International Journal of Computational Intelligence Systems
Vol. **14(1)**, 2021, *pp.* 367–375
DOI: https://doi.org/10.2991/ijcis.d.201207.001; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

# A Short Text Classification Method Based on Convolutional Neural Network and Semantic Extension

Haitao Wang[1], Keke Tian[1], Zhengjiang Wu[1,*], Lei Wang[2]

[1]*Henan Polytechnic University, Jiaozuo City, Henan Province, 454003, China*
[2]*Louisiana State University, Baton Rouge, Louisiana, 70803, United States*

**ARTICLE INFO**

*Article History*
Received 06 July 2020
Accepted 22 Nov 2020

*Keywords*
Short text
Classification
CNN
Semantic extension
Attention mechanism
Conceptualization

**ABSTRACT**

In order to solve the problem that traditional short text classification methods do not perform well on short text due to the data sparsity and insufficient semantic features, we propose a short text classification method based on convolutional neural network and semantic extension. Firstly, we propose an improved similarity to improve the coverage of the word vector table in the short text preprocessing process. Secondly, we propose a method for semantic expansion of short texts, which adding an attention mechanism to the neural network model to find related words in the short text, and semantic expansion is performed at the sentence level and the related word level of the short text respectively. Finally, the feature extraction of short text is carried out by means of the classical convolutional neural network. The experimental results show that the proposed method is feasible during the classification task of short text, and the classification effectiveness is significantly improved.



## 1. INTRODUCTION

In recent years, the rapid development of a new generation of information technology represented by cloud computing and big data has promoted the arrival of a new era of the Internet. While the Internet brings convenience to people's lives, a large amount of text data is also generated every day. Among them, short texts in the form of comments, Weibo, Q&A, and so on have the characteristics of rapid growth and huge number. These short texts usually have obvious limitations: lack of sufficient contextual information, there may be polysemy, and sometimes there are spelling errors. How to quickly and effectively extract truly valuable information from the massive short text data is precisely the problem that needs to be solved in the field of natural language processing (NLP), and it also has far-reaching significance.

Traditional machine learning text classification approaches such as Naive Bayes (NB) [1,2], Support Vector Machine (SVM) [3,4], K Nearest Neighbors (KNN) [5,6] and Decision Trees [7], and so on, often use the Bag of Words (BOW) [8] model to represent text, and the Term Frequency-Inverse Document Frequency (TF-IDF) to represent the word weights. However, the BOW model used by these classification approaches [9], sentences and documents are considered to be independent of each other, and there is no contextual relationship between them, so that the fine-grained semantic information in the text may not be effectively extracted, and the BOW model has problems such as dimensional disaster and sparse data [10].

With the emergence of deep learning in recent years, models based on deep neural networks have attracted more and more researchers' attention in the field of NLP, Such as recurrent neural network (RNN) model and convolutional neural network (CNN) model. Mikolov *et al.* [11] proposed the famous RNN-based model utilized RNN to take the expression of the whole sentence into consideration, this model can capture the long-term dependencies and learn the meaning of words. Kim *et al.* [12] proposed a multi-size filter CNN model to extract richer text semantic features, and produced good results on the text classification task, thus becoming one of the most representative models in the field of NLP. Yang *et al.* [13] propose a hierarchical attention network for document classification, the model not only applies the attention mechanism on the document hierarchy, but also on the word and sentence levels, so that it can pay attention to the more and more important contents when constructing the document representation. Zhou *et al.* [14] proposed a network model based on mixed attention mechanism for Chinese short text classification, which not only considered word-level text features and character-level features, but also extracted semantic features related to classification through the attention mechanism.

However, if only the neural network model is used to extract the abstract features of short text semantic information, the classification effect will largely depend on the number of layers of the neural network, so it will cause a geometric level increase in the number of

* *Corresponding author. E-mail:* jz_wht@hpu.edu.cn

parameters of the entire model, thereby significantly increasing the training time of the model. Therefore, in order to overcome the lack of short text semantic information, the external knowledge base can be used to expand the semantics of the text, thereby enriching the semantic features of the short text.

In summary, in this paper we propose a short text classification method based on CNN and semantic extension (SECNN). Under the condition that the neural network model has a certain number of layers, we propose a novel method to find related words in short texts, and perform semantic expansion at the sentence level and related word level of the short text at the same time, thereby the classification effect of the short text can be improved.

To sum up, our contributions are as follows:

Firstly, in the text preprocessing process, we propose an improved Jaro–Winkler similarity to find possible spelling errors in short text, so the coverage of the pretraining word vector table can be improved.

Secondly, we propose a CNN model based on attention mechanism to find related words of short text, and then use external knowledge base to conceptualize short text and related words respectively, thus expanding the semantic features of short text.

Finally, we use the classical CNN model to extract short text features and complete the classification process.

The rest of this paper is organized as follows. In Section 2, the related works regarding text classification are reviewed. Section 3 presents a short text preprocessing method. In Section 4, we present a short text classification method in details. Section 5 conducts the extensive experiments. Section 6 discusses the experiment results. The conclusion was drawn in Section 7.

