CNN-based Skip-Gram Method for Improving Classification Accuracy of Chinese Text

Wenhua Xu, Hao Huang, Jie Zhang, Hao Gu, Jie Yang, and Guan Gui*

College of Telecommunications and Information Engineering, Nanjing University of Posts and
Telecommunications, Nanjing 210003, China.
* Corresponding author: Guan Gui (guiguan@njupt.edu.cn)

Abstract

Text classification is one of the fundamental techniques in natural language processing. Numerous studies are based on text classification, such as news subject classification, question answering system classification, and movie review classification. Traditional text classification methods are used to extract features and then classify them. However, traditional methods are too complex to operate, and their accuracy is not sufficiently high. Recently, convolutional neural network (CNN) based one-hot method has been proposed in text classification to solve this problem. In this paper, we propose an improved method using CNN based skip-gram method for Chinese text classification and it conducts in Sogou news corpus. Experimental results indicate that CNN with the skip-gram model performs more efficiently than CNN-based one-hot method.

1. Introduction

Text classification has always been one of the most basic techniques in natural language processing (NLP). In news classification, the algorithm based on the text classification processes news to determine which type it belongs to, such as finance, sports, military, and entertainment, among others. News classification can help readers quickly search or locate interest news [1]. In sentiment analysis field, classification is conducted using two types (positive, negative) or multiple categories (angry, happy, sad, etc.) for film reviews, product evaluations, and service evaluations to facilitate users in their selection process [2]. When processing messages, text classification can be used to quickly identify spam or harassment for interception, protecting users from a massive amount of useless information [3]. As part of other NLP systems, such as in question answering systems, text classification can be used to distinguish questions [4]. A high-accuracy classification model needs to be developed.

In recent years, the development of deep learning makes significant progress in various fields, such as the intelligent wireless communications [5]–[12], artificial intelligence and the internet of things [13]–[16], and the computer vision [17]–[21]. With the success of neural networks in deep learning, the efficiency of convolutional neural networks (CNNs) in extracting data features has been recognized [22]. This accomplishment has encouraged a number of researchers locally and globally to apply deep learning in NLP and achieved good results. In many aspects of NLP, deep neural networks exceed the performance of many traditional machine learning methods, including text classification. The effect of CNNs on text classification has considerably surpassed traditional machine learning. Compared with traditional techniques, CNNs are more suitable for processing high-dimensional data. Their distinct convolution and pooling structure can extract text features, which has been confirmed by early studies [23].

The classification model depends on the quality of inputs. A basic issue in NLP is the representation of text [24]. Feature extraction from text is related to the accuracy of the classification model. Owing to the inability of a computer to process text directly, text needs to be represented as text vectors, and then its features can be extracted. In traditional classification, features have to be extracted and then classified, so the text representation is important. In deep neural networks, the accuracy of the classification model is also related to the quality of text representation, because CNNs extract feature information from the text by the input text vectors and then train the model [25]. Word vectors, which can be summarized as one-hot vectors and distributed vectors. Distributed vectors can be represented using various techniques. The present study uses the skip-gram model to train distributed vectors. In a big data environment, distributed vectors can reflect lexical-level information compared with one-hot vectors. For example, the distance of words in the same category in a vector space would be close.

Chinese words present a challenge because Chinese words have no space between them, rendering Chinese text classification more complex than other text. Accordingly, this paper mainly studies Chinese text classification. This study uses a classification method based on CNNs combined with the skip-gram model to classify long Chinese texts. CNNs have performed efficiently in text classification and they are better than traditional machine learning methods, such as linear regression classification, Bayesian classification, and decision tree classification, among others [23]. Compared with one-hot vectors, distributed word vectors can reflect the relationship between words [24]. This paper uses CNNs with the

skip-gram model to train distributed word vector. The performance of the classification model can be enhanced by combining CNNs with the skip-gram model. The experimental results show that the combination of CNNs and the skip-gram model can improve the accuracy of Chinese text classification.

  This paper is organized as follows: Section 2 introduces the word vectors and the development of CNNs in deep learning for text classification. It introduces the skip-gram model and the CNNs model in Section 3. Section 4 provides the experimental results. Conclusion is given in Section 5.

2. Related Work

Bengio *et al*. proposed a neural probabilistic language model in 2003 [26]. The current study is the first paper to propose word representation. The skip-gram model provides an effective way to learn high-quality distribution vector representations, which can capture a large number of precise syntactic and semantic relationships [24]. In addition, paper [24] also shows that good vector representations of millions of phrases can be learned. In the present study, distributed word vectors are trained using the skip-gram model. The most popular traditional classification techniques include Naive Bayes, support vector machines, decision trees, and linear regression classification. Compared with traditional classification techniques, deep neural networks have recently become the mainstream of classification, which is first manifested in image classification. The first application of CNNs is image classification [22]. Krizhevsky *et al*. created large-scale deep neural networks and won the 2012 ImageNet Large-Scale Visual Recognition Challenge. The use of CNNs in NLP was first introduced by Yoon Kim who applied CNNs to text classification and conducted comparative experiments on data sets to demonstrate the effectiveness of CNNs in text classification tasks [23]. He also proved that CNN was highly significant for many NLP tasks.
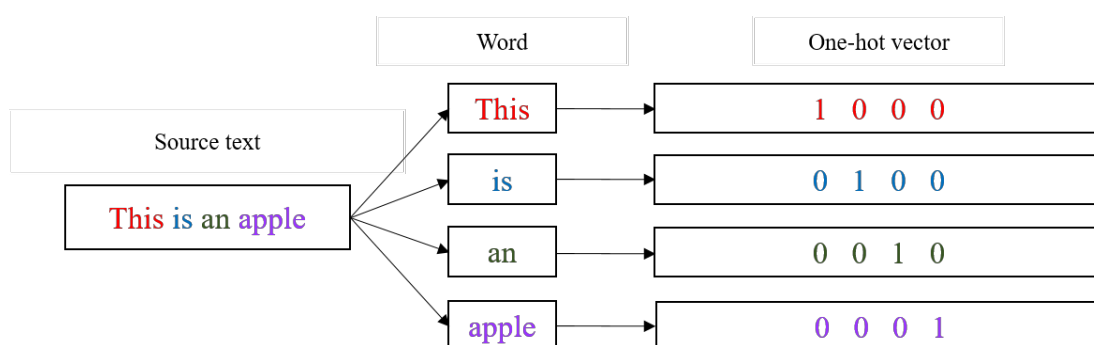

Fig. 1. Using the one-hot method to represent word vectors.

