ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20190623122323
Original scientific papers

News Text Classification Based on an Improved Convolutional Neural Network

Wenjing TAO, Dan CHANG

Abstract: With the explosive growth in Internet news media and the disorganized status of news texts, this paper puts forward an automatic classification model for news based on a Convolutional Neural Network (CNN). In the model, Word2vec is firstly merged with Latent Dirichlet Allocation (LDA) to generate an effective text feature representation. Then when an attention mechanism is combined with the proposed model, higher attention probability values are given to key features to achieve an accurate judgment. The results show that the precision rate, the recall rate and the F1 value of the model in this paper reach 96.4%, 95.9% and 96.2% respectively, which indicates that the improved CNN, through a unique framework, can extract deep semantic features of the text and provide a strong support for establishing an efficient and accurate news text classification model.

Keywords: attention mechanism; Convolutional Neural Network (CNN); feature representation; text classification; Word2vec

1 INTRODUCTION

With the development of information technology, the spread of news has been rapidly improved. *Headlines Today, Sina Microblog* and other Internet news media platforms have a huge impact on people's daily life. Internet news can keep its readers updated because it is usually concise and extremely informational. The Chinese network newsreaders reached 660 million people in the first half of 2018, an increase of 160 million compared with the number at the end of 2017, those readers accounted for 82.7% of the total Internet users, as shown in Fig. 1 [1]. How to effectively and accurately classify the huge amount of news texts and extract valuable information is one of the most popular research subjects.

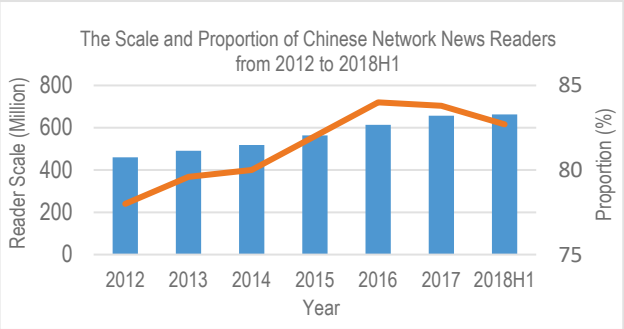

Figure 1 The situation of Chinese network news readers from 2012 to 2018H1

Text classification is one of the key links in information processing. The feature representation of text is the premise of text classification. A lack of semantic information about the text brings about unsatisfactory classification results. To overcome this problem, S. Banerjee and S. Cucerzan put forward a text classification method based on Wikipedia by expanding key concept sets. This method extracts category information from Wikipedia on potential concepts, then uses these concepts [2, 3] to expand the synonym features of the text, thus enriching the text features. Paolo Ferragin, et al adopted a graph model to classify all semantically similar concepts in Wikipedia. They used search engines to increase the feature information about the text and calculated the similarity between the search results and the text. However, this algorithm makes the number of nodes of the graph very large, which increases the time and space costs of the training [4]. Ning Yahui used high-frequency words as characteristic words in related fields, which is equivalent to conducting a short text classification based on introducing a professional background. In this way, the generalization ability of the classifier is improved [5]. In response to the challenge of categorization posed by a lack of semantic information in the text, Wang Sheng introduced the hyponymy relation of *HowNet* into the process of feature expansion [6]. The experimental results show that this method could improve the supplementary to textual semantic information and reduce the text space dimension. However, this method is not applicable to words that are not included in the background knowledge base. J. Hynek put forth a frequently used word set method based on Apriori to classify texts in the digital library [7]. All of the above methods improve the classification accuracy to some extent, but they all heavily rely on the external knowledge base or cannot automatically represent and extract features at the semantic level.

There is much related research on text classification algorithm, such as the Naive Bayes Model, the k-Nearest Neighbor, the Decision Tree and the Support Vector Machine. Substantially traditional machine learning is based on probability statistics, which requires constructing complex and inefficient feature projects [8]. As a result, the model has an unstable performance and poor robustness. Deep learning thought put forward by Hinton in 2006 gradually became the mainstream. As the numbers of network layer increase, more and more abstract data representation can be learned automatically, thus solving the problems of the limited expression capability of complex functions that the previous shallow structure algorithms possessed [9]. The Convolutional Neural Network (CNN) was applied to NLP tasks for the first time in 2008 by Collobert and realized the automatic feature extraction, which had significant efficiency in part-of-speech tagging, semantic analysis and identification of a named entity [10]. Wang P., et al. conducted sentiment classification to short Sina microblog text to improve the classification performance by using an information-extended convolutional neural network [11, 12]. Yao, et al. proposed a clinical text classification method that combines rule-based features and knowledge-guided deep learning techniques to capturing domain knowledge and

learning hidden features [13]. However, massive news text classification has been a challenge task for a long time, a single model could not present the best result, compared to a quantity of single algorithms, hybrid methods always have higher accuracy [14-17]. The attention mechanism has been popular in natural language processing in recent years. Rush A. M., et al. applied attention models to the abstract extraction of brief sentences [18]; Chorowski J., et al. introduced attention models in the field of speech recognition [19], and Luong M. T., et al. introduced attention models to machine translation [20]. Zhou P, et al. applied a two-way attention LSTM to relation detection and recognition [21]. Yang Z., et al. used two neural networks to model sentences and documents and introduced an attention mechanism to update the feature weight [22]. The above analysis showed that an attention mechanism could focus on key features that affect the model performance and the optimized model effect combined with the deep learning model.

As concerns the remainder of this paper, in Section 2, we describe the text feature fusion method based on Word2vec and LDA model. In Section 3, we describe improved text classification model based on CNN. In Section 4, we present experimental results and discussion. Finally, conclusions are summarized in section 5.

2 TEXT FEATURE FUSION METHOD

2.1 Word2vec Representation

Word2vc is a type of distributed representation method. The distributed hypothesis indicates that words appearing in the same context tend to have similar semantics. By calculating the cosine of two words, it is possible to determine whether there is a semantic correlation between them. The word2vec trained from Eq. (1) can obtain:

$$vec(king) - vec(man) \approx vec(queen) - vec(woman) \quad (1)$$

The above formula indicates that a linear combination of word2vec can usually produce a meaningful result and obtain semantic information between words. Word2vec has two training models, CBOW and SKIP-GRAM. The CBOW model predicts the central word based on the context window words, while the SKIP-GRAM model predicts the context window words based on the central word. The schematic diagram of the two types of models is shown in Fig. 2.