## 2. RELATED WORK

In the traditional text classification methods, the corresponding text semantic features are usually ignored during the classification process, and the fine-grained semantic information in the text cannot be effectively extracted, resulting in a low interpretability of the final classification results. In order to settle these problems, Wei *et al.* [15] proposed text representation and feature selection strategies for Chinese text classification based on n-grams, in the feature selection strategy, preprocessing within classes is combined with feature selection between classes. Post *et al.* [16] proposed the use of part-of-speech (POS) tagging and tree kernel technology to extract the explicit and implicit features of text. Gautam *et al.* [17] proposed a unary word segmentation technique and semantic analysis to represent the features of the text. Song *et al.* [18] used the probabilistic knowledge base to conceptualize short text, thereby improved the understanding of text semantics during the text classification progress. Zhang *et al.* [19] proposed a short text classification method based on the latent dirichlet allocation (LDA) topic model, which further solved the problem of context dependence of short text. Although these methods can extract rich text feature information relatively, they also have some limitations.

Word embedding is currently the most commonly used, and it is also the most effective word vector representation for retaining semantic and grammatical information. The word vector technology was first proposed by Hinto [20]. Collobert *et al.* [21] used the pretrained word vectors and CNN technology to classify texts for the first time, demonstrating the effectiveness of CNNs in text processing. Mikolov *et al.* [22] used the neural network model to learn a new vector representation called word vector or word embedding, which contains the grammatical and semantic information of the word. Compared with the traditional word bag model representation, word vectors are characterized by low dimensionality, denseness and continuity. So we also used word vector technology in text preprocessing.

In recent years, natural language modeling methods have been relying on CNN to learn word embedding and they have shown promising results. Our method also use CNN to automatically and effectively extract short text features. Sotthisopha *et al.* [23] used the clustering of word vectors to find semantic units. At the same time, Jaro–Winkler similarity was used in the process of text preprocessing to find spelling errors in the text, but the Jaro–Winkler similarity used in this method only considers the matching degree of common prefixes between strings, and ignores the suffix matching of strings. Zhang *et al.* [24] proposed a character-level CNN text classification model (CharCNN), which can obtain more fine-grained semantic features, but the model also ignores the word-level semantic features in the text.

Compared with the traditional machine learning text classification methods, the deep neural network model can effectively simulate the information processing process of the human brain, which can further extract more abstract semantic features from the input features, thereby, the information that the model depends on when it is finally classified is more reliable. However, short text usually lacks sufficient context information and semantic features. If we only rely on increasing the number of neural networks to improve the classification effect of short text, it will cause a geometric increasing in the number of parameters of the entire model. Wang *et al.* [25] proposed a method for classifying short texts using external knowledge bases and CNNs. While conceptualizing short texts enriches semantic features, it also captures the finer-grained features in aspect of character level. Wu *et al.* [26] proposed two methods (CNN-HE and CNN-VE) to combine word and contextual embeddings, then apply CNNs to capture semantic features. The paper also uses an external knowledge base to conceptualize the target word, but does not consider the semantic expansion of the entire sentence. This document is close to the short text classification method proposed in this paper, and the method in this paper is further studied and improved on the basis of it.

Based on the attention mechanism, we can dynamically extract the main features of the text instead of directly processing the information of the entire text, so this mechanism has been widely used [27–29]. Peng *et al.* [30] proposed a bidirectional long short-term memory (LSTM) neural network based on attention mechanism to capture the most important semantic information in sentences and use it for relationship classification. Zhang *et al.* [31] proposed bidirectional gated recurrent units which integrates a novel attention pooling mechanism with max-pooling operation to force the model to pay attention to the keywords in a sentence and maintain the most meaningful information of the text automatically. Wang *et al.* [32] proposed a novel CNN architecture for relationship

classification, which uses two levels of attention mechanisms to better identify the context. Qiao *et al.* [33] proposed a word–character attention model for Chinese text classification, this model integrates two levels of attention models: word-level attention model captures salient words which have closer semantic relationship to the text meaning, and character-level attention model selects discriminative characters of text.

Through these methods, when faced with the problem of insufficient semantic information of short texts, they did not fully consider the semantic expansion of sentence level and word level at the same time. Therefore, based on the neural network based on the attention mechanism, we propose a novel method to find related words in short texts, and perform semantic expansion at the sentence level and related word level of the short text at the same time, also we propose the improved Jaro–Winkler similarity to find possible spelling errors in short texts in the preprocessing of short texts, thereby improving the coverage of the pretraining word vector table.

## 3. SHORT TEXT PREPROCESSING

This paper uses external corpus and Word2vec technology to train short text into a word vector table. Since the accuracy of the generated word vector will affect the subsequent text feature extraction effect of the CNN. Therefore, in the process of short text vectorization, each word should match the words in the word vector table as much as possible. However, due to the characteristics of short text, some words are often spelled incorrectly, which will lead to the failure to find the corresponding word from the word vector table in the subsequent word vectorization process, thus ignoring the key features that the word may represent.

In text preprocessing, we utilize the Jaro–Winkler similarity to find spelling errors in text, and carry out similarity comparison between the words in the short text and the similar words in the word vector table, if the two are partially different but have a high similarity, it means that the corresponding word in the short text may be misspelled, then replace it according to the corresponding words in the word vector table. The misspelling position of short text is very random, it may be in the second half of the word, or occur in the first half of the word. However, the Jaro–Winkler distance metric only considers the matching degree of the common prefix between the strings, that is, the spelling error of the word appears in the second half. But at the same time, it also ignores the suffix matching of the string, that is, the spelling error occurs in the first half of the word. For example, "*argument*" and "*argument,*" there is only one common prefix, so the Jaro–Winkler distance cannot reflect the matching degree of these two strings well.