3. Text Classification Model

3.1 Word Representation Model

Words cannot be typed directly into the neural networks; thus, they need to be encoded, and their vectors can input into the neural networks. The word vector mainly includes two representation techniques: the one-hot representation and the distributed representation.

3.1.1 One-hot Representation

All word vectors in the one-hot method are represented by 1 and 0. This one-hot representation method involves a relatively simple process. First, build a vocabulary containing *V* words in the text. The *V* words are fixed sequentially, and each word can be represented by a *V*-dimensional sparse vector. Then, encode the corresponding position of the word is 1, and other elements are 0. Traditional sparse representations denote words that can cause dimensional explosion when solving certain tasks (such as building a language model) [27]. The shortcomings of one-hot vectors have led to the emergence of distributed word vectors.

3.1.2 Distributed Representation

Distributed word vectors use multiple elements to represent words, and these elements can exist in arbitrary numbers. The dimension of the vectors can be determined beforehand. Generally, vectors can be 50-dimensional or 100-dimensional. The greatest advantage of distributed word vectors is that they can help the classification model to learn text features. The distances between vectors represent their relationship, helping models understand the relationship between words [27]. Two training methods exist for distributed word vectors. One is the continuous bag-of-words model, which uses the context to predict the central word; the other is the skip-gram model, which uses the central word to predict the context. This study uses the skip-gram model.

The training process of the skip-gram model is as follows. The first step is to build word pairs, which is presented in **Fig. 2**. Word pairs are created by a combination of adjacent words. The word pairs indicate that words associated with the same subject will appear together. For instance, football and basketball must have a higher probability than football and mobile phones to appearing together.

Then, the second step is to train the networks. The process of getting the word vectors is training the hidden layer matrix of network step-by-step [28]. The input is the first word in the word pairs, and the output is the second word in the word pairs. In training these networks on word pairs, the inputs are one-hot representations of the first words, and the training outputs are one-hot representations of the words. The purpose of the training is to make the input approach the output after passing through the hidden layer matrix. When input words are used to evaluate a trained network, the output vector becomes a probability distribution. Training a simple neural network only involves learning the weight of the hidden layer, which is the real objective of learning. The mapping function between the input and output is shown in formula (1), where Y is the output, X is the input, matrix A represents the text vectors, and $\boldsymbol{f}_1$ is the activation function.

$$Y = \boldsymbol{f}_1(A) \cdot X \tag{1}$$

The mapping process is provided in **Fig. 3**. The input and the output are both the collections of one-hot word vectors, and they are separated from the word pairs. The word pairs indicate that the paired words in the same group appear together with a high probability, which means that these words are close. In order to find the needed word vector matrix, the input and the output are utilized.
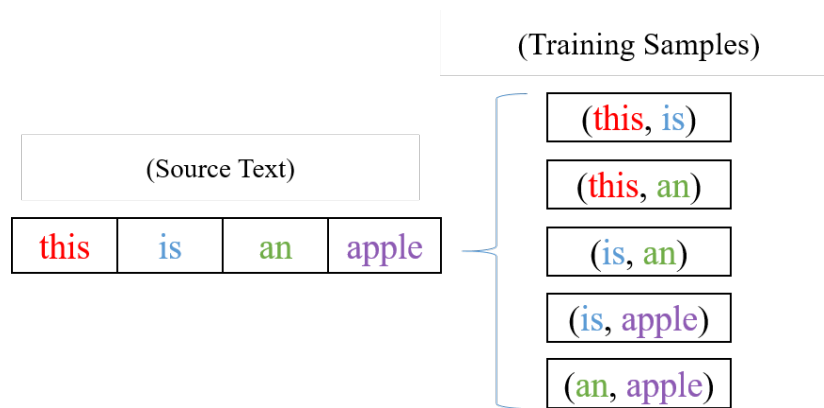
Fig. 2. Setting the sliding window to one to create word pairs for the training word vector.
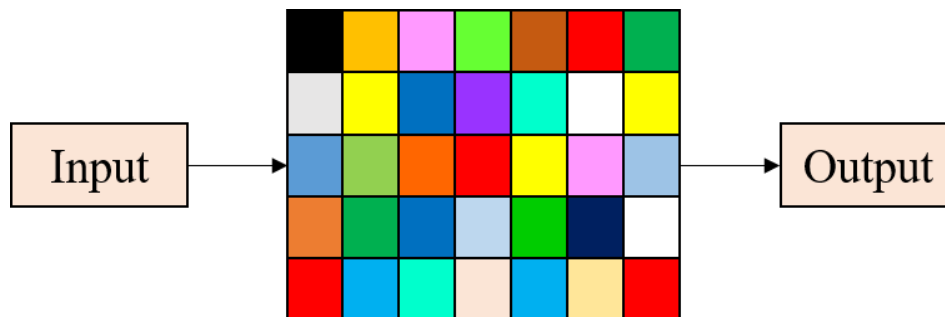

Fig. 3. The hidden layer matrix in the middle is the word vector matrix needed.

3.2 CNNs Model

After the word vectors are trained, the text is directly represented as input to CNNs by the trained word vectors. The whole model is provided in **Fig. 4**.