The Neural Network Language Model (NNLM) can evaluate the N-gram conditional probability through a neural network structure and predict the current word by using n-1 historical words. Word2vec is developed on the basis of the NNLM, which can be expressed by a distributed representation of words through large-scale corpus training. Fig. 3 shows the principle of the NNLM, which contains a three-layer structure. The first layer of the neural network language model is the input layer, represented by the green square on the bottom layer. The word2vec of each word is obtained from the matrix C by the index of the present words. The input layer is a $(n-1)d$ word2vec matrix. The number of word2vec is represented by $n-1$, and the dimensions of word2vec are represented by d. The second layer is a hidden layer, which is a normal neural network. *tanh* represents the activation function. The third layer is an output layer. The output layer consists of $|N|$ nodes, which means the size of the word list is $|N|$. The output is normalized by Softmax.

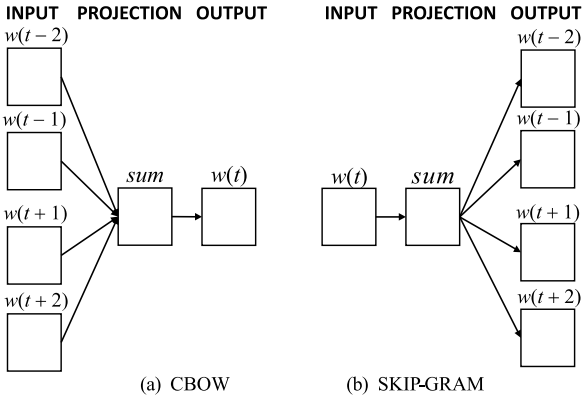

Figure 2 Schematic diagram of two types of word2vec models

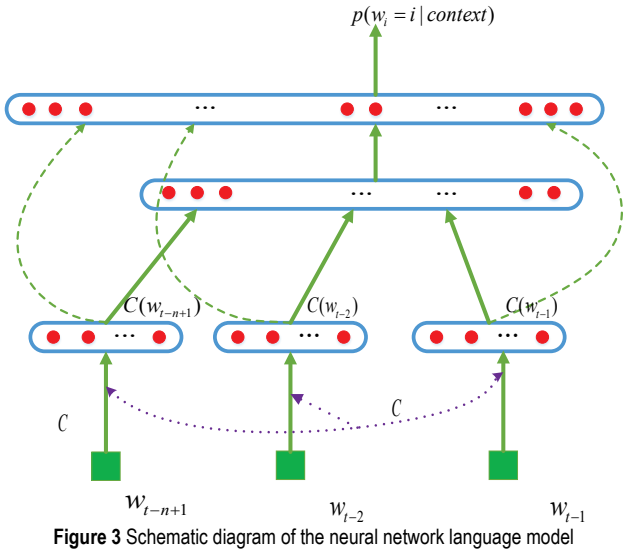

Figure 3 Schematic diagram of the neural network language model

C_{w_t} represents the word2vec representation of one of the words, and the output target is used to maximize the probability of the current word.

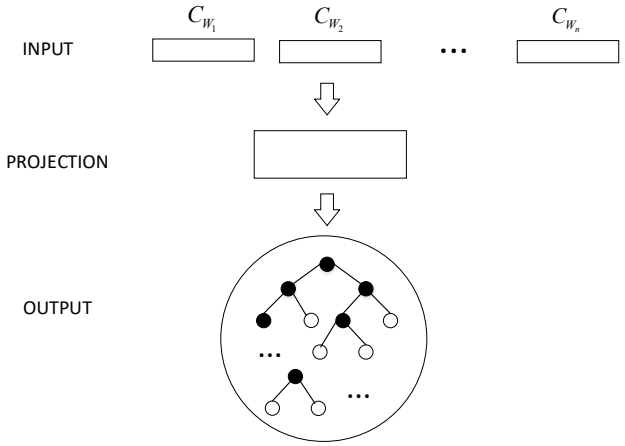

Figure 4 Schematic diagram of the structure of the CBOW model

In this paper, the CBOW model is used to train word2vec. A schematic diagram of the structure of the

CBOW model is shown in Fig. 4. On the basis of NNLM, the CBOW is simplified and improved as follows:

(1) It is changed from a three-layer structure to a two-layer structure, including an input layer and an output layer. In this way, the training speed is accelerated and has almost no impact on the training results.

(2) An additive average synthesis method is used as the input of the CBOW model instead of the vector mosaic method, by which the operation is simplified again, as shown from Eq. (2) that follows:

$$c = \left\{ w_{i-(n-1)/2}, w_{i-1}, w_{i+1}, w_{i+(n-1)/2} \right\} \tag{2}$$

The vector of the projection layer based on the accumulation of n vectors is expressed from Eq. (3):

$$x = \sum_{w_i \in c} \frac{c(w_i)}{n-1} \tag{3}$$

(3) Compared with the NNLM that just adopts (n−1) previous words of the w_j, the CBOW can adopt context words for which the window size is n to make predictions about current words. The training goal of the CBOW is to maximize the logarithmic probability. N is the total number of words in the entire corpus.

$$\begin{aligned} &p\left(w \mid context(w)\right) = \\ &= \sum_{i=1}^{N} \log p\left(w_{i-(n-1)/2}, w_{i-1}, w_{i+1}, ...w_{i+(n-1)/2}\right) \end{aligned} \tag{4}$$

(4) A massive corpus increases the training difficulty of the Softmax layer. Word2vec adopts the method constructing a hierarchical Huffman tree to accelerate the algorithm training so that the time complexity is reduced from $O(N)$ to $O(\log_2 N)$, and word2vec can easily be used in tasks related to natural language processing.