Based on the Jaro–Winkler distance, this paper proposes an improved Jaro–Winkler similarity with both prefix matching and suffix matching, defined in formula (1) as follows:

$$sim_w = sim_j + \left(l + l'\right) p' \left(1 - sim_j\right) \tag{1}$$

Since spelling errors in short text words are more common in the second half of the word, the common prefix matching of the two words is given more weight, and the similarity result must not exceed 1. So $l$ is the length of the selected common prefix of two English words, maximum value is 3, $l'$ is the length of the selected common suffix of two English words, maximum value is 2, $p'$ is the scaling factor for how much the score is adjusted upward for having common prefixes and suffixes, its value cannot exceed 0.2. $sim_j$ is the Jaro similarity of two English words, as defined in formula (2), its result range is between 0 and 1, $s_1$ and $s_2$ respectively represent the two English words to be compared, where $|s_i|$ is the number of characters of the corresponding word, that is, the length of the string, $m$ represents the number of matched characters ignoring the character order and $t$ is one-half of the number of character conversions required to convert one word to another.

$$sim_j = \begin{cases} 0 & m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & m \neq 0 \end{cases} \tag{2}$$

Therefore, in the process of vectorization preprocessing of short text, for words that are not covered in the data set, if the number of matching characters between the corresponding words in the word vector table does not exceed a certain threshold, then the two words deemed as match. Further count all the words in the word vector table that match the uncovered word, calculate their similarity according to formula (1) respectively, and finally select the word with the highest similarity and exceeding the minimum similarity threshold, then replace the uncovered words in the data set with the words in the corresponding word vector table. It can be seen from the above that after this preprocessing process, the spelling errors of short text in the data set can be found as soon as possible, thereby improving the coverage of the Word2vec word vector table.

## 4. THE PRESENTED METHOD

As shown in Figure 1, the overall framework of our proposed short text classification method is composed of four main components. Firstly, we use the improved Jaro–Winkler similarity to find possible spelling errors in short texts in the preprocessing of short texts. Secondly, related words of short text are found through a CNN model based on the attention mechanism. Thirdly, the external knowledge base Probase is used to conceptualize the short text and the related words separately to generate the corresponding word vector matrix. Finally the classic CNN model is used to extract short text features to complete the classification process..

### 4.1. Find Related Word

In short text, usually only a few words can represent the semantics of the entire sentence, and most words do not contribute much to the semantic features of the short text. According to the different effects of different words on the classification effect, adding an attention mechanism to the neural network model can enable the model to find the words that really affect the semantics in the context through the attention mechanism when establishing the relationship between the current word and the context. This paper refers to these words as related words in short text.

In order to find related words of short text, this paper designs a CNN model, which consists of a single convolution layer and a pooling layer with an attention mechanism, as shown in Figure 2.

The convolutional layer is composed of a series of filters with learnable parameters. In this layer, by changing the weight values of these