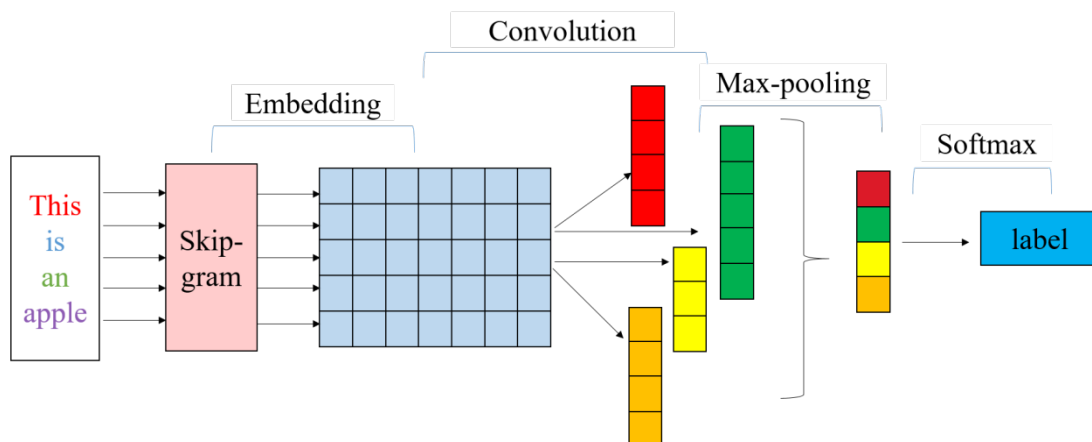

Fig. 4. This illustration is a model flow chart for text classification of the entire CNNs.
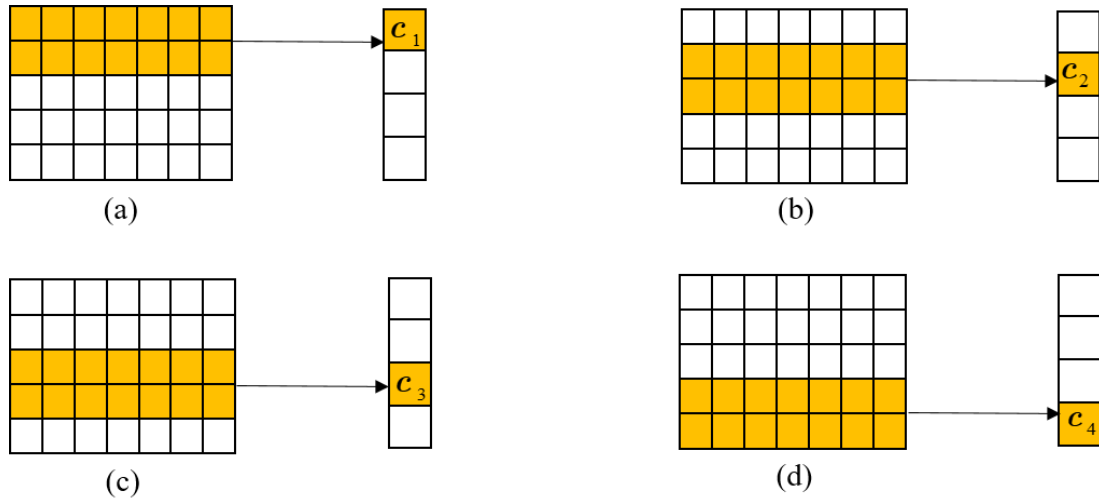
Fig. 5. (a)–(d) are the first, second, third, and fourth steps of the convolution. The convolution step is one. The width of the convolution kernel is equal to the dimension of the word vector.

The first layer is the word embedding layer. After the word vector training is completed, the distributed word vector matrix is used as the input of the convolution layer.

The second layer is the convolution layer. The text convolution varies from the image convolution, which operates from left to right and top to bottom [22]. The text convolution sets the width of the convolution kernel equal to the length of the word vector. The kernel can only move up and down. This movement does not separate the word vectors and can ensure that the complete word vector is scanned each time. The operator of convolution is described as formula (2). The convolution process can be seen in **Fig. 5**:

$$c_i = f_2(W \cdot X + b) \tag{2}$$

where W is the convolution matrix, c_i is the convolution output, X is the text vector, and f_2 is the activation function. Only one convolutional layer is used here. In the Section 4, models with one convolutional layer and two convolutional layers are used for comparison.
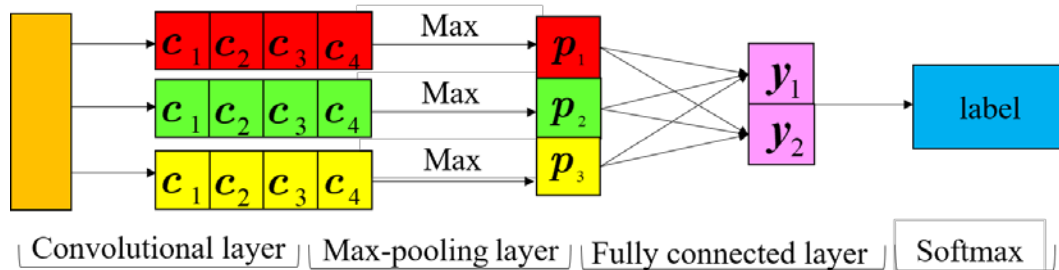

Fig. 6. Convolution layer, max-pooling layer, and fully connected layer with the softmax classifier of CNNs.

Different convolution kernels convolve the text matrices to obtain different feature outputs. The max-pooling layer simply combines the largest values from the previous one-dimensional feature map and combines them into a new vector because the maximum value represents the most important signal. Another advantage is that if the sentences have not been filled previously, the length of the sentences will be different, resulting in different vector dimensions after convolution. After max-pooling, the difference in length between sentences can be eliminated.

The last layer is a fully connected layer followed by a softmax function; the max-pooling layer is connected to the softmax function via the fully connected layer. The softmax function reflects the probability distribution of the final category, and the output is the category of the text.

