word w; D is the number of documents that do not belong to class c and do not contain the word w.

4.4 Experiment Settings

The classifier uses the SVM function provided in the machine learning library sklearn [30] in Python environment. We change the kernel function by using the interface it provides. We observe how the statistical similarity matrix S and the second-order knowledge-based similarity matrix Z^2 affect the performance of the classifier when the ratio of training set and parameter λ are different.

In the experiment, we set parameter values based on the experience gained from the validation set. First, we describe several parameters mentioned in Section 2.2. We compute word similarity using α 1.6, β_1 0.5, β_2 0.2, β_3 0.17 and β_4 0.13. Second, the length of VSM used for representing Sogou corpus is 10 000, and Fudan corpus is 1 000. Sogou corpus is large, so the feature vectors need to be longer. Finally, we set some parameters of the model. Penalty parameter C of the error term is 1.0 and there is no hard limit on iterations within solver, so that the SVM algorithm will stop training before over-fitting.

A number of parameters have been used to assess the performance of classification model output, such as accuracy [31] and f-measure (F1) [32]. To demonstrate that the combined kernel does improve the accuracy and F1 value of text classification, we use other machine learning methods for comparison, including KNN, Naive Bayes, and SVM with linear kernel and RBF kernel. The corpora is processed in the same way in Section 4.2 and Section 4.3, and then we call the interface of different machine learning algorithms in sklearn. We compare the results with character-level convolutional [33] networks which is the state-of-the-art method in text classification. Finally we adopt the accuracy and the F1 value of text classification as the evaluation standard.

4.5 Experiments and Results

As shown in Tables 3 and 4, the first column in the table shows the compared training algorithms, and the first row indicates the value of λ_1. The value of λ_2 corresponding to this is $1 - \lambda_1$. The second row shows the performance of the combined kernel in the case of different λ_1 value. The rest rows represent the performance of other machine learning methods. The values in Table 3 represent the accuracy rate of Sogou corpus and the values in Table 4 represent the values of F1 of Sogou corpus.

The values obtained in Tables 3 and 4 are shown by the line chart in Figures 1 and 2, respectively, from which it is easier to see the effect of the combination of the statistical similarity matrix S and the second-order knowledge-based similarity matrix Z^2 on the classification accuracy.