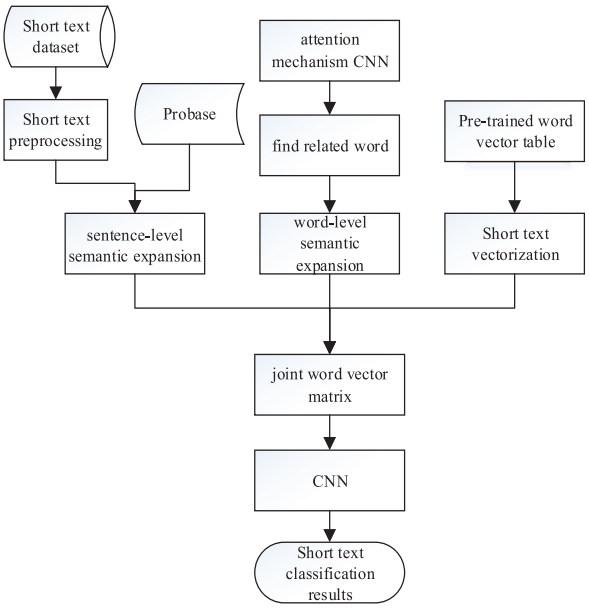


**Figure 1** | Overview of short text classification method.

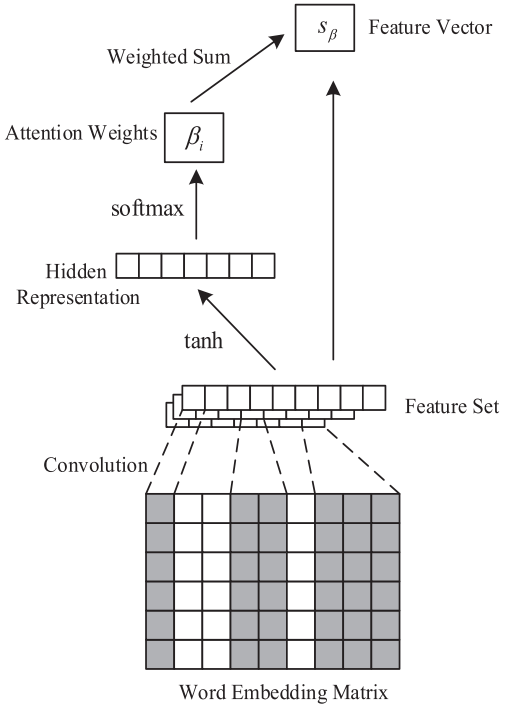


**Figure 2** | Structure diagram of attention mechanism.

filters, these filters can obtain higher activation values for specific features. Therefore, it is possible to extract higher-level short text semantic features. The width of the filter is fixed to the value $m$, which is the same as the dimension of the word vector, and the height of the filter is $h$. Such as, the feature $s_i$ can be extracted through a filter $w \epsilon R^{h*m}$, defined in formula (3) as follows:

$$s_i = f\left(w \cdot [v_i : v_{i+h-1}] + b\right) \tag{3}$$

where $f$ is a nonlinear function, in this paper we use *ReLU* as the nonlinear function, operator $(\cdot)$ represents convolution operation, $b$ is a bias term, $[v_i : v_{i+h-1}]$ represents a sequence of words of length $h$, $v_i$ represents a word.

Using filters of different heights in the convolution operation, a feature set $S$ can be obtained by sliding the filter window, defined in formula (4) as follows:

$$S = [s_1, s_2, \ldots, s_l] \tag{4}$$

where $s_i$ is the feature vector generated by the convolution operation of each filter, $l$ is the size of the set $S$.

By performing the *tanh* activation operation on the feature set $S$, the hidden representation $u_i$ of the feature vector $s_i$ can be obtained, as defined in formula (5). Then performing *softmax* operation on the parameter $u_i$, the attention weight $\beta_i$ of the feature vector $s_i$ is obtained, as defined in formula (6). Finally, each feature vector $s_i$ is weighted sum according to its attention weight $\beta_i$ to obtain the pooled feature vector $s_\beta$, as defined in formula (7).

$$u_i = tanh\,(w \cdot s_i + b) \tag{5}$$

$$\beta_i = softmax\left(w' \cdot u_i\right) \tag{6}$$

$$s_\beta = \sum_1^l \beta_i s_i \tag{7}$$

In the word vector space, semantically similar words usually have a similar distance, therefore the feature vector $s_\beta$ obtained through the attention mechanism is calculated with the Euclidean distance of the word vector corresponding to each word of the short text, and the closest word is the related word of the short text.

## 4.2. Short Text Semantic Expansion

Short text usually lacks sufficient contextual information, sometimes does not follow the grammatical rules of natural language, also there may be polysemy. This section uses the external knowledge base Probase to semantically extend short text and generate conceptual vectors of short text, which can effectively enrich the semantic features of short text, so as to achieve the purpose of short text semantic expansion.

By scanning the Probase knowledge base, for each instance, we will obtain a corresponding series of related concepts, then score the instances, concepts and their relationships. For a given short text instance, we can acquire the corresponding conceptual relationship through the conceptual application programming interface (API) provided by Probase. Here we remark the concept vector as $C = \{< c_1, w_1 >, < c_2, w_2 >, \ldots, < c_k, w_k >\}$, where $c_i$ is a concept in the knowledge base, and $w_i$ is a weight to represent the relevance of the short text associated with $c_i$.

After obtaining the concept sequence relationship of a short text instance, the next step is to generate the corresponding word vector. In this paper, the Word2vec model is used to complete the pretraining vectorization operation. Since the conceptualization of short text contains a series of concept weights, the corresponding weights

must also be taken into account when generating word vectors. The formula for vectorization is defined in formula (8) as follows:

$$W_c = w_1 v_{c1} \oplus w_2 v_{c2} \oplus \ldots \oplus w_k v_{ck} \tag{8}$$

where $W_c$ represents the word vector matrix conceptualized by short text, $w_i$ represents the weight of the degree of association between the short text and the concept $c_i$, $v_{ci}$ represents the word vector corresponding to the concept $c_i$, and $\oplus$ is the concatenation operation.

The sentence level conceptual sequence relationships obtained through short text conceptualization can extract richer short text semantic information. However, the concept sequence relationship at the word level in short text is also important because it can extract more fine-grained text semantic information. Therefore, on the basis of the conceptualization of short text at the sentence level, this paper puts forward the conceptualization of related words.

After obtaining the related words in the short text by using the attention mechanism, the related words can be conceptualized to obtain the conceptual sequence, as $C' = \{< c'_1, w'_1 >, < c'_2, w'_2 >, \ldots, < c'_k, w'_k >\}$, Where $c'_i$ is the concept of related words in the knowledge base, and $w'_k$ is the weight corresponding to this concept.

After the concept sequence relationship of related word instances is obtained above, the next step is to generate corresponding word vectors. Use the Word2vec model to complete the pretraining vectorization operation. Since the conceptualization of related words also includes a series of concept weights, when generating word vectors, the corresponding weights must also be taken into account. The formula for vectorization is defined as follows:

$$W'_c = w'_1 v'_{c1} \oplus w'_2 v'_{c2} \oplus \ldots \oplus w'_k v'_{ck} \tag{9}$$

where $W'_c$ represents the word vector matrix conceptualizing related words, $v'_{c1}$ represents the word vector corresponding to the concept $c'_i$, and $\oplus$ is the concatenation operation.