$$C = (c_1, c_2 \ldots c_n) \tag{3}$$

$$P_i = \max(C) \tag{4}$$

$$L(y)_j = \frac{e^{y_j}}{\sum_{i=1}^{n} e^{y_i}} \tag{5}$$

where P_i and y_i denote the output of the max-pooling layer and the fully connected layer, respectively. Formula (5) denotes the operator of the softmax function, which can calculate probability of a text sample in order to distinguish its true category.
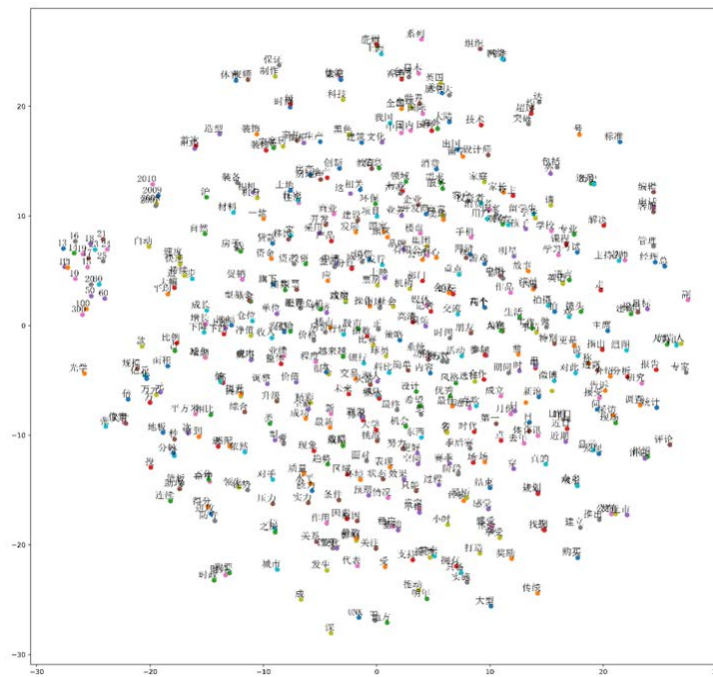

Fig. 7. These are word vectors that trained by the skip-gram model.

4. Experimental Classification Results and Analysis

4.1 Data Set

The data set is the publicly available Sogou news corpus. It is a Chinese news corpus on the Internet. The corpus contains 10 categories, each containing 10,000 training news, 1,000 validation news, and 500 test news.

4.2 Distributed Representation
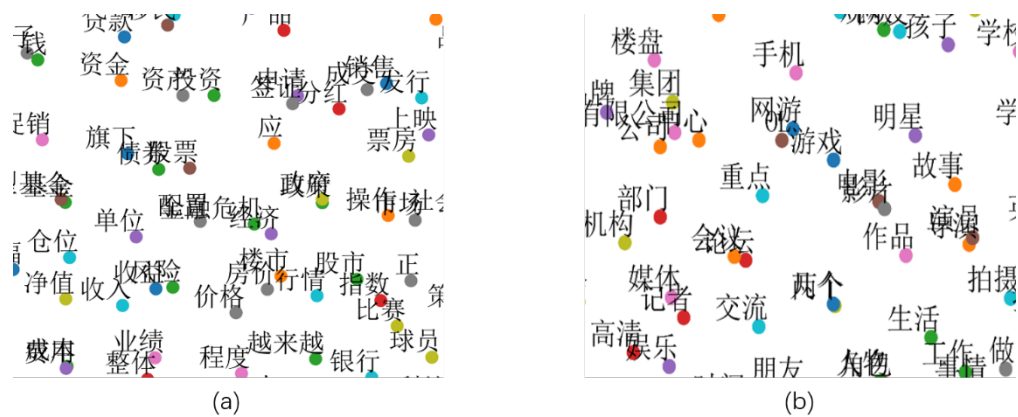
Fig. 8. Parts of **Fig. 7.** (a) Chinese words related to investment. (b) Chinese words related to entertainment (These words are in the training corpus).

The data set is first used to train the 128-dimensional word vectors. After training, the word vector is projected onto a two-dimensional coordinate system for observation. Some training results of word vectors are shown in **Fig. 7**. The word vectors in **Fig. 7** are too many to be observed. For a more intuitive observation, **Fig. 8** is extracted from **Fig. 7**, and the words with the same theme are relatively close.

4.3 Comparison of the One-hot Vector and Distributed Vector with CNNs

This section mainly compares the skip-gram model with the one-hot model. To ensure the rigor of the conclusion, this study separates the section into two parts. One is the case of the original Sogou news corpus, and the other is the case of one-fifth size of the original data. The original corpus contains ten categories, and each category contains over 10000 pieces of news. It is enough to train the proposed model using this large corpus. In contrast, the corresponding small corpus containing over 2000 pieces of news are tested on the same model.

4.3.1 The Big Data Case
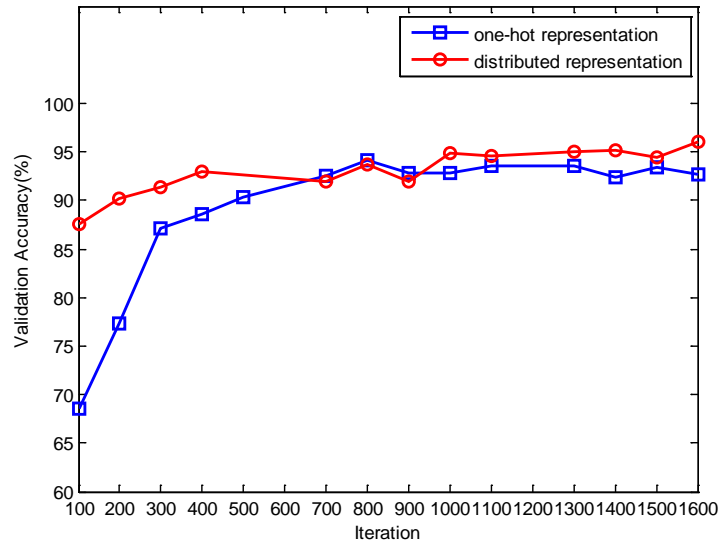
Fig. 9. Performance of one-hot model and skip-gram model on the validation set during training.

With regard to the Sogou news corpus, the one-hot vector is combined with CNNs for training. The word vectors based on the skip-gram model are combined with the CNNs for training. Comparisons of the performance of the two methods are shown in **Fig. 9** and **Table 1**, respectively.

The selected application lists for each class and the number of applications in each class are shown in **Table 1**.

Table 1. Performance of various models in the test set of the big data.