2.2 LDA Topic Modelling

Blei proposed the Latent Dirichlet Allocation (LDA) in 2003, which is an unsupervised text clustering method. Through training the corpus, the LDA topic model can mine the hidden semantics in the text and represent the conditional probability distribution of the feature words in the document. The theme is the central idea expressed by the text. The model believes that the co-occurrent words appearing in different documents are not completely isolated and that there is some semantic relevance. The probability of each word appearing in each document is calculated from Eq. (5):

$$\begin{aligned} &p(word \mid document) = \\ &= \sum_{topic} p(word \mid topic) \times p(topic \mid document) \end{aligned} \tag{5}$$

The method of dimensionality reduction used by the topic model is to model the words in the document and abstract them into the topic space and select K topics to represent the document with a certain probability. The theme model can deal with the problem of polysemy through the theme information of the text as a whole and can filter out the insignificant information of the theme and reduce the noise in the process of text categorization.

Sometimes two texts in the corpus are semantically correlated even though they have no common words. Word2vec may cause the loss of semantic information of the whole text. The topic model is a type of unsupervised approach that, through large-scale training, obtains topic semantic information implied in the text. The topic model is a three-layer Bayesian model generated completely with probability. The text-topic-word generation process in the LDA topic model is described in Fig. 5:

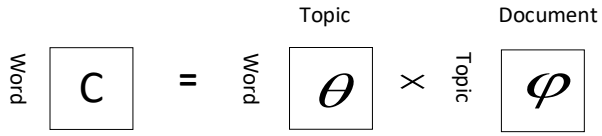

Figure 5 The generation process of the LDA topic model

(1) For any document, a topic is randomly selected from the polynomial conditional probability distribution multinomial (θ) of its corresponding topic distribution θ;

(2) For the selected topic z , randomly select a word w_n from the polynomial conditional probability distribution multinomial (φ) of its corresponding word distribution φ;

(3) Repeat this process until all the words N_m in the text have been traversed.

2.3 Text Feature Fusion Process

Word2vec is a type of continuous and dense real number vector. The degree of semantic correlation between words can be measured by the spatial distance between the vectors. The topic vector represents the probability distribution that each word belongs to a different topic. The LDA model can better represent the text and improve the text classification effect by combining with Word2vec. The feature fusion algorithm is shown as follows:

(1) Text vectorization. The corpus used in this paper is represented by C (*corpus*), and D represents the total number of documents from Eq. (6):

$$D = \left\{ d_1, d_2, ..., d_{|D|} \right\} \tag{6}$$

$|N|$ represents the vocabulary of the whole corpus with word2vec training, and the dimension of the word2vec is set as d. The matrix representation of the word2vec is expressed from the following Eq. (7):

$$C_{|N| \times d} = \begin{bmatrix} \vec{v}_{w_1} \\ \vec{v}_{w_2} \\ \cdots \\ \vec{v}_{w_n} \end{bmatrix} = \begin{bmatrix} v_{w_1,1} & v_{w_1,2} & \cdots & v_{w_1,d} \\ v_{w_2,1} & v_{w_2,2} & \cdots & v_{w_2,d} \\ \cdots & \cdots & \cdots & \cdots \\ v_{w_n,1} & v_{w_n,2} & \cdots & v_{w_n,d} \end{bmatrix} \tag{7}$$

w_i is one of the words; therefore, the word2vec of w_i is expressed from Eq. (8) that follows:

$$\vec{V}_{w_i} = (v_{w_i,1}, v_{w_i,2}, v_{w_i,3}, ..., v_{w_i,d}) \tag{8}$$

(2) Considering that words from different parts of speech contribute in different degrees to text classification, the feature selection is carried out in this paper based on the word frequency of parts of speech and *TF-IDF* to distinguish the importance degree of various contribution terms in text classification and to obtain the weighted word2vec.

The part of speech corresponding to word w_i is $pos(w_i)$, the part-of-speech weight of w_i is $W_{pos(w_i)}$, and the weighted word2vec of w_i is expressed from Eq. (9):

$$\vec{V}'_{w_i} = weighted_{w_i} = TF - IDF_{w_i} * W_{pos(w_i)} \odot \vec{V}_{w_i} \tag{9}$$

$\odot$ represents the dot product by elements, and the final matrix of the weighted word2vec is expressed from Eq. (10):

$$C'_{|N|\times d} = \begin{bmatrix} v'_{w_1 1} & v'_{w_1 2} & v'_{w_1 3} & \cdots & v'_{w_1 d} \\ v'_{w_2 1} & v'_{w_2 2} & v'_{w_2 3} & \cdots & v'_{w_2 d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v'_{w_n 1} & v'_{w_n 2} & v'_{w_n 3} & \cdots & v'_{w_n d} \end{bmatrix} \tag{10}$$

(3) The LDA topic model is trained to get the topic vector. The text-topic matrix and the topic-word matrix can be obtained after training the LDA topic model. T represents the number of topics, $|V|$ represents the vocabulary selected by the topic model, N represents the topic vector dimension, and quantity is the same as the word2vec dimension.

The text-topic matrix is expressed from Eq. (11):

$$M_{|D|\times|T|} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,|T|} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,|T|} \\ \vdots & \vdots & \vdots & \vdots \\ p_{|D|,1} & p_{|D|,2} & \cdots & p_{|D|,|T|} \end{bmatrix} \tag{11}$$

The topic-word matrix is expressed from Eq. (12)

$$N_{|T|\times|V|} = \begin{bmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,|V|} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,|V|} \\ \vdots & \vdots & \vdots & \vdots \\ q_{|T|,1} & q_{|T|,2} & \cdots & q_{|T|,|V|} \end{bmatrix} \tag{12}$$

(4) The process is represented based on the features of the combination of word2vec and LDA. Each column in the topic-word matrix can be seen as the probability distribution of each word on a different topic. The topic vector of any word w_i is expressed as V_{lda}, and the weighted word2vec is expressed as V_{w2a}. After splicing the word2vc and topic vector, the final vectorization of word w_i is represented from Eq. (13)

$$V(w_i) = [V_{w2v}; V_{lda}] \tag{13}$$

The word2vec and the LDA topic vector are combined to get richer semantic information from the text through optimizing the feature representation of the input layer of the CNN model.

3 IMPROVED TEXT CLASSIFICATON MODEL

3.1 Convolutional Neural Network

The convolutional neural network (CNN) is a feedforward network model structure based on artificial neural network. It has the advantages of local connection and weight sharing, which greatly reduces the number of parameters needed to learn in the network. Through multilayer nonlinear transformations, a CNN can learn the implicit features in large-scale text and is widely used in the fields of image analysis and speech recognition.