So far, the word vector matrix $W_w$ of the short text, the word vector matrix $W_c$ of the short text conceptualization, and the word vector matrix $W'_c$ of the related word conceptualization have been obtained.

## 4.3. CNN Short Text Classification

When using CNNs to classify short text, it is necessary to represent the short text as a matrix as the input of the network model. Therefore, it is necessary to cascade the word vector matrix $W_w$ of short text, the word vector matrix $W_c$ of conceptualization of short text, and the word vector matrix $W'_c$ of conceptualization of related words, Then form the joint word vector matrix $W$ of short text, as shown in Figure 3, the corresponding formula is defined as follows:

$$W = W_w \oplus W_c \oplus W'_c \tag{10}$$

where $W \in R^{(n+2k)*m}$, $n$ is the number of words in the short text, $k$ represents the number of concepts conceptualized as short text and related words, $m$ is expressed as the dimension of the word vector.

Next, use the classic CNN model to perform convolution processing, pooling processing and fully connected *Softmax* classification, the structure of which is shown in Figure 4.

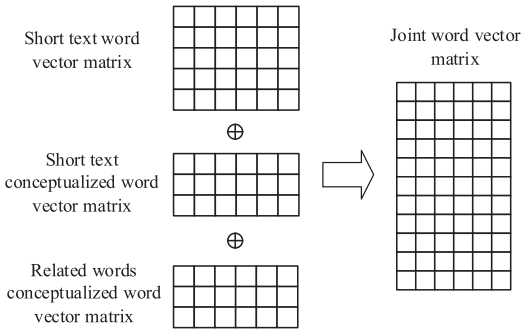


**Figure 3** Input layer joint word vector matrix.

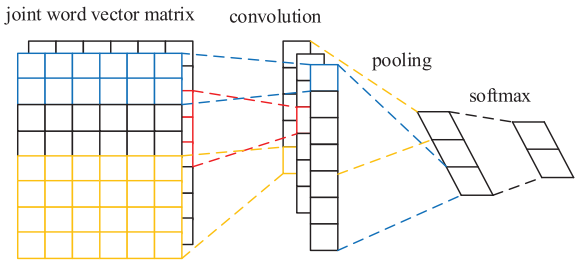


**Figure 4** Convolutional neural network (CNN) short text classification structure.

First, the joint word vector matrix is used as the input of the convolutional layer, and a variety of height-size filters are used to perform the convolution operation to extract the features of the short text and generate a set of feature vectors.

In the pooling layer, Max Pooling is used, that is, the maximum value input in a certain area is used as the output of the area. This can reduce the number of parameters in the network, and can also effectively prevent overfitting and improve the generalization ability of the model. Through maximum pooling, a fixed-length vector can be extracted from the feature map. The specific calculation process is shown as follows:

$$s_{max} = max(s_i) \tag{11}$$

where $s_i$ represents a feature map formed by a filter performing convolution operation on short text, $0 < i \leq M$, $M$ is the number of feature maps. With maximum pooling, each feature map will get a maximum value. After pooling all the feature maps, each feature value needs to be stitched together to obtain the final feature vector of the pooling layer. At the same time, in order to reduce the phenomenon of overfitting, dropout and L2 regularization mechanisms are introduced in the hidden layer to randomly set some feature vectors to zero.

As the last component of the entire CNN, the fully connected layer plays the role of classifier. The classification model ultimately needs to complete the classification of the input short text. After the fixed-length feature vectors obtained by the convolutional layer and the pooling layer processing, a fully connected softmax layer is introduced to complete the classification operation. The softmax classification layer converts a series of classification score values into

classification probabilities. A larger classification score value indicates a greater likelihood of belonging to the corresponding category. Conversely, a category with a smaller classification score value has a lower probability.

## 5. EXPERIMENTAL SETUP

To validate the classification result, we conduct the extensive experiments on the different short text data sets. The experimental configuration mainly includes the Intel(R)i7-7700 3.60GHz processor, 16GB memory and Python3.7 programming environment.

### 5.1. Datasets

In order to demonstrate the effectiveness of the short text classification method proposed, we adopt the classical short text data sets which widely used in recent years for text classification tasks, and basic information of data set is listed as follows:

- *MR*. The MR data set is an English movie review data set, with a total of 10662 data, the number of categories is 2, half of the positive and negative examples. The average sentence length is 20.
- *TREC*. The TREC data set is a question and answer data set, with a total of 6452 data, including 5952 data in the training set, 500 data in the test set, and 6 categories. The average sentence length is 10.
- *AG News*. The AG News data set is an English news article data set, with a total of 127,600 data, including 120,000 data in the training set, 7600 data in the test set, and 4 categories. The average sentence length is 7.
- *Twitter*. The Twitter data set is an English sentiment classification data set, with a total of 11,209 data, including 8204 data in the training set and 3005 data in the test set. The number of categories is 3, including positive, neutral and negative. The average sentence length is 19.
- *SST-2*. The SST-2 data set is an extension of the MR data set, with a total of 9613 data. The number of categories is 2 and the average sentence length is 19.