	Precision	Recall	F1-score	Support
Skip-gram	0.97	0.97	0.97	10000
One-hot	0.96	0.96	0.96	10000

	sports	finance	estate	home	education	technology	fashion	politic	game	entertainment
sports	991	0	0	0	2	1	0	4	1	1
finance	0	992	0	0	2	1	0	5	0	0
estate	0	1	996	0	1	1	0	0	0	1
home	0	14	0	912	7	15	9	29	3	11
education	2	9	0	12	892	22	18	21	10	14
technology	0	0	0	10	1	968	4	3	12	2
fashion	1	0	0	9	4	4	971	0	2	9
politic	1	16	0	4	18	12	1	941	1	6
game	2	4	1	5	4	5	10	1	962	6
entertainment	1	0	1	6	4	3	5	0	1	979

(a)

	sports	finance	estate	home	education	technology	fashion	politic	game	entertainment
sports	998	0	0	0	2	0	0	0	0	0
finance	0	989	5	0	1	0	0	5	0	0
estate	0	24	953	8	3	0	4	6	1	1
home	0	13	22	892	5	14	42	8	4	0
education	1	3	0	2	962	13	4	9	2	4
technology	0	0	3	2	0	977	8	1	9	0
fashion	0	0	0	2	2	0	996	0	0	0
politic	0	11	8	0	18	17	0	943	1	2
game	0	0	0	1	0	2	13	1	981	2
entertainment	0	1	0	3	3	1	11	0	2	979

(b)

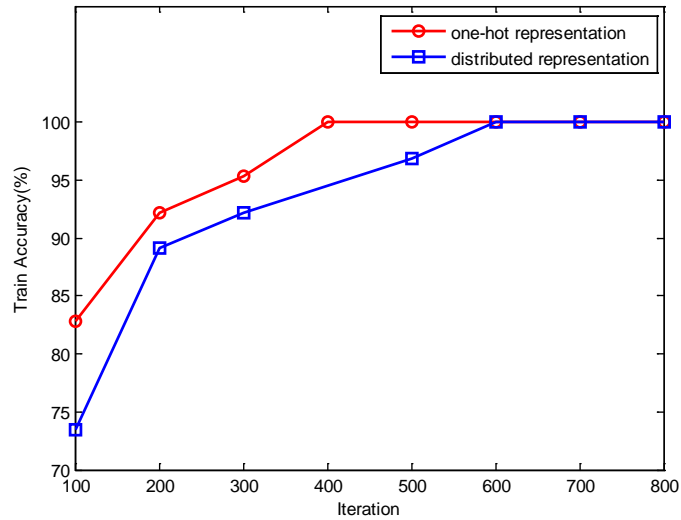**Fig. 10.** Confusion matrices of the two methods. (a) One-hot; (b) Skip-gram.

Fig. 11. Convergence speed of the one-hot and skip-gram models on one-fifth the original size data.

The F1-score in the **Table 1** is calculated as in formula (6):

$$F1-score = 2 \bullet \frac{precision \bullet recall}{precision + recall} \tag{6}$$

The accuracy of the skip-gram model is 2–3 percentage points higher than that of the one-hot model on the validation set in **Fig. 9**, which draws a point every 100 iterations. The total training step is about 1600 times. The comparison of accuracy between the one-hot model and the skip-gram model on 1,000 test sets is presented in **Table 1**; on the test set, the skip-gram model is one percentage point higher than the model with one-hot. Each test set has 10 categories. The aforementioned experiments prove that training the word vector with the skip-gram model can improve the classification accuracy of CNNs.

	sports	finance	estate	home	education	technology	fashion	politic	game	entertainment
sports	97	0	0	0	0	0	1	1	1	0
finance	0	99	0	0	0	0	0	1	0	0
estate	0	1	94	3	1	0	0	1	0	0
home	0	0	2	66	3	0	22	0	4	3
education	0	0	0	3	91	0	1	2	2	1
technology	0	0	0	6	48	2	1	27	4	12
fashion	0	0	0	3	1	2	91	0	3	0
politic	0	3	0	3	2	0	0	91	0	1
game	0	1	0	0	1	3	1	3	89	2
entertainment	2	0	0	0	2	0	5	2	1	88

(a)

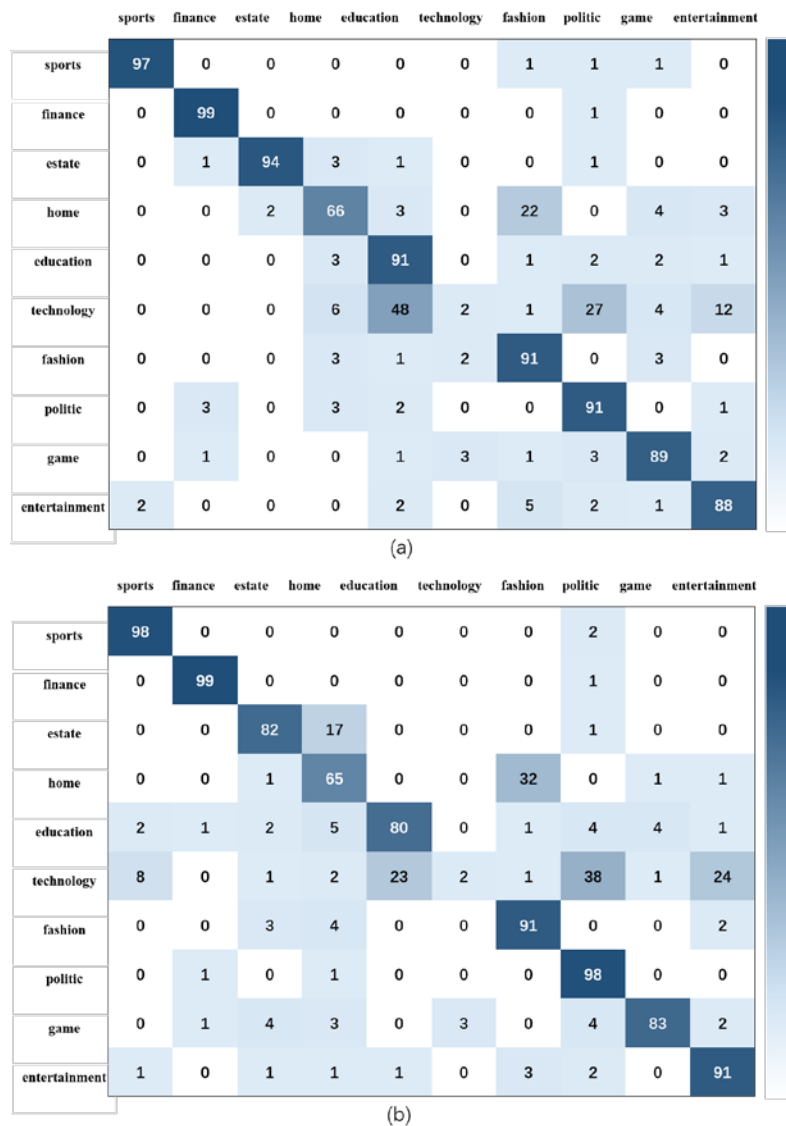|	sports	finance	estate	home	education	technology	fashion	politic	game	entertainment
sports	98	0	0	0	0	0	0	2	0	0
finance	0	99	0	0	0	0	0	1	0	0
estate	0	0	82	17	0	0	0	1	0	0
home	0	0	1	65	0	0	32	0	1	1
education	2	1	2	5	80	0	1	4	4	1
technology	8	0	1	2	23	2	1	38	1	24
fashion	0	0	3	4	0	0	91	0	0	2
politic	0	1	0	1	0	0	0	98	0	0
game	0	1	4	3	0	3	0	4	83	2
entertainment	1	0	1	1	1	0	3	2	0	91