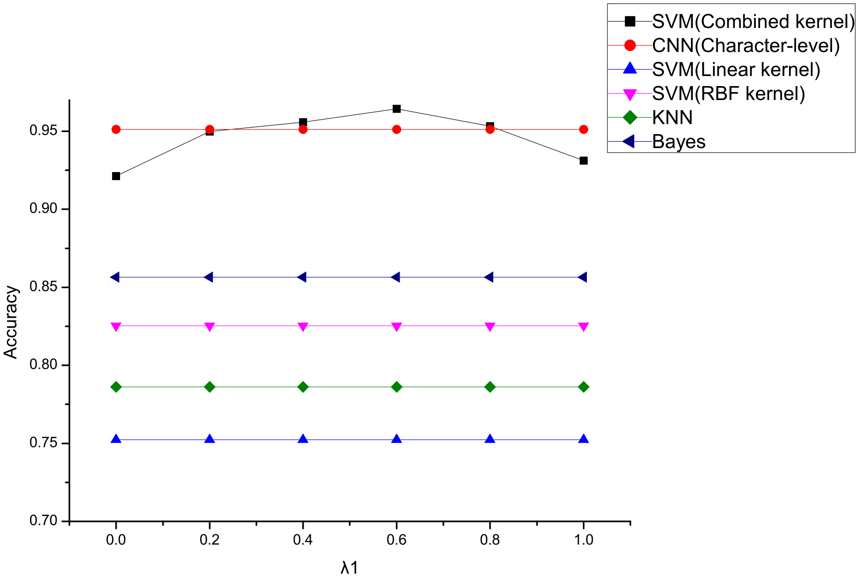


Figure 1. Curve: accuracy rate of Sogou corpus

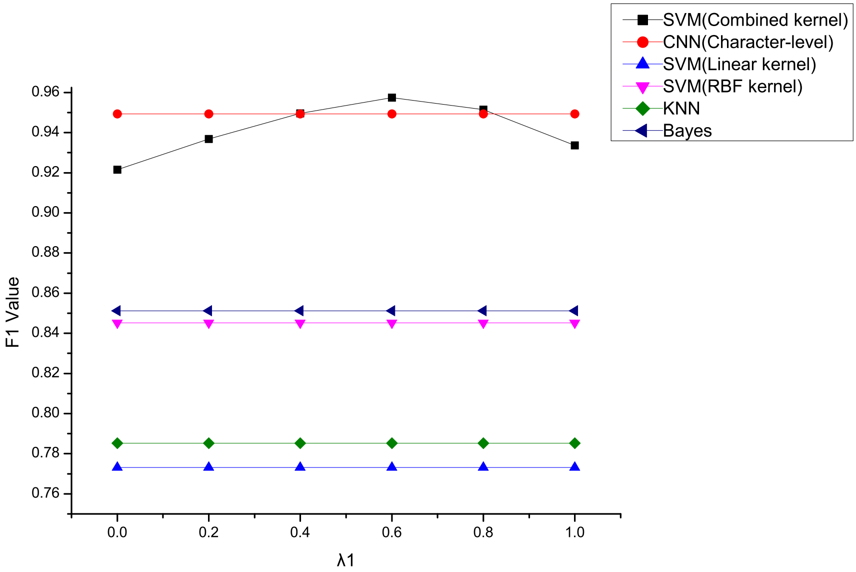


Figure 2. Curve: F1 value of Sogou corpus

	0	0.2	0.4	0.6	0.8	1
SVM (Combined kernel)	92.12 %	94.98 %	95.58 %	96.43 %	95.32 %	93.12 %
CNN (Character-level)	95.12 %					
SVM (Linear kernel)	75.23 %					
SVM (RBF kernel)	82.53 %					
KNN	78.62 %					
Bayes	85.65 %					

Table 3. Accuracy rate of Sogou corpus

	0	0.2	0.4	0.6	0.8	1
SVM (Combined kernel)	92.16 %	93.68 %	94.96 %	95.75 %	95.14 %	93.36 %
CNN (Character-level)	94.93 %					
SVM (linear kernel)	77.32 %					
SVM (RBF kernel)	84.52 %					
KNN	78.53 %					
Bayes	85.12 %					

Table 4. F1 value of Sogou corpus

As shown in the Figure 1, the accuracy rate is lower than that using character-level convolutional networks which is a very effective classification method in text categorization when λ_1 is 0, 0.2, and 1. However, the accuracy rate can be maintained at a high level when λ_1 is between 0.4 and 0.8. And the accuracy rate is always higher than that using KNN, Bayes and SVM with linear kernel and RBF kernel, proving the combination of the two is meaningful for Chinese text classification.

The values in Table 5 represent the accuracy rate of Fudan corpus and the values in Table 6 represent the values of F1 of Fudan corpus. The values obtained in Tables 5 and 6 are shown by the line chart in Figures 3 and 4, from which we can confirm that the combination of the two kernels is meaningful for Chinese text classification.

	0	0.2	0.4	0.6	0.8	1
SVM (Combined kernel)	95.34 %	95.75 %	96.68 %	96.23 %	95.56 %	95.14 %
CNN (Character-level)	95.15 %					
SVM (linear kernel)	89.41 %					
SVM (RBF kernel)	94.32 %					
KNN	90.21 %					
Bayes	95.59 %					