A CNN is usually composed of a convolution layer, a pooling layer, a fully connected layer and a Softmax classifier. The model structure diagram is shown in Fig. 6. The pooling layer and the convolution layer are unique to the CNN model. Multiple layers of convolution and pooling can be set to make the hierarchical structure of the CNN more complex and enhance the automatic learning and feature extraction capabilities of deep neural networks.

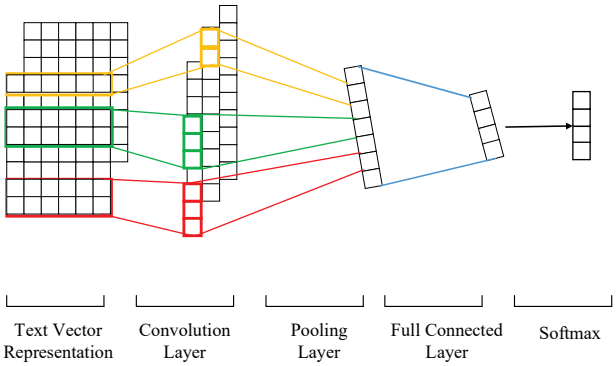

Figure 6 Structure diagram of a convolutional neural network

The leftmost part in Fig. 6 is the input layer, which is the feature representation matrix corresponding to the text. In this paper, we use the distributed word2vec of d dimension. Then, for n length of a text, it forms an $n \times d$ matrix.

The second part in Fig. 6 is the convolutional layer, which is used to extract local features of the text from a higher level. A convolution operation is performed in a sliding manner from top to bottom through an $l \times d$ convolution kernel, and a feature vector is obtained by the convolution operation. The column of the feature vector is 1, and the row is $(n + l - 1)$. The convolution operation extracts features of adjacent words of different lengths according to the window size l, and local information will be integrated into the overall information in the subsequent fully connected layer. The convolution operation process is shown in in Fig. 7. When convolution kernels of different sizes act on the matrix in the middle, they will convolute to different feature vectors.

The third part in Fig. 6 is the pooling layer, which plays the role of reducing overfitting and improving the training speed of the model. When the pooling function chooses the maximum value of the feature vectors in the region as the most important feature, it is the maximum pooling method,

as shown in Fig. 8. The average pooling method is used when the pooling function selects the average value of the feature vectors in the region. The only parameter of the pooling layer is the size of the pooling window.

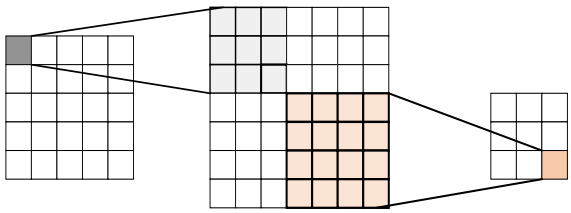

Figure 7 Schematic diagram of the convolution operation

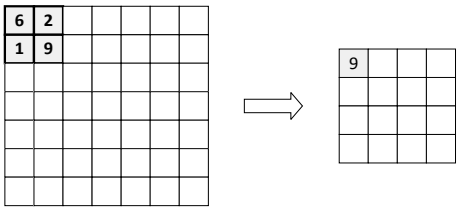

Figure 8 Schematic diagram of the max-pooling operation

After the multilayer convolution and the pooling operation, all the neurons are fully connected and classified by the Softmax, thus obtaining the category to which a text belongs.

3.2 Attention Mechanism

To obtain more detailed information about the target to which we need to pay attention, human beings can quickly screen out high-value information from a large amount of information with limited attention resources. An attention mechanism in deep learning is essentially similar to the visual attention mechanism in humans, and the core goal is to select more critical information for task processing.

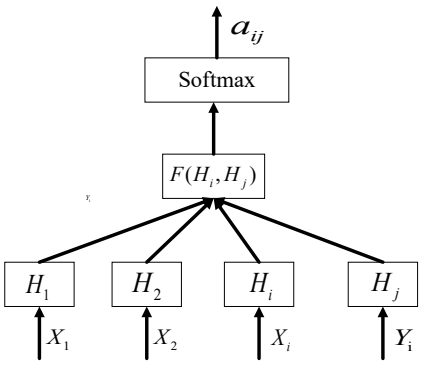

Figure 9 The Calculation Method of Attention Probability

An attention mechanism can be well combined with text categorization, focusing on the key features that affect the text categorization result, and dynamically allocating more attention weight to these features. Currently, most attention models are based on the Encoder-Decoder framework. The method in Fig. 9 is used to calculate the probability distribution of the attention mechanism. H_1, H_2, …, H_i are the hidden layer states of the Encoder layer, and H_j is the hidden layer state of the Decoder layer. For output y_i, function F is used to calculate the hidden layer state of the encoding stage and the current decoding stage and obtain the distribution probability of attention. The attention mechanism can intuitively explain the importance of features to the classification and is an effective strategy for dynamic learning in terms of the contribution of different features to a specific task.

3.3 Text Classification Model Based on CNN Framework

The news text classification model based on the CNN consists of an input layer, an attention layer, a convolution layer, a pooling layer, a full connection layer, and a Softmax classifier. The model framework diagram is shown in Fig. 10. A shallow semantic feature representation is performed in the input layer. The attention layer, the convolution layer, and the pooling layer are combined in a unified structure to form a three-layer feature extraction to generate an advanced feature representation that fuses the distributed probability of the attention mechanism, highlights the distinguishing effect of key characteristics, deepens the CNN network structure and learns the abstract features of the text features. Finally, the generated advanced features are subsampled and travel through the full connection layer to *Softmax*. In this way, the model can have an improved understanding of the text semantics. The word2vec and the LDA topic vector are combined to get richer semantic information from the text through optimizing the feature representation of the input layer of the CNN model.

To deal with news text classification tasks, this paper puts forward the CNN classification model integrated with attention mechanisms. The specific modeling steps are as follows:

(1) Data preprocessing

When using a CNN to classify text, the original text should be preprocessed first, primarily including text segmentation, part-of-speech tagging, removing stop words and filtering low-frequency words. After preprocessing, each text becomes a long sentence.