### 5.2. Experimental Parameters

During our classification method, the Word2vec tool is used to train the word vectors on the data sets, and the size of the convolution kernel is *3xdim, 4xdim, 5xdim*, the number of convolution kernels is 100, the batch_size is 64 and the learning rate is 0.001. To prevent overfitting phenomenon happens, a dropout mechanism was introduced during training, with a Dropout rate of 0.5.

The evaluation indicators used in the experimental part are *Accuracy* and *F1* value to measure the classification effect of short text, the formulas are defined in (12) and (13) as follows:

$$Accuracy = \frac{TP + TN}{P + N} \tag{12}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{13}$$

where $TP$ is the number of actual positive classes and predicted to be positive classes, $TN$ is the actual negative class and predicted to be negative classes, $P$ and $N$ represent the number of positive and negative classes, respectively.

## 6. EXPERIMENTAL RESULTS AND ANALYSIS

In order to validate that the short text classification method based on CNN and SECNN we proposed has a better classification effectiveness, the CNN-rand model and CNN-static model proposed by Kim are compared. Then, Zhang's character-level CNN text classification model (CharCNN) is compared. At the same time, we also compared our work with Wang's short text classification method based conceptualization and convolutional neural network method (WCCNN). Finally, the methods most similar to our work (CNN-HE, CNN-VE) proposed by Wu are compared with our classification method.

Among them, In the CNN-rand model, all word vectors are randomly generated and trained as model parameters. In the CNN-static model, Word2vec pretrained vectors are used, and the word vectors are no longer updated during the training process. At the same time, if there are words in the short text that are not in the pretrained dictionary, they are replaced by randomly generated vectors. In the WCCNN short text classification method, the external knowledge base is used to semantically extend the short text, and the word vector matrix of the short text and the conceptualized word vector matrix are combined as the input of the CNN. In the CNN-HE, word embedding matrix and contextual embedding matrix are concatenated in horizontal orientation to obtain the final embedding matrix. And in the CNN-VE, word embedding matrix and contextual embedding matrix are concatenated in vertical orientation to obtain the final embedding matrix.

First, the classification accuracy of the six methods on the short text data sets MR, TREC, AG News, Twitter and Sogou News is tested through experiments. The experimental results are shown in Table 1.

As can be seen from Table 1, the classification method proposed by us has better accuracy results than other six methods. Among them, CNN-rand is closer to CNN-static, and the latter is higher than the former. This is because the former's word vector model is randomly initialized and modified during training. The latter is a word vector obtained by Word2vec training in advance, which can better express

**Table 1** | Accuracy comparison of different classification methods (%)

| Methods | MR | TREC | AG News | Twitter | SST-2 |
|---|---|---|---|---|---|
| CNN-rand | 76.72 | 86.17 | 84.38 | 56.64 | 82.59 |
| CNN-static | 80.77 | 89.26 | 85.34 | 57.21 | 86.23 |
| CharCNN | 76.93 | 76.05 | 78.31 | 45.14 | 81.25 |
| WCCNN | 82.95 | 90.68 | 85.76 | 57.74 | 86.93 |
| CNN-HE | 82.29 | 91.28 | 85.84 | 56.91 | 87.16 |
| CNN-VE | 82.08 | 91.05 | 85.80 | 57.53 | 86.98 |
| **SECNN** | **83.89** | **91.34** | **86.02** | **57.93** | **87.37** |

the text semantics. In this paper, the improved Jaro–Winkler similarity is used in text preprocessing to find possible spelling errors in short text and replace them, which improves the coverage of the word vector table in the data set.

We can see that the CharCNN does not perform well in these short text data sets, the reason is that short text usually lacks sufficient semantic features, if only extracting features from the character level will not achieve a good classification effect. Compared with the WCCNN, CNN-HE, CNN-VE, SECNN not only uses short text conceptualization but also proposes related word conceptualization to further improve the semantic information of short text, the problem of insufficient semantic information in short texts has been fully resolved, and the classification effect also be improved.

In order to better reflect the advantages of the short text classification method proposed in this paper, five other different models were selected to perform multiple iteration experiments on the *MR* data set, and then the results were compared, as shown in Figure 5.

Where the abscissa of the graph is the number of CNN training epochs, and the ordinate is the accuracy of the model. It can be clearly seen from the figure: although the accuracy of each classification method is gradually increasing with the increase of the number of iterations, the advantages of SECNN and WCCNN are already reflected in the 1st epoch, and when the number of epoch is 5th, the accuracy rate is the highest, and the subsequent values are basically stable, indicating that the model has converged. It can be seen that the short text classification method proposed in this paper is also superior to other classification methods in terms of stability.

In order to validate the other evaluation indicators, the classification effect of the method in this paper has also been improved to a certain extent. Next, the classification result *F1* value of the method proposed in this paper is compared with six comparison methods on the *MR* data set and *AG News* data set. The experimental results are shown in Figures 6 and 7.

As can be seen from Figures 6 and 7, the *F1* value of the WCCNN for *MR* dataset classification is significantly higher than the classic text classification method such as: CNN-rand and CharCNN, then CNN-HE and CNN-VE are slightly below the WCCNN's *F1* value. From the result, we also found that SECNN's *F1* value is slightly better than WCCNN. This shows that the method proposed in this paper is feasible in the classification task of short text, and the classification effect has been significantly improved.