(b)

Fig. 12. Confusion matrices of the (a) is one-hot and (b) skip-gram models.

4.3.2 The Small Data Case

For rigor in the experimental conclusions, we reduced the experimental data to one-fifth of the original size to retrain and classify. The final experimental results are shown in **Fig. 11** and **Table 2**. It draws a point per 100 steps in **Fig. 11**.
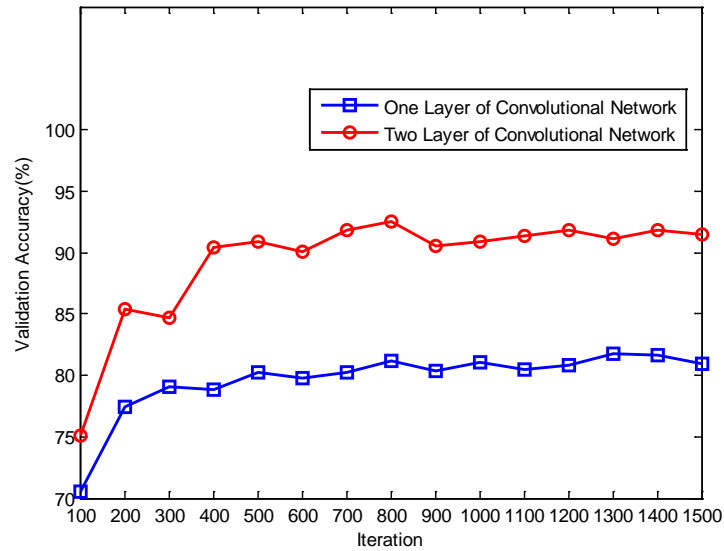

Fig. 13. Comparison of single-layer and double-layer convolutional neural networks on the validation set.

Table 2. Performance of various models in the test set of the small data

	Precision	Recall	F1-score	Support
Skip-gram	0.77	0.81	0.77	1000
One-hot	0.76	0.79	0.75	1000

Confusion matrices of the proposed models on big data set and small data set are provided in **Fig. 10** and **Fig. 12**, respectively. The Sogou news corpus is reduced to one-fifth of its original size, and the one-hot and skip-gram models are trained respectively. As determined from the training results, the one-hot model converges much faster than the skip-gram model, the convergence speed of the one-hot model is about 200 steps faster than that of the skip-gram model. The total step size is 800.

Compared with the one-hot model, the skip-gram model doesn't provide higher accuracy on the test set. According to **Table 2**, the performance of the one-hot model is superior to that of the skip-gram model on 1000 test sets. Because the training data set is not enough, the word vectors under the reduced corpus are not as good as the original corpus. Therefore, by using the skip-gram model to train the classifier, the accuracy is not as high as the one-hot model. Moreover, owing to the relatively high vector dimension of the skip-gram word, the calculated amount is relatively large, and the convergence speed is not as fast as the one-hot model.

4.4. Comparison of One-layer and Two-layers CNNS

To further improve the accuracy of classification, this study attempts to increase the convolutional layer of the CNNs to two layers. The one convolutional layer and two

convolutional layers are compared in the small-data environment.

Comparison between the one-layer and two-layer neural networks in the **Fig. 13**, which draws a point every 100 iterations. It indicates that the performance of the latter is 10% higher than the former on the test set. The results are also listed in **Table 3**. The performance of the two-layer CNNs is better than that of the one-layer CNNs.

Table 3. Performance of various models in the test set of the small data

	Precision	Recall	F1-score	Support
One-layer	0.77	0.81	0.77	1000
Two-layers	0.81	0.81	0.78	1000

	sports	finance	estate	home	education	technology	fashion	politic	game	entertainment
sports	98	0	0	0	0	0	0	2	0	0
finance	0	99	0	0	0	0	0	1	0	0
estate	0	0	82	17	0	0	0	1	0	0
home	0	0	1	65	0	0	32	0	1	1
education	2	1	2	5	80	0	1	4	4	1
technology	8	0	1	2	23	2	1	38	1	24
fashion	0	0	3	4	0	0	91	0	0	2
politic	0	1	0	1	0	0	0	98	0	0
game	0	1	4	3	0	3	0	4	83	2
entertainment	1	0	1	1	1	0	3	2	0	91

(a)