(2) Text feature representation

An effective feature representation method is used as the input of the CNN model, including the contextual semantic information of the words of the word2vec and the global semantic information of the topic model.

The sentence matrix $\boldsymbol{A} \in R^{n \times d}$ is established, and n means the length of sentence from Eq. (14).

$$\boldsymbol{A} = \left[x_1, x_2, ..., x_i, ..., x_n\right]^{\mathrm{T}} \tag{14}$$

$x_i \in R^d$ means the d-dimensional representation of the final word2vec in a sentence.

(3) The introduction of attention mechanisms

After the input layer, the attention layer is embedded and randomly initialized from Eq. (15):

$$R = (r_1, r_2, ..., r_n) \tag{15}$$

The attention mechanism is characterized by a dynamic distribution of weights. Feature information that is more relevant to a particular text category is captured, to distribute attention resources to more effective information and generate a new feature representation after weight.

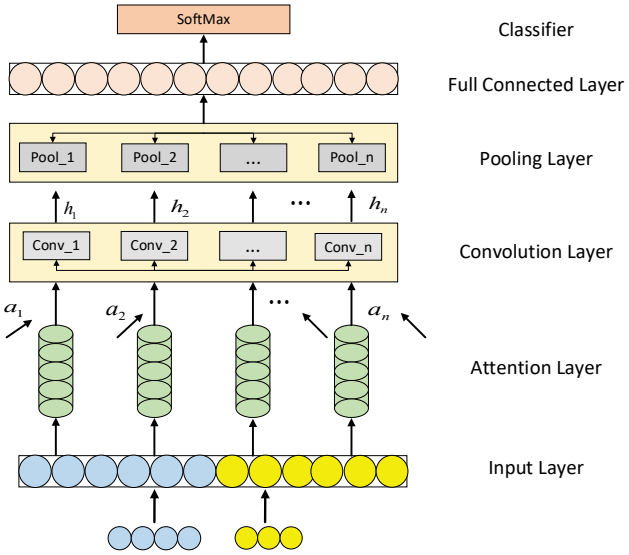

Figure 10 Framework of the news text classification model based on a CNN

a_{ij} means attention weight, which is calculated by a *Softmax* normalization. a_{ij} is defined from Eq. (16):

$$a_{ij} = \frac{\exp(score)}{\sum_{j=1}^{n} \exp(score)} \tag{16}$$

where $a_{ij} \geq 0$, and $\sum_{j=1}^{n} a_{i,j} = 1$.

The score function is used to measure the correlation coefficient of the features between the input layer and the attention layer. The function is defined from Eq. (17):

$$score = v \bullet \tanh\left(r_j \odot \sum_{i=1}^{n} x_i \right) \tag{17}$$

Where v represents the offset vector of the attention mechanism and $\odot$ represents the dot product operation of the matrix. The resulting attention weights are multiplied with the corresponding vector in A, and the new text features optimized by the attention mechanism are represented by x'. The formula is shown from Eq. (18). x' enters the convolutional layer and participates in the subsequent processing.

$$x' = \sum_{n} a_{ij} x_i \tag{18}$$

(4) Convolutional operation

$$h_i = f(W_1 \odot x'_{i:i+l-1} + b) \tag{19}$$

The formula for the convolution operation is shown from Eq. (19), h_i represents the result after the convolution operation, i.e., the dot product of the input matrix, and the filter adds the biasing. On this basis, the output of the convolution layer is obtained by the activation function. l represents the sliding window size, i.e., for a certain convolution kernel, the convolution operation is carried out on l words by sliding them up and down. $x'_{i:i+l-1}$ represents the local characteristic matrix of word i to word $i + l - 1$. W_1 means the filter, b represents the offset, and f means the Relu activation function.

Three sizes of convolution kernel are designed in this paper, i.e., 3×200, 4×200, and 5×200. The number of convolution kernels is set as 128. H is the output of the convolution layer, and it can be expressed from Eq. (20):

$$H = \left[h_1, h_2, ..., h_i, ..., h_{n-l+1} \right] \tag{20}$$

(5) Pooling operation

The maximum pooling operation is carried out on the features produced by the convolution kernel as shown from Eq. (21):

$$\hat{s} = \max[h_1, h_2, h_3, ..., h_{n-l+1}] \tag{21}$$

Where $h_i (i = 1, 2, \ldots, n-l+1)$ is the result of the after weight allocation by the attention mechanism and the convolution operation, $\hat{s}$ represents the result after the maximum pooling operation, by which the most important of the local features are obtained. The feature vector of 384×1 is obtained by splicing them together.

(6) Full connection and classification

A new vector is composed after the pooling operation, and it serves as the fully connected input; the probability of each category is obtained by using the Softmax function. Thus, all the local features extracted can be considered comprehensively, completing the task of news text classification and outputting the final classification result. The nonlinear transformation to feature representation $\hat{s}$ is carried out to predict the text category $\hat{y}$, and the calculation formula is shown from Eq. (22):

$$\hat{y} = \arg\max(W_2 \odot \hat{s} + b_1) \tag{22}$$

where $W_2 \in R^{C \times d}$ represents the weight matrix of the nonlinear transformation, $b_1 \in R^c$ represents the offset of the nonlinear transformation, and C represents the number of text categories. The function argmax calculates the text category label that results in the largest probability value.

The mini-batch gradient descent is used in this paper to train the CNN network. The algorithm constantly approximates the local minimum along the negative gradient direction of the loss function. Only a fixed number of samples need to be calculated for each iteration; further, the number is preset, which can improve the speed of training and avoids the problem of trapping in a local optimum. During the training process, the parameter θ is updated from Eq. (23):

$$\theta = \theta - \alpha \frac{\partial L}{\partial \theta} \tag{23}$$

where α means the learning rate, which is between 0 and 1. The parameter learning rate decreases with the increase in training iteration numbers.