Table 5. Accuracy rate of Fudan corpus

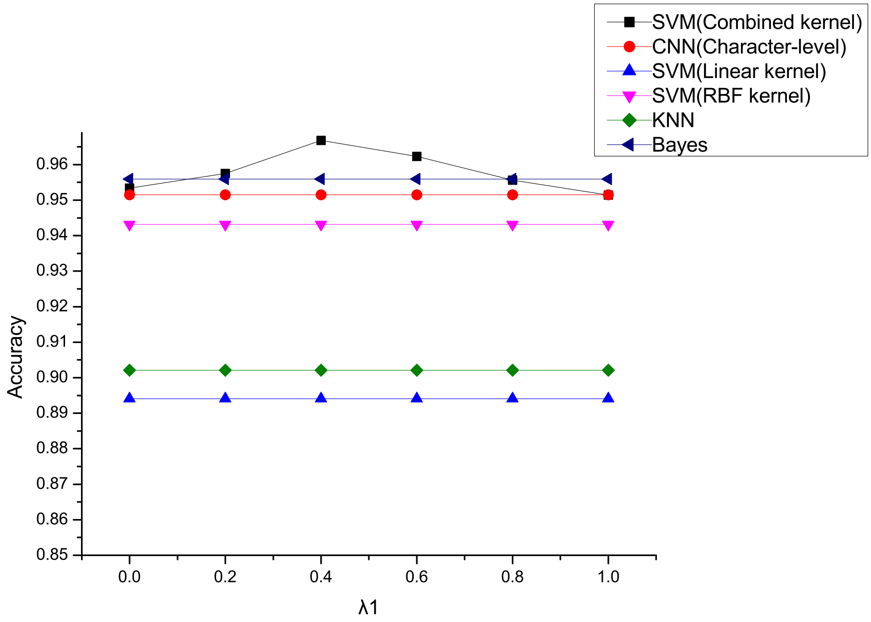


Figure 3. Curve: accuracy rate of Fudan corpus

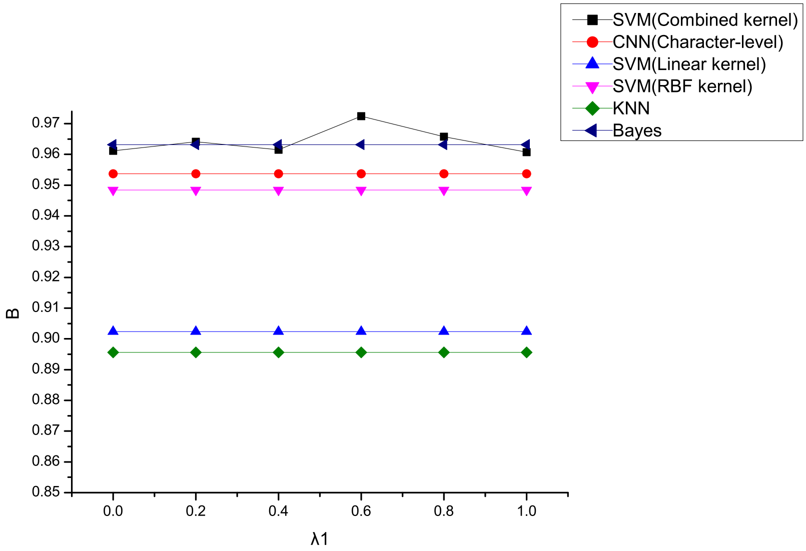


Figure 4. Curve: F1 value of Fudan corpus

	0	0.2	0.4	0.6	0.8	1
SVM (Combined kernel)	96.12 %	96.41 %	96.15 %	97.24 %	96.58 %	96.07 %
CNN (Character-level)	95.37 %					
SVM (linear kernel)	90.24 %					
SVM(RBF kernel)	94.84 %					
KNN	89.56 %					
Bayes	96.32 %					

Table 6. F1 value of Fudan corpus

5 CONCLUSION AND FUTURE WORKS

In this paper, a new combined kernel function is proposed based on semantic similarity and corpus similarity. Experiments show that the proposed method can improve the accuracy of classification compared with traditional machine learning methods.

In the future, we plan to study the statistical similarity matrix and how to capture the semantic information based on the weighting calculation. And we plan to further optimize the Chinese word-based similarity calculation method based on the HowNet.

Acknowledgements

This work is supported by the Shandong Provincial Natural Science Foundation, China (Grant No. ZR2014FQ018), BUPT-SICE Excellent Graduate Students Innovation Fund, National Natural Science Foundation of China (Grant No. 61471056). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] Dumais, S.—Platt, J.—Heckerman, D.—Sahami, M.: Inductive Learning Algorithms and Representations for Text Categorization. Proceedings of the Seventh International Conference on Information and Knowledge Management (CIKM '98), ACM, 1998, pp. 148–155, doi: 10.1145/288627.288651.

[2] Le, Q. V.—Mikolov, T.: Distributed Representations of Sentences and Documents. International Conference on Machine Learning, June 22–24, 2014, Bejing, China. Proceedings of Machine Learning Research (PMLR), Vol. 32, 2014, No. 2, pp. 1188–1196.

[3] Pennington, J.—Socher, R.—Manning, C. D.: GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543, doi: 10.3115/v1/D14-1162.

[4] Boser, B. E.—Guyon, I. M.—Vapnik, V. N.: A Training Algorithm for Optimal Margin Classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory (COLT '92), ACM, 1992, pp. 144–152, doi: 10.1145/130385.130401.

[5] Sain, S. R.: The Nature of Statistical Learning Theory. Technometrics, Vol. 38, 1996, No. 4, pp. 409–409.

[6] Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Nédellec, C., Rouveirol, C. (Eds.): Machine Learning: ECML-98. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1398, 1998, pp. 137–142, doi: 10.1007/BFb0026683.

[7] Amari, S.-I.—Wu, S.: Improving Support Vector Machine Classifiers by Modifying Kernel Functions. Neural Networks, Vol. 12, 1999, No. 6, pp. 783–789, doi: 10.1016/S0893-6080(99)00032-5.