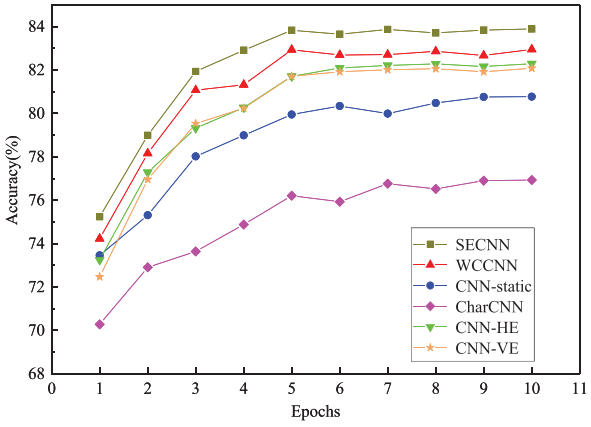


**Figure 5** Comparison of accuracy under different epochs.

## 7. CONCLUSION

Aiming at the problem that the traditional short text classification method relies heavily on the number of neural network layers and do not perform well on short text due to the data sparsity and insufficient semantic features, we propose a short text classification method based on CNN and semantic expansion. In order to improve the coverage of the pretrained word vector table in the process of short text vectorization, the improved Jaro–Winkler similarity is used to find possible spelling errors in short text during text preprocessing, thus it can more accurately match the corresponding words in the corpus. At the same time, facing the problem of limited semantic information that short text can provide, we introduce an external knowledge base to conceptualize short text and related words in short text, extend the semantics of short text. Experiment results demonstrate that the method proposed is feasible in the classification task of short text, and the classification effectiveness is improved remarkably. When this classification method obtains the

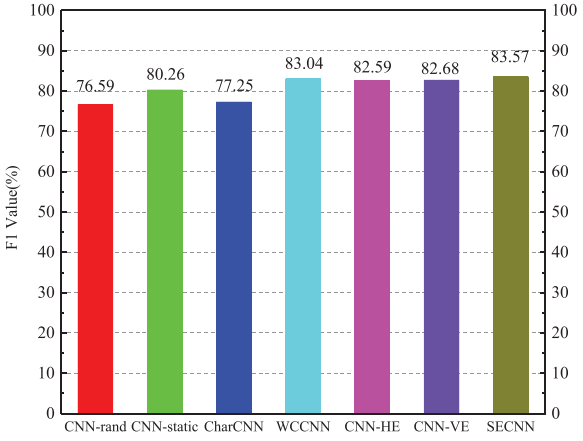


**Figure 6** Comparison of *F1* values of different classification methods on *MR* data set.

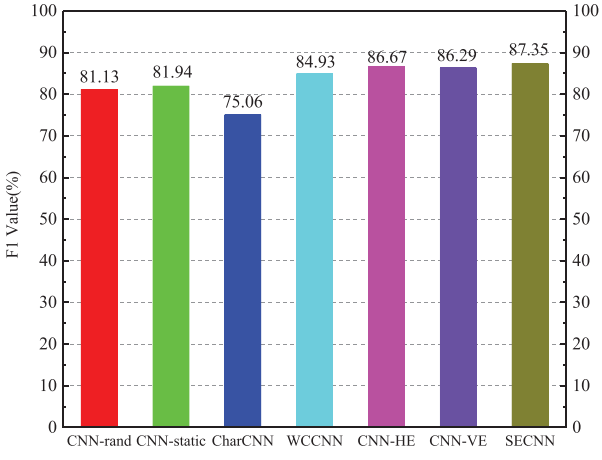


**Figure 7** Comparison of *F1* values of different classification methods on *AG News* data set.

related words of short text, because it involves a large number of word vector distance calculations, it's inevitable to need researching the time consumption. In the following research, time complexity will be taken into account, and the method of obtaining related words in short text will be optimized to improve the classification efficiency of short text.

## AUTHORS' CONTRIBUTIONS

Haitao Wang contributed to the conception of the study; Keke Tian performed the experiment; Keke Tian performed the data analyses and wrote the manuscript; Zhengjiang Wu provided technical support; Lei Wang helped perform the analysis with constructive discussions, also helped check the grammar of the paper.

## ACKNOWLEDGMENTS

This work is support by the National Natural Science Foundation of China (No. 11601129 61503124), Henan Science and Technology Key Project (No.192102210280), the Fundamental Research Funds for the Universities of Henan Province, Doctor Foundation of Henan Polytechnic University (No. B2017-36).