Loss function is defined as follows:

$$L(y) = -\sum_{i}^{C} y_i \log p_i \tag{24}$$

where C is the number of possible categories for the output layer, y_i represents the true probability, and p_i is the predicted value of the probability of belonging to a certain category from Eq. (24). The cross-entropy loss function will reduce the rate of learning during the training of partial derivatives, which easily causes the gradient to disappear, and the model cannot converge. In addition, for a fully connected layer, regularized item L2 is used to constrain the fully connected parameters, and the regularized item is added to the loss function. Therefore, the final loss function consists of two parts, one is the cross-entropy, and the other is the regularized item L2. The loss function can be defined from Eq. (25):

$$Loss(\theta) = -\frac{1}{|N|}\sum_{x\in N}\sum_{c\in C} y_i \log p_i + \beta\|\theta\| \tag{25}$$

where N means the batch size; $||\theta||$ means the parameter for the full connection layer; and β means the coefficient of the regularized item. The concept of backpropagation is to transfer the loss function up layer by layer, then to solve the gradient of the loss function for each weight with the help of a gradient descent strategy and update the weight. Training of the neural network model is conducted until the loss function converges in a range.

Batch normalization is carried out on the parameters of each layer in the network after being trained once. That is, the data of each layer are standardized before input, as shown from Eq. (26):

$$\hat{\bar{X}} = X - \frac{\bar{X}}{S_X} \tag{26}$$

where $\hat{\bar{X}}$ represents the average of the sample data, and S_X represents the standard deviation of the sample data, which can improve the rate of convergence, eliminate the data distribution variations and increase the activation rate of each neuron. The idea of drop out is to randomly set some of the parameters as zero in the forward transmission of the network and randomly decrease the number of node activations. In this way, the interdependence of neurons can be effectively prevented, thus achieving the effect of preventing overfitting, and accelerating the training speed.

To evaluate the reliability of the training model and the scientificity of the experimental result, this paper adopts the method of 10-fold cross validation to evaluate the results.

4 THE EXPERIMENT RESULTS AND DISCUSSION

4.1 Experimental Preparation

The corpus used in this experiment is the THU Chinese news dataset, which has a variety of data contents and categories and is suitable for the text classification task of this paper. The link to the dataset is http://thuctc.thunlp.org. This paper uses 10 of these categories, and the selected text numbers of different categories are shown in Tab. 1.

Table 1 Category information of the dataset

No	Class label	Training samples	Testing samples
1	Technology	7200	800
2	Stock	7200	800
3	Sports	7200	800
4	Entertainment	7200	800
5	Politics	7200	800
6	Sociology	7200	800
7	Education	7200	800
8	Finance	7200	800
9	Furnish	7200	800
10	Game	7200	800

In evaluating the classification performance of the news text classification model in this paper, the precision rate, recall rate and F1 value are used as evaluation indexes, and their definitions are as follows:

The precision rate is defined from Eq. (27):

$$P = \frac{TP}{TP + FP} \tag{27}$$

The recall rate is defined from Eq. (28):

$$R = \frac{TP}{TP + FN} \tag{28}$$

The F1 value is defined from Eq. (29):

$$F_1 = \frac{2PR}{P + R} \tag{29}$$

Among these indexes, the precision rate inspects the correctness of the classification results, and the recall rate inspects the completeness of the classification results [18]. The F1 value includes both the precision rate and the recall rate, which is the harmonic average of the two indexes and should be paid more attention in classification.

4.2 Experimental Results

In this paper, experiments are carried out using the neural network library Keras that is based on Python language. The third-party Python open source toolkit Gensim is used to train word2vec. The algorithm classification process is shown in Fig. 11.

Training word2vec is the cornerstone of the news text classification task. To determine the best dimension of word2vec, we performed experiments where we set a 50-dimensional, 100-dimensional, 200-dimensional and 300-dimensional word2vec for result verification on the THU corpus. The results in Fig. 12 show that when the dimension of the word2vec is selected between 50 and 200 dimensions, the F1 value can be improved. When the dimension of the word2vec is selected between 200 and 300 dimensions, the F1 value tends to be stable and does not dramatically change. Finally, a 200-dimensional word2vec is selected as the input of the model and the basis for subsequent tasks.

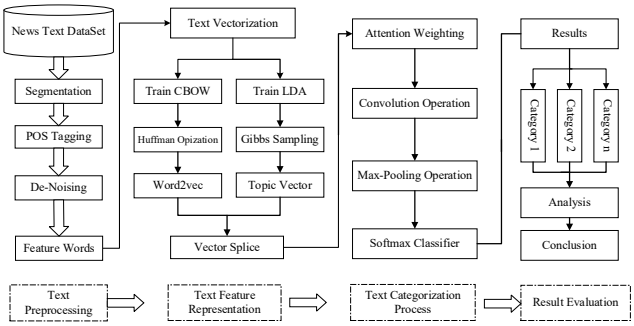

Figure 11 Schematic diagram of proposed text classification model

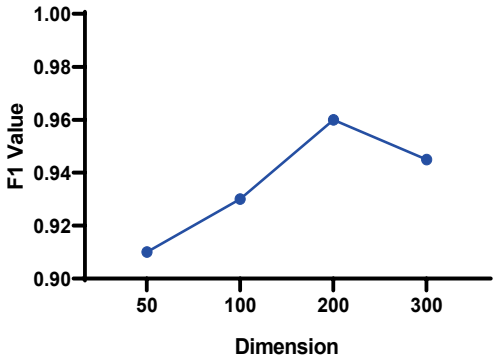

Figure 12 Changes in the F1 value with different dimensions

Some words are randomly selected, and the top 10 words most relevant to their semantics are found through word2vec. The cosine distance is used to calculate the degree of correlation, and the closer the cosine distance is to 1, the higher the semantic correlation is. Take the word "休☒(leisure)" as an example. The trained effect of the word2vec model is shown in Tab. 2. It can be found that these words are indeed related to 休☒(leisure), which shows that the word2vec training is sufficient and can rise to the semantic level in expressing the features of the text. From the results, the trained word2vec model meets the actual expectations.

Table 2 The 10 words with the highest similarity to "休☒" in corpus

Word	Cosine value
☒放(Release)	0.78360198
愉悦(Pleasure)	0.73041021
身心(Mind and Body)	0.68989992
慢下来(Slow Down)	0.68913924
舒☒(Relief)	0.68006539
散心(Distraction)	0.67704129
好山好水(Ideal Landscape)	0.67617172
舒☒(Briskness)	0.67607120
星座(Constellation)	0.67098677
空☒(Free)	0.66703512
悠☒(Carefree)	0.66632254

The dimension of word2vec is set to 200, and the window size is set to 10. The parameter settings of the CNN-based news text classification model are shown in Tab. 3.

