Chinese Journal of Electronics
Vol.29, No.2, Mar. 2020

A Short Text Classification Method Based on *N*-Gram and CNN

WANG Haitao[1], HE Jie[1], ZHANG Xiaohong[1] and LIU Shufen[2]

(1. *College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454000, China*)

(2. *College of Computer Science and Technology, Jilin University, Changchun 130012, China*)

Abstract — Text classification is a fundamental task in Nature language process (NLP) application. Most existing research work relied on either explicate or implicit text representation to settle this kind of problems, while these techniques work well for sentence and can not simply apply to short text because of its shortness and sparseness feature. Given these facts that obtaining the simple word vector feature and ignoring the important feature by utilizing the traditional multi-size filter Convolution neural network (CNN) during the course of text classification task, we offer a kind of short text classification model by CNN, which can obtain the abundant text feature by adopting none linear sliding method and *N*-gram language model, and picks out the key features by using the concentration mechanism, in addition employing the pooling operation can preserve the text features at the most certain as far as possible. The experiment shows that this method we offered, comparing the traditional machine learning algorithm and convolutional neural network, can markedly improve the classification result during the short text classification.

Key words — Short text, Classification, Convolution neural network, *N*-gram, Concentration mechanism.

I. Introduction

In recent years, due to the rapid development of accession and technology for Mobile Internet, this brings about the great number of customs of Internet, which means that the year of Internet is coming. With the very firmly connection between users and Internet, the text information, being the mainly means of network interaction, is constantly emerging from users and devices of network. Facing a large amount of text information, how to filter and classify the valuable text information efficiently becomes more and more important and critical, while text content can be roughly divided into long text and short one according to its length. Short text usually appears in comments, micro-blog, Q&A and so on, which owns the characteristics of rapid growth and huge quantity. Therefore, it is very important to research on the short text in the Natural language processing (NLP) field, and short text classification work is becoming a critical scientific task for researches in the recent year[1].

For the research task of short text classification, the key issue is how to extract text features to express the meaning of short text, and the traditional sentence model uses the word bag model, which is a representation method based on vector space model, in addition, sentences and documents are deemed as the disordered word sets, therefore every feature word is independent from each other[2]. This kind of model does not contain word order and grammatical information, which has the representative problems, such as dimensional disaster, sparse feature and so on.

Therefore, with the research work going head in recent years, the deep neural network is becoming the mainstream framework of text sentence process, which is very suitable for the text classification research and is sensitive to the word order of text content. At present, because the Convolution neural network (CNN) has a significantly high accuracy rate of feature, it is widely applied in the field of image recognition and is one of the most representative neural network structures during the course of deep learning. Subsequently it was also applied to the field of natural language processing. Using CNN technology carries out the task of text processing, the first important thing is to execute the digitalization process in order to express the meaning of text correctly, and the traditional method is to utilize word vector technology, which was first proposed by Hinton[3], where word embedding is the most effective and

Manuscript Received May 21, 2019; Accepted July 22, 2019. This work is supported by the National Natural Science Foundation of China (No.61503124, No.61572379).

DOI:10.1049/cje.2020.01.001

representative method because it can maintain semantic and grammatical information as far as possible.

Because the convolution neural network has already gained many tasks of challenge champions tournament in the field of image, which enables researchers to apply this method to the other fields, and CNN[4] has made great achievements in the field of natural language processing in recent years. The CNN model which designed by Lei *et al.* adopts a nonlinear and discontinuous convolution method, in addition, the good classification results are achieved through large amount of experiments. Liang *et al.*[5] proposed a sentiment classification model based on multi concentration CNN on specific target datasets, they got an ideal emotional classification effect than the general convolution neural network does when combining three kinds of concentration mechanism. Bahdanau *et al.*[6] introduced the concentration mechanism during the course of machine translation task, which achieved significant result, after that, applied in the Google translation system and proved that the concentration mechanism is feasible in the field of natural language processing. Guo *et al.*[7] proposed an enhanced CNN model, which integrates three methods optimized, improved the efficiency and performance in aspect of lexical semantic features by convolution and pooling operations, it extracted the semantic features from different angles to enrich the semantic expression ability of the model. A multi-size filter CNN model proposed by Kim *et al.* suits for the static and no-static word vectors input, being one of the most representative model can effectively resolve the problem of similarity for antonyms vector under the same semantic situation in the field of natural language processing because of simple structure and effectiveness. Refs.[8,9] analyzed the sensitivity of CNN model which Kim offered on the task of text categorization, conducted the several groups of experiments by tuning the different super parameters, and compared its impact on the accuracy and stability for model, finally, some suggestions are presented for optimization and improvement.

From the detail what we discussed above, the goal of this paper is to survey existing methods of text categorization by comparing their advantages and defects, being a major extension of previously published paper[10], it also provided a overall theoretical analysis. Other surveys about text classification have been conducted, such as Refs.[8,10], but they pursue a different goal. In Ref.[11] the author only focus on centralized solutions to optimize classification model whereas we aim at the parallel and distributed solutions. Research goal of Refs.[12,13] is also oriented towards centralized techniques and is solely based on a theoretical performance analysis. Our approach integrates both theoretical design and practical performance analysis, obtained the ideal result through extensive experiments. To the best of our knowledge, it is a novel classification solutions for the short text, the breakthrough of this paper is that solutions are experimentally compared in the same setting: same hardware and same data-set, moreover we presented in this paper experimental settings and configurations which did not appear in the original paper.

Overall, our contributions of this paper are listed as follows.