## REFERENCES

[1] J. Chen, H. Huang, S. Tian, Y. Qu, Feature selection for text classification with Naive Bayes, Expert Syst. Appl. 36 (2009), 5432–5435.
[2] S.B. Kim, H.C. Rim, S.H. Myaeng, K.S. Han, Some effective techniques for naive Bayes text classification, IEEE Trans. Knowl. Data Eng. 18 (2006), 1457–1466.
[3] M. Haddoud, A. Mokhtari, T. Lecroq, S. Abdeddaïm, Combining supervised term-weighting metrics for SVM text classification with extended term representation, Knowl. Inf. Syst. 49 (2016), 909–931.
[4] H. Kim, P. Howland, H. Park, Dimension reduction in text classification with support vector machines, J. Mach. Learn. Res. 6 (2005), 37–53.
[5] R. Li, Y. Hu, A density-based method for reducing the amount of training data in kNN text classification, J. Comput. Res. Dev. 41 (2004), 539–545.
[6] H. Wandabwa, D. Zhang, K. Sammy, Text categorization via attribute distance weighted k-nearest neighbor classification, in 2016 International Conference on Information Technology (ICIT), Bhubaneswar, India, 2016.
[7] Y. Wang, Z. Wang, Text categorization rule extraction based on fuzzy decision tree, Comput. App. 4 (2005), 1634–1637.
[8] S. Wang, C.D. Manning, Baselines and Bigrams: Simple, Good Sentiment and Topic Classification, Association for Computational Linguistics (ACL), Jeju Island, Korea, 2012.
[9] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, Comput. Sci. (2013). https://www.engineeringvillage.com/search/doc/abstract.url?SEARCHID=4982a786a73c4269a1e8ca8c9ee99349&DOCINDEX=1&database=1&pageType=quickSearch&searchtype=Quick&dedupResultCount=null&format=quickSearch&usageOrigin=recordpage&usageZone=detailedtab&toolsinScopus=Noload
[10] Y. Bengio, R. Ducharme, P. Vincent, C. Janvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (2003), 1137–1155.
[11] T. Mikolov, M. Karafiatm, S. Khudanpur, Recurrent Neural Network Based Language Model, International Speech Communication Association, Prague, Czech Republic, 2010.
[12] Y. Kim, Convolutional Neural Networks for Sentence Classification, Association for Computational Linguistics (ACL), Doha, Qatar, 2014.
[13] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, 2016.
[14] Y. Zhou, J. Xu, J. Cao, B. Xu, C. Li, B. Xu, Hybrid attention networks for Chinese short text classification, Comp. y Sist. 21 (2017), 759–769.
[15] Z. Wei, D. Miao, J.H. Chauchat, R. Zhao, W. Li, N-grams based feature selection and text representation for Chinese text classification, Int. J. Comput. Int. Sys. 2 (2009), 365–374.
[16] M. Post, S. Bergsma, Explicit and Implicit Syntactic Features for Text Classification, Association for Computational Linguistics (ACL), Sofia, Bulgaria, 2013.
[17] G. Gautam, D. Yadav, Sentiment Analysis of Twitter Data Using Machine Learning Approaches and Semantic Analysis, Institute of Electrical and Electronics Engineers Inc., Noida, India, 2014.
[18] Y. Song, Z. Wang, H. Wang, Short Text Conceptualization Using a Probabilistic Knowledgebase, International Joint Conferences on Artificial Intelligence, Barcelona, Spain, 2011.
[19] Z. Zhang, D. Miao, C. Gao, Short text classification using latent Dirichlet allocation, J. Comput. App. 33 (2013), 1587–1590.
[20] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science. 313 (2006), 504–507.
[21] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (2011), 2493–2537.
[22] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed Representations of Words and Phrases and their Compositionality, Neural Information Processing Systems Foundation, Lake Tahoe, NV, USA, 2013.
[23] N. Sotthisopha, P. Vateekul, Improving Short Text Classification Using Fast Semantic Expansion on Multichannel Convolutional Neural Network, Institute of Electrical and Electronics Engineers Inc., Busan, Korea, 2018.
[24] X. Zhang, J. Zhao, Y. Lecun, Character-Level Convolutional Networks for Text Classification, Neural Information Processing Systems Foundation, Montreal, Canada, 2015.
[25] J. Wang, Z. Wang, D. Zhang, J. Yan, Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification, International Joint Conferences on Artificial Intelligence, Melbourne, Australia, 2017.
[26] X. Wu, Y. Cai, Q. Li, J. Xu, H.F. Leung, Combining Contextual Information by Self-attention Mechanism in Convolutional Neural Networks for Text Classification, Springer Verlag, Dubai, UAE, 2018.
[27] J. Su, J. Zeng, D. Xiong, Y. Liu, M. Wang, J. Xie, A hierarchy-to-sequence attentional neural machine translation model, IEEE/ACM Trans. Audio Speech Language Process. 26 (2018), 623–632.

[28] Z. Yang, X. He, J. Gao, L. Deng, A. Smola, Stacked attention networks for image question answering, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2015.

[29] L. Gao, Z. Guo, H. Zhang, X. Xu, H.T. Shen, Video captioning with attention-based LSTM and semantic consistency, IEEE Trans. Multimedia. 19 (2017), 2045–2055.

[30] Z. Peng, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, B. Xu, Attention-based bidirectional long short-term memory networks for relation classification, in Meeting of the Association for Computational Linguistics, Berlin, Germany, 2016.

[31] D. Zhang, M. Hong, L. Zou, F. Han, F. He, Z. Tu, Y. Ren, Attention pooling-based bidirectional gated recurrent units model for sentimental classification, Int. J. Comput. Int. Sys. 12 (2019), 723–732.

[32] L. Wang, Z. Cao, G. de Melo, Z. Liu, Relation classification via multi-level attention CNNs, in Meeting of the Association for Computational Linguistics, Berlin, Germany, 2016.

[33] X. Qiao, C. Peng, Z. Liu, Y. Hu, Word-character attention model for Chinese text classification, Int. J. Mach. Learn. Cybern. 10 (2019), 3521–3537.