The text classification effect of the model proposed in this paper is shown in Tab. 4. The overall classification situation of the model is above 95%, and the results of the precision rate, recall rate, and F1 value are 96.4%, 95.9%, and 96.2%, respectively. Generally, the results of the model are in line with expectations. The results from the four categories of Sociology, Sports, Education, and Finance are slightly poor, which may be caused by insufficient training. A larger amount of data will be used to improve the effectiveness of the model.

Table 3 Parameter settings of the CNN model

Model parameters	Parameter setting
Size of convolution kernel	3×200, 4×200, 5×200
Number of convolution kernel	128
Drop out	0.5
Learning rate	1e-3
Number of categories	10
Batch size	64
Neuron amount in the full connected layer	128

Table 4 Results of news text classification model-based CNN

Category	Precision	Recall	F1
Technology	0.989	0.986	0.983
Stock	0.986	0.984	0.978
Sports	0.949	0.937	0.944
Entertainment	0.937	0.948	0.953
Politics	0.992	0.988	0.991
Sociology	0.929	0.937	0.948
Education	0.938	0.932	0.935
Finance	0.948	0.942	0.934
Furnish	0.985	0.974	0.981
Game	0.987	0.962	0.973
Total	0.964	0.959	0.962

4.3 Discussion

In this paper, text features are primarily expressed based on the word2vec and the LDA model. Considering the contextual semantic information of the word and the overall topic information, the method of feature representation is compared with the one-hot vector, the single word2vec and the glove. The classification results obtained by these four text representation models, when applied to the classification method in this paper, are compared (W2V+LDA+ATCNN), as shown in Fig. 13.

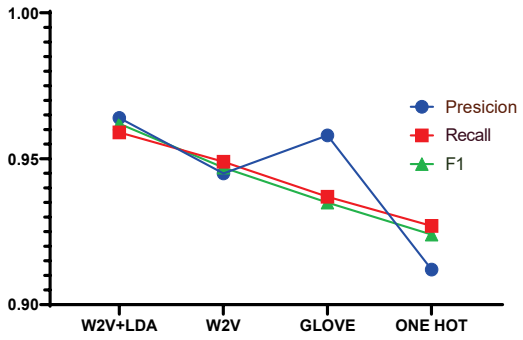

Figure 13 Results of different news text representation models

Fig. 13 shows that word2vec fusion LDA feature representation method has achieved good results in the precision rate, the recall rate, and the F1 value, which use the dynamic word2vec training method. Word2vec will be used as a parameter in the training process of subsequent models and will be continuously adjusted and optimized. This method can more comprehensively represent news text features and overcome the problems of sparse news

text features and semantic information, thus providing a more ideal feature representation form for subsequent models than what was previously available.

The one-hot method obtained a relatively poor result because it ignores the order and semantics between words and cannot describe the feature distribution of the original input data at a more abstract level. The classification model that only uses word2vec for text representation does not fully consider the overall semantic information of the text. Glove comprehensively utilizes the global co-occurrence information, obtained a better precision rate and increased the speed for training the model. However, this model does not consider the word order relationship, and the overall semantic acquisition is not as good as the vector mosaic method of feature representation based on word2vec and LDA proposed in this paper.

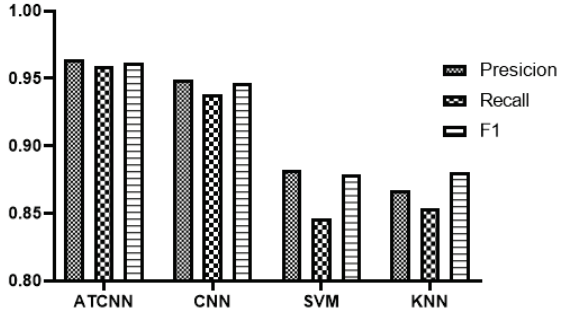

Figure 14 Results of different text classification models

SVM and KNN are classifiers that achieve good results in classification tasks in machine learning algorithms, and this paper chooses to compare its results with these two traditional machine learning algorithms. The results in Fig. 14 show that the deep learning model (W2V+LDA+ATCNN) can achieve better results than the traditional machine learning model in news text classification tasks. This is because it is composed of a multilayer neural network structure and can automatically learn more important semantic features for classification through a large amount of training data. After the attention mechanism is introduced, the precision rate, recall rate, and F1 value are increased by 1.4%, 0.8%, and 1.9%, respectively. This shows that the combination of the attention mechanism and the CNN is completely feasible, can make the model pay attention to more important features and can improve the effect of the news text classification model.

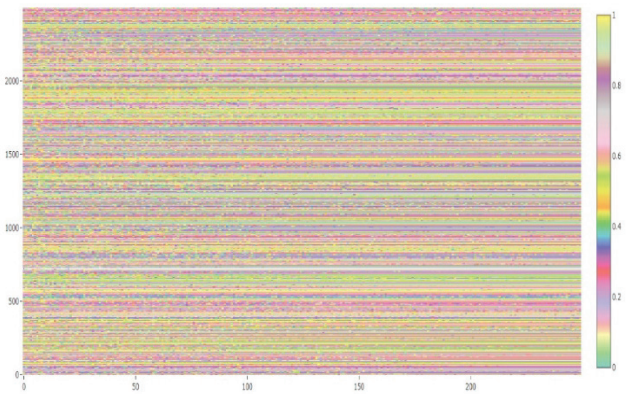

Figure 15 The visualization of the attention mechanism

Fig. 15 shows more intuitively the effect of different features on the classification model after the attention mechanism is introduced according to different colors. The attention model calculates the probability distribution value of the attention assigned to each feature, and different colors represent different probability values of the attention assigned, which can effectively extract features in the text, enhance the model's judgment on key features, and illustrate that the introduction of the attention mechanism plays a certain role in improving the classification performance of news texts. Attention mechanisms that have achieved excellent results in the field of machine translation can also be applied to the field of text classification.

The experimental results show that the model this paper proposed can capture deep semantic features of the text. Compared with SVM, KNN and other algorithms, this method has the advantages of good learning ability and powerful generalization ability and applicability to complex news text classification tasks.