[8] Nasir, J. A.—Karim, A.—Tsatsaronis, G.—Varlamis, I.: A Knowledge-Based Semantic Kernel for Text Classification. In: Grossi, R., Sebastiani, F., Silvestri, F. (Eds.): String Processing and Information Retrieval (SPIRE 2011). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7024, 2011, pp. 261–266.

[9] Altinel, B.—Diri, B.—Ganiz, M. C.: A Novel Semantic Smoothing Kernel for Text Classification with Class-Based Weighting. Knowledge-Based Systems, Vol. 89, 2015, pp. 265–277, doi: 10.1016/j.knosys.2015.07.008.

[10] Siolas, G.—d'Alché-Buc, F.: Support Vector Machines Based on a Semantic Kernel for Text Categorization. Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN 2000), 2000, IEEE, Vol. 5, pp. 205–209, doi: 10.1109/IJCNN.2000.861458.

[11] Mavroeidis, D.—Tsatsaronis, G.—Vazirgiannis, M.—Theobald, M.—Weikum, G.: Word Sense Disambiguation for Exploiting Hierarchical Thesauri in Text Classification. In: Jorge, A. M., Torgo, L., Brazdil, P., Camacho, R., Gama, J. (Eds.): Knowledge Discovery in Databases: PKDD 2005. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3721, 2005, pp. 181–192.

[12] Fellbaum, C.—Miller, G.: WordNet: An Electronic Lexical Database. MIT Press, 1998.

[13] Zhu, Y.-L.—Min, J.—Zhou, Y.-Q.—Huang, X.-J.—Wu, L.-D.: Semantic Orientation Computing Based on HowNet. Journal of Chinese Information Processing, Vol. 20, 2006, No. 1, pp. 14–20.

[14] Zhang, P.-Y.: A HowNet-Based Semantic Relatedness Kernel for Text Classification. Indonesian Journal of Electrical Engineering and Computer Science (TELKOMNIKA), Vol. 11, 2013, No. 4, pp. 1909–1915.

[15] Mei, J. L.: Tongyi ci Cilin. Shangai Cishu Chubanshe, 1985.

[16] Evangelopoulos, N. E.: Latent Semantic Analysis. Wiley Interdisciplinary Reviews, Cognitive Science, Vol. 4, 2013, No. 6, p. 683–692, doi: 10.1002/wcs.1254.

[17] Altinel, B., Ganiz, M. C.—Diri, B.: A Novel Higher-Order Semantic Kernel for Text Classification. 2013 International Conference on Electronics, Computer and Computation (ICECCO), 2013, IEEE, pp. 216–219, doi: 10.1109/ICECCO.2013.6718267.

[18] Altinel, B.—Ganiz, M. C.—Diri, B.: A Semantic Kernel for Text Classification Based on Iterative Higher-Order Relations Between Words and Documents. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L. A.,

Zurada, J. M. (Eds.): Artificial Intelligence and Soft Computing (ICAISC 2014). Springer, Cham, Lecture Notes in Computer Science, Vol. 8467, 2014, pp. 505–517.

[19] Altinel, B.—Ganiz, M. C.—Diri, B.: A Simple Semantic Kernel Approach for SVM Using Higher-Order Paths. Proceedings of the 2014 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA), IEEE, 2014, pp. 431–435, doi: 10.1109/INISTA.2014.6873656.

[20] Altinel, B.—Ganiz, M. C.—Diri, B.: A Corpus-Based Semantic Kernel for Text Classification by Using Meaning Values of Terms. Engineering Applications of Artificial Intelligence, Vol. 43, 2015, pp 54–66, doi: 10.1016/j.engappai.2015.03.015.

[21] Sparck Jones, K.: A Statistical Interpretation of Term Specificity and Its Application in Retrieval. Journal of Documentation, Vol. 28, 1972, No. 1, pp. 11–21, doi: 10.1108/eb026526.

[22] Salton, G.—Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval. Information Processing and Management, Vol. 24, 1988, No. 5, pp. 513–523, doi: 10.1016/0306-4573(88)90021-0.

[23] Ko, Y.—Seo, J.: Automatic Text Categorization by Unsupervised Learning. Proceedings of the 18th Conference on Computational Linguistics, Vol. 1, Association for Computational Linguistics, 2000, pp. 453–459, doi: 10.3115/990820.990886.

[24] Lertnattee, V.—Theeramunkong, T.: Analysis of Inverse Class Frequency in Centroid-Based Text Classification. IEEE International Symposium on Communications and Information Technology (ISCIT 2004), 2004, Vol. 2, pp. 1171–1176, doi: 10.1109/ISCIT.2004.1413903.