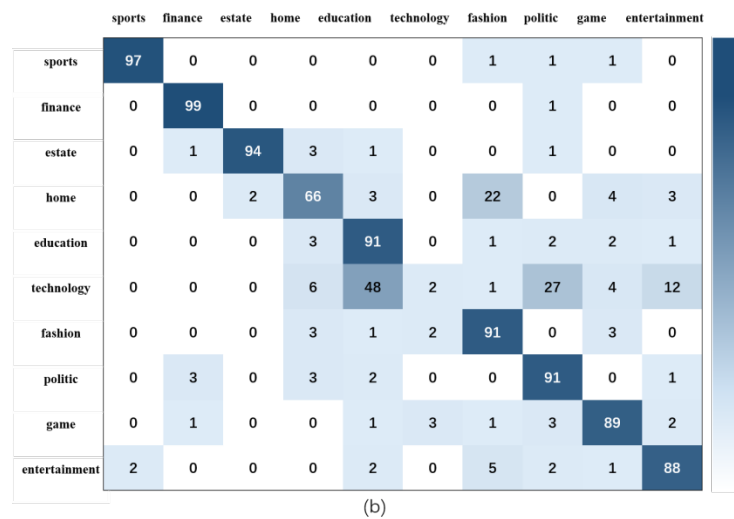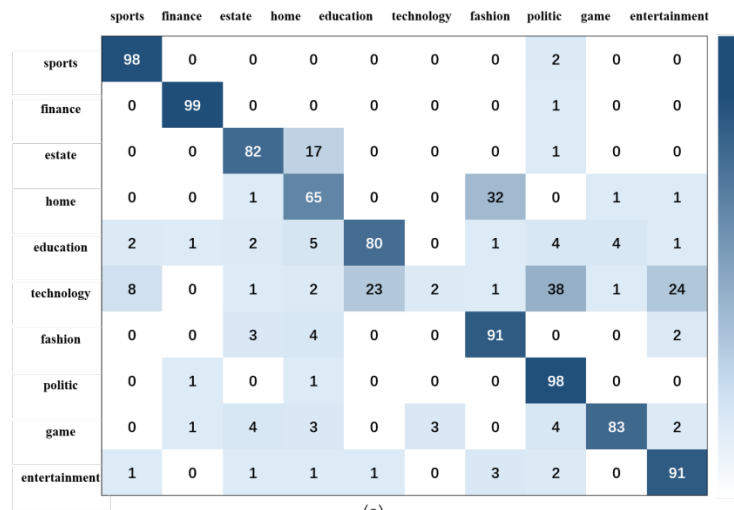|	sports	finance	estate	home	education	technology	fashion	politic	game	entertainment
sports	97	0	0	0	0	0	1	1	1	0
finance	0	99	0	0	0	0	0	1	0	0
estate	0	1	94	3	1	0	0	1	0	0
home	0	0	2	66	3	0	22	0	4	3
education	0	0	0	3	91	0	1	2	2	1
technology	0	0	0	6	48	2	1	27	4	12
fashion	0	0	0	3	1	2	91	0	3	0
politic	0	3	0	3	2	0	0	91	0	1
game	0	1	0	0	1	3	1	3	89	2
entertainment	2	0	0	0	2	0	5	2	1	88

(b)

Fig. 14. Confusion matrices of the two methods.
(a) One convolutional layer; (b) Two convolutional layers.

5. Conclusion

This paper proposed a method that combined the skip-gram model with CNNs to improve the accuracy of text classification. In order to confirm the applicable scenario of the method, the data set was reduced to one-fifth of its original size for experiments. The skip-gram model was better than one-hot model only when the training data set was enough large. Therefore, the final results indicated that the performance of the skip-gram model with CNNs could be improved in the big data case. In the small-data case, the skip-gram model could not train good word vectors, so the CNNs with skip-gram model cannot improve the accuracy. Finally, this study added another convolutional layer of CNNs. The final experimental results indicated that the two-layer CNNs could indeed improve the performance of the classifier.

References

[1] R. Carreira, J. Crato, D. Goncalves, and J. Jorge, "Evaluating Adaptive User Profiles for News Classification," in *Proc. of the 9th International Conference on Intelligent User Interface, Funchal, Madeira, Portuga*, pp. 206-212, 2004. Article (CrossRef Link)

[2] K. Zhao and Y. Jin, "A Hybrid Method for Sentiment Classification in Chinese Movie Reviews Based on Sentiment Labels," in *Proc. of 2015 International Conference on Asian Language Processing (IALP), Suzhou, China*, pp. 86–89, 2015. Article (CrossRef Link)

[3] N. Jatana and K. Sharma, "Bayesian Spam Classification: Time Efficient Radix Encoded Fragmented Database Approach," in *Proc. of 2014 International Conference on Computing for Sustainable Global Development, Canberra, Australia*, pp. 939–942, 2014. Article (CrossRef Link)

[4] T. Dodiya and S. Jain, "Question Classification for Medical Domain Question Answering System," in *Proc. of IEEE International WIE Conference on Electrical and Computer Engineering, Pune, India*, pp. 204–207, 2016. Article (CrossRef Link)

[5] M. Liu, T. Song, and G. Gui, "Deep Cognitive Perspective: Resource Allocation for NOMA Based Heterogeneous IoT with Imperfect SIC," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2885-2894, Apr, 2019. Article (CrossRef Link)

[6] G. Guan, *et al.*, "Deep Learning for An Effective Non-Orthogonal Multiple Access Scheme," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8440–8450, 2018. Article (CrossRef Link)

[7] Y. Wang, *et al.*, "Data-Driven Deep Learning for Automatic Modulation Recognition in Cognitive Radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074-4077, Apr. 2019. Article (CrossRef Link)

[8] H. Huang, *et al.*, "Unsupervised Learning Based Fast Beamforming Design for Downlink MIMO," *IEEE Access*, vol. 7, pp. 7599-7605, 2018. Article (CrossRef Link)

[9] N. Kato *et al.*, "The Deep Learning Vision for Heterogeneous Network Traffic Control: Proposal, Challenges, and Future Perspective," *IEEE Wirel. Commun.*, vol. 24, no. 3, pp. 146–153, 2017. Article (CrossRef Link)

[10] B. Mao *et al.*, "A Novel Non-Supervised Deep Learning Based Network Traffic Control Method for Software Defined Wireless Networks," *IEEE Wirel. Commun.*, vol. 25, no. 4, pp. 74-81, 2018. Article (CrossRef Link)

[11] F. Tang *et al.*, "On Removing Routing Protocol from Future Wireless Networks: A Real-time Deep Learning Approach for Intelligent Traffic Control," *IEEE Wirel. Commun.*, vol. 25, no. 1, pp. 154–160, 2018. Article (CrossRef Link)

[12] Z. M. Fadlullah *et al.*, "State-of-the-Art Deep Learning: Evolving Machine Intelligence toward Tomorrow's Intelligent Network Traffic Control Systems," *IEEE Comst*, vol. 19, no. 4, pp. 2432–2455, 2017. Article (CrossRef Link)