5 CONCLUSION

In this paper, we propose a kind of hybrid text classification model based on CNN. First, characterize the text. Word2vec, which integrates topic information, is used as the basis of the text feature representation. Then, combined with the attention mechanism, better precision rates, recall rates, and F1 values are indeed obtained. Finally, experimental results have demonstrated its effectiveness of text classification in Chinese network news.

6 REFERENCES

[1] CNNIC. (2019). The 43rd China Statistical Report on Internet Development. *Netcom Civil-Military Int Egration on Cyberspace, (02)*: 37-38. http://kns.cnki.net/kns/brief/default_result.aspx

[2] Banerjee, S., Ramanathan, K., & Gupta, A. (2007). Clustering short texts using Wikipedia. *Proceedings of the 30th annual international ACM Sigir conference on research and development in information retrieval*, Amsterdam, The Netherlands. https://doi.org/10.1145/1277741.1277909

[3] Cucerzan, S. (2007). Large-scale Named Entity Disambiguation based on Wikipedia Data. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Czech Republic. https://www.aclweb.org/anthology/D07-1074

[4] Ferragina, P. & Scaiella, U. (2010). TAGME: on-the-fly annotation of short text fragments (by Wikipedia entities). *Proceedings of the 19th ACM Conference on Information and Knowledge Management*, Toronto, Canada. https://doi.org/10.1145/1871437.1871689

[5] Ning, Y. H., Fan, X. H., & Wu, Y. (2009). Short Text Classification Based on Domain Word Ontology. *Computer Science, 36*(03), 142-145.

[6] Wang, S. & Fan, X. H. (2010). Chinese Short Text Categorization Using Hyponymy. *Computer Applications, 30*(3), 603-606. https://doi.org/10.3724/SP.J.1087.2010.00603

[7] Hynek, J., Jezek, K., & Rohlik, O. (2000). Short Document Categorization-Item Sets Method. In *4th European Conference on Principles and Practice of Knowledge Discovery in Databases, Workshop Machine Learning and*

Textual Information Access, Lyon, France. http://eric.univlyon2.fr/~pkdd2000/Download/WS4_02.pdf

[8] Cai, H. P., Wang, L. D, & Duan, S. H. (2016). Sentiment Classification Model Based on Word Embedding and CNN. *Computer Applied Research, 33*(10), 2902-2905+2909. http://www.cnki.com.cn/Article/CJFDTotal-JSYJ201610006.htm

[9] Leswei, Y. & Zhao, J. (2013). Exploration of Chinese Word Segmentation Algorithm based on Representation Learning. *Chinese Journal of Information Science, 27(*05), 8-14. http://www.cnki.com.cn/Article/CJFDTOTALMESS201305002.htm

[10] Collobert, R. & Weston, J. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. *Proceedings of the 25th International Conference on Machine Learning*, New York, USA. https://doi.org/10.1145/1390156.1390177

[11] Wang, P., Xu, B., & Xu, J. (2016). Semantic Expansion Using Word Embedding Clustering and Convolutional Neural Network for Improving Short Text Classification. *Neurocomputing, 174*(PB), 806-814. https://doi.org/10.1016/j.neucom.2015.09.096

[12] Shi, Y., Tang, Y. R., & Long, W. (2018). A Text Mining Based Study of Investor Sentiment and Its Influence on Stock Returns. *Economic Computation and Economic Cybernetics Studies and Research*, 52. 183-199. https:/doi.org/10.24818/18423264/52.1.18.11

[13] Yao, L., Mao, C., & Luo, Y. (2018). Clinical Text Classification with Rule-based Features and Knowledge-guided Convolutional Neural Networks. *2018 IEEE International Conference on Healthcare Informatics Workshop (ICHI-W)*. IEEE. https://doi.org/10.1109/ICHI-W.2018.00024

[14] Zhao, P. X., Gao, W. Q., Han, X., & Luo, W. H. (2019). Bi-objective collaborative scheduling optimization of airport ferry vehicle and tractor. *International Journal of Simulation Modelling, 18*(2), 355-365. https://doi.org/10.2507/IJSIMM18(2)CO9

[15] Tang, M., Gong, D., Liu, S., & Lu, X. (2017). Finding Key Factors for Electric Vehicle Charging Station Location: A Simulation and ANOVA Approach. *International Journal of Simulation Modelling, 16*(3). https://doi.org/10.2507/IJSIMM16(3)CO15

[16] Wang, J. J. & Que, D. F. (2018). An Experimental Investigation of Two Hybrid Frameworks for Stock Index Prediction Using Neural Network and Support Vector Regression. *Economic Computation and Economic Cybernetics Studies and Research*, 52, 193-210. https://doi.org/10.24818/18423264/52.4.18.13

[17] Masoud, R., Zahedifard, S., & Rezaie-Malek, M. (2018). An Integrated Multi-Criteria Decision-Making Approach for Portfolio Problem in Energy Service Companies under Uncertainty. *Economic Computation and Economic Cybernetics Studies and Research*, 52, 305-322. https://doi.org/10.24818/18423264/52.4.18.20

[18] Rush, A. M., Chopra, S., & Weston, J. A. (2015). Neural Attention Model for Abstractive Sentence Summarization. *Computer Science.* https://doi.org/10.18653/v1/D15-1044

[19] Chorowski, J., Bahdanau, D., & Serdyuk, D. (2015). Attention-Based Models for Speech Recognition. *Computer Science*. https://arxiv.org/abs/1506.07503

[20] Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. *Computer Science*. https://doi.org/10.18653/v1/D15-1166

[21] Zhou, P., Shi, W., & Tian, J. (2016). Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. *Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/P16-2034

[22] Yang, Z., Yang, D., & Dyer, C. (2016). Hierarchical Attention Networks for Document Classification. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. https://doi.org/10.18653/v1/N16-1174

Contact information:

Wenjing TAO, Master
(Corresponding author)
Beijing Jiaotong University,
No. 3, Shangyuan Village, Haidian District, Beijing, China
16120618@bjtu.edu.cn

Dan CHANG, Professor
Beijing Jiaotong University,
No. 3, Shangyuan Village, Haidian District, Beijing, China
changdan@bjtu.edu.cn