[25] Biricik, G.—Diri, B.—Sönmez, A. C.: A New Method for Attribute Extraction with Application on Text Classification. Fifth International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control (ICSCCW 2009), IEEE, 2009, pp. 1–4, doi: 10.1109/ICSCCW.2009.5379479.

[26] Biricik, G.—Diri, B.—Sönmez, A. C.: Abstract Feature Extraction for Text Classification. Turkish Journal of Electrical Engineering and Computer Sciences, Vol. 20, 2012, No. 1, pp. 1137–1159.

[27] Parsons, S.: Introduction to Machine Learning by Ethem Alpaydin, MIT Press, 0-262-01211-1, 400 pp. The Knowledge Engineering Review, Vol. 20, 2005, No. 4, pp. 432–433.

[28] Cristianini, N.—Shawe-Taylor, J.—Lodhi, H.: Latent Semantic Kernels. Journal of Intelligent Information Systems, Vol. 18, 2002, No. 2-3, pp. 127–152.

[29] Loper, E.—Bird, S.: NLTK: The Natural Language Toolkit. ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics (ETMTNLP '02), Vol. 1, 2002, pp. 63–70.

[30] Pedregosa, F.—Varoquaux, G.—Gramfort, A.—Michel, V.—Thirion, B.—Grisel, O.—Blondel, M.—Prettenhofer, P.—Weiss, R.—Dubourg, V.—Vanderplas, J.—Passos, A.—Cournapeau, D.—Brucher, M.—Perrot, M.—Duchesnay, E.: SciKit-Learn: Machine Learning in Python. Journal of Machine Learning Research, Vol. 12, 2011, No. 10, pp. 2825–2830.

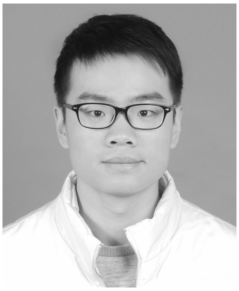
[31] El Kourdi, M.—Bensaid, A.—Rachidi, T.-E.: Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm. The Workshop on Computa-

tional Approaches to Arabic Script-Based Languages (Semitic '04), 2004, pp. 51–58, doi: 10.3115/1621804.1621819.

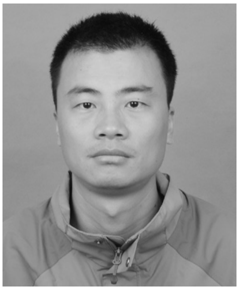
[32] Syiam, M. M.—Fayed, Z.T.—Habib, M. B.: An Intelligent System for Arabic Text Categorization. International Journal of Cooperative Information Systems, Vol. 6, 2006, No. 1, pp. 1–19.

[33] Zhang, X.—Zhao, J.—LeCun, Y.: Character-Level Convolutional Networks for Text Classification. Proceedings of the 28^{th} International Conference on Neural Information Processing Systems (NIPS '15), Vol. 1, 2015, pp. 649–657.

Haipeng Yao is currently Associate Professor with the School of Information and Communication Engineering, from the State Key Laboratory of Networking and Switching Technology in Beijing University of Posts and Telecommunications. His main research interests include future network architecture, big data for networking, the architecture and key technology of the new generation mobile communication system.



Bo Zhang is currently Postgraduate Student with the School of Information and Communication Engineering, from the State Key Laboratory of Networking and Switching Technology in Beijing University of Posts and Telecommunications. His main research interests include natural language processing, big data and deep learning for networking.



Peiying Zhang received his Master degree from China University of Petroleum (East China) in 2006. He is currently Lecturer in the College of Computer and Communication Engineering from China University of Petroleum (East China). He is a Ph.D. candidate in information and communication engineering, from the State Key Laboratory of Networking and Switching Technology in Beijing University of Posts and Telecommunications. His research interests include natural language processing, semantic computing, future internet architecture, network virtualisation, and data center network.

Maozhen Li is currently Professor in the Department of Electronic and Computer Engineering at Brunel University London, UK. He received his Ph.D. degree from the Institute of Software, Chinese Academy of Sciences in 1997. He was Post-Doctoral Research Fellow in the School of Computer Science and Informatics, Cardiff University, UK in 1999–2002. His research interests are in the areas of high performance computing, big data analytics and intelligent systems. He is on the Editorial Boards of a number of journals. He has over 150 research publications in these areas. He is a Fellow of the British Computer Society.