[13] J. Pan, *et al.*, "Deep Learning-Based Unmanned Surveillance Systems for Observing Water Levels," *IEEE Access*, vol. 6, pp. 73561–73571, 2018. Article (CrossRef Link)

[14] X. Sun, G. Gui, Y. Li, R. P. Liu, "ResInNet: A Novel Deep Neural Network with Feature Reuse for Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 679-691, Feb. 2019. Article (CrossRef Link)

[15] H. Li, K. Ota, and M. Dong, "Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing," *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, 2018. Article (CrossRef Link)

[16] F. Tang, B. Mao, Z. M. Fadlullah, and N. Kato, "On a Novel Deep-Learning-Based Intelligent Partially Overlapping Channel Assignment in SDN-IoT," *IEEE Commun Mag.*, vol. 56, no. 9, pp. 80–86, 2018. Article (CrossRef Link)

[17] X. Ma, J. Zhang, Y. Zhang, and Z. Ma, "Data Scheme-Based Wireless Channel Modeling Method: Motivation, Principle and Performance," *J. Commun. Inf. Networks*, vol. 2, no. 3, pp. 41–51, 2017. Article (CrossRef Link)

[18] F. Zhu *et al.*, "Image-Text Dual Neural Network with Decision Strategy for Small-Sample Image Classification," *Neurocomputing*, vol. 328, pp. 182-188, 2019. Article (CrossRef Link)

[19] R. Zhu, Z. Wang, Z. Ma, G. Wang, and J.-H. Xue, "LRID: A New Metric Of Multi-Class Imbalance Degree Based on Likelihood-Ratio Test," *Pattern Recognit. Lett.*, vol. 116, pp. 36–42, 2018. Article (CrossRef Link)

[20] X. Li *et al.*, "Supervised Latent Dirichlet Allocation with A Mixture of Sparse Softmax," *Neurocomputing*, vol. 312, pp. 324–335, 2018. Article (CrossRef Link)

[21] J. Xiong, *et al.*, "Background Error Propagation Model Based RDO in HEVC for Surveillance and Conference Video Coding," *IEEE Access*, vol. 6, pp. 67206–67216, 2018. Article (CrossRef Link)

[22] A. Krizhevsky, I. Sutskever, and H. Geoffrey, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no, 6, pp. 84-90, 2017. Article (CrossRef Link)

[23] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *Empirical Methods in Natural Language Processing, Doha, Qatar*, 2014, pp. 1746–1751, 2014. Article (CrossRef Link)

[24] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," *Neural Information Processing Systems, Nevada, United States*, pp. 1–9, 2013.

[25] F. Figueiredo, L. Rocha, T. Couto, T. Salles, M. A. Gonçalves, and W. Meira, "Word co-occurrence Features for Text Classification," *Inf. Syst.*, vol. 36, no. 5, pp. 843–858, 2011. Article (CrossRef Link)

[26] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A Neural Probabilistic Language Model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, 2001. Article (CrossRef Link)

[27] X. Zhang, Junbo Zhao, "Character-level Convolutional Networks for Text Classification," *Neural Information Processing Systems, Montreal Canada*, pp. 649–657, 2015.

[28] P. Liu, X. Qiu, and X. Huang, "Learning Context-sensitive Word Embeddings with Neural Tensor Skip-gram Model," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 2, pp. 232–244, 2000.

Wenhua Xu received her B.S. degree in Electronic Engineering from Shangdong Normal University in 2017. She is currently working toward the master degree in Nanjing University of Posts and Telecommunications (NJUPT). Her research interests include deep learning, optimization, and its application in image processing.

Hao Huang received his B.S. degree in Mathmatics from Nanjing University of Posts and Telecommunications (NJUPT) a in 2017. He is currently working toward the master degree in NUPT. His research interests include deep learning, optimization, and its application in image processing.

Jie Zhang received her B.S. degree in In and Engineering from Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, China in 2017. She is currently working toward the M.Sc degree in NUPT. His research interests include deep learning and its application in natural language processing.

Hao Gu received his B.S. degree in Communication Engineering from Nantong University in 2018. He is currently working toward the master degree in Nanjing University of Posts and Telecommunications (NJUPT). His research interests include deep learning, optimization, and its application in image processing.
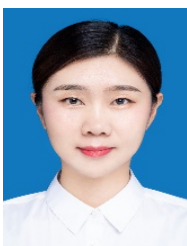
Jie Yang received the B.Sc. degree, M.Sc degree, PhD degree in communication engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2003 and 2006, 2018, respectively. She is Assistant Professor with Nanjing University of Posts and Telecommunications, Nanjing China.

Guan Gui received the Dr. Eng degree in Information and Communication Engineering from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2012. From 2009 to 2014, he joined the wireless signal processing and network laboratory (Prof. Adachi laboratory), Department of Communications Engineering, Graduate School of Engineering, Tohoku University as for research assistant as well as postdoctoral research fellow, respectively. From 2014 to 2015, he was an Assistant Professor in Department of Electronics and Information System, Akita Prefectural University. Since 2015, he has been a professor with Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, China. He is currently engaged in research of deep learning, compressive sensing and advanced wireless techniques. Dr. Gui has published more than 200 international peer-reviewed journal/conference papers. He received Member and Global Activities Contributions Award in IEEE ComSoc and seven best paper awards, i.e., ICEICT 2019, ADHIP 2018, CSPS 2018, ICNC 2018, ICC 2017, ICC 2014 and VTC 2014-Spring. He was also selected as for Jiangsu Specially-Appointed Professor (2016), Jiangsu High-level Innovation and Entrepreneurial Talent (2016), Jiangsu Six Top Talent (2018), Nanjing Youth Award (2018). Dr. Gui was an Editor of Security and Communication Networks (2012~2016). He has been the Editor of IEEE Transactions on Vehicular Technology, since 2017, the Editor of IEEE Access, since 2018, the Editor of KSII Transactions on Internet and Information Systems since 2017, the Editor of Journal of Communications, since 2019, and the Editor-in-Chief of EAI Transactions on Artificial Intelligence, since 2018. He is IEEE Senior Member.