- This paper presented an innovative classification method integrated the N-gram and the CNN technology.
- We adopted the concentration mechanism based on N-gram language model obtain the text features, adjusted the different super parameters and draw advantages about concentration model.
- Our method can preserve the effective features as soon as possible by the pooling operation improved, which promoted the accuracy increasing in the task of text classification.

The rest of this paper is organized as follows, Section II describes the fundamental concepts and theory of CNN model which we will adopt in the next section, our proposed method is described and analyzed in detail in Section III, and experimental results are discussed in Section IV. Section V concludes this paper finally.

II. Related Theory

It's a common way for human to understand everything by text information in the world, however, if using computer, this's unfeasible for text to regard as the input of model directly because the machine can't master the meaning of text straightly. So, a suitable approach of text expression needs to seek at the beginning of task, which can turn the text into the digital form in order to recognize the text information for the computers. At present, the most traditional method utilized is the word vector expression, and the input unit is the word vector in the field of natural language processing. The most common representation of word vectors includes one-hot representation and word embedding.

$$w_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \cdots, w_n = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (1)$$

where $w_i (i = 1, 2, \cdots, n)$ denotes the each word. The principle of one-hot word vector is to count all words, then to the number of each word, and uses a vector to represent a word, which each dimension of word vector represents a word, and the dimension value corresponding number is one, and the other dimension value is zero. The characteristics of representation are simple and effective, however, it will result in the fact of lexical gap, namely, any of two words are isolated. Moreover, it is prone to lead to the phenomenon of data sparseness and dimension disaster. So, the word embedding vector representation method was invented at a later time, and it is widely used in NLP. The basic idea is to express words in form of a low dimensional vector after words training, the vector obtained makes the similar words distribute closely, which can indicate the semantic connection between words and make up for the defects of on-hot expression, is suitable for the abstract expression of text. If the model adopts the word-embedding form to express, the training course will utilize the word2vector tools to execute.

The CNN model has many different size-scale filters, so the different features of input vectors should be extracted, which results in a good classification effect for text categorization. The detailed steps are listed as follows:

1) Initialize a two-dimensional matrix for input layer, namely $n * d$, where n denotes the number of words in a sentence, and d represents the dimension of the word vector for each word.

2) Conduct convolution operation for input word vector, the main thought is listed as follows, slide windows from top to bottom through multiple convolution kernels, where there are 3 kinds of convolution kernel size, namely $3 * d,\ 4 * d,$ and $5 * d$, the number of convolutions kernel of every convolution is 100, the convolution calculation is expressed as Eq.(2).

$$c_i = f(W \times x_{i:i+h-1} + b) \tag{2}$$

where c_i is the output after convolution, $x_{i:i+h-1}$ is the word vector matrix in the sliding window from i to $i+h-1$, W is the convolution kernel, b is the offset value, f is the activation function.

3) Carry out the pooling operation in order to extract more effective features and reduce the workload of computation. The pooling way adopts the maximizing pooling operation, and the expression is listed in Eq.(3).

$$c = \max\left\{c_1, c_2, c_3, \cdots, c_{n-h+1}\right\} \tag{3}$$

where c is the output result after maximum pooling operation, c_i is the output result after convolution operation.

4) Execute the concentration operation for 3 kinds of filter convolution, feature map after pooling, namely connection operation, finally get the characteristic map of $300 * 1$.

5) Perform the whole connection operation with final classification neurons, and softmax operation, then get the result of classification. The expression is listed in Eq.(4).

$$i = \operatorname{argmax}\left[\frac{a_i}{\sum_{j=1}^{k} a_j}\right] \tag{4}$$

where i is the output of final classification results, and a_i is the output result of softmax layer. The loss function used in model is cross entropy loss, The expression is listed in Eq.(5).

$$H_y = -\sum_i \left(y_i{}' \times \log\left(y_i\right)\right) \tag{5}$$

where $y_i{}'$ is the forecast category, y_i is real category, H_y is the cross entropy of both.

In an $n-$gram language model, we treat tow histories as equivalent if they end in the same $n-1$ words, *i.e.*, we assume that for $k \geq n$, $P\left(w_k|w_1^{k-1}\right)$ is equal to $P\left(w_k|w_1^{k-n+1}\right)$. For a vocabulary of size V, a 1-gram model has $V-1$ independent parameters, one for each word minus one for the constraint that all of the probabilities add up to 1. A 2-gram model has $V(V-1)$ independent parameters of the form $P\left(w_2|w_1\right)$ and $V-1$ of the form $P(w)$ for a total of V^2-1 independent parameters. In general, an n-gram model has V^n-1 independent parameters: $V^{n-1}-1$ of the form $k \geq n$, $P\left(w_n|w_1^{n-1}\right)$, which we call the order-n parameters, plus the $V^{n-1}-1$ parameters of an $(n-1)$-gram model.

We estimate the parameters of an n-gram model by examining a sample of text, t_1^T, which we call the training text, in a process called training, if *C(w)* is the number of times that the string w occurs in the string t_1^T, then for a 1-gram language model the maximum likelihood estimate for the parameter $P(w)$ is $C(w)/T$. To estimate the parameters of an n-gram model, we estimate the parameters of the $(n-1)$-gram model which it contains and then choose the order-n parameters so as to maximize $P\left(t_n^T|t_1^{n-1}\right)$. Thus, we call this model of parameter estimation sequential maximum likelihood estimation.

We can think of the order-n parameters of an n-gram model as constituting the transition matrix of a Markov model the states of which are sequences of $n-1$ words. Thus, the probability of a transition between the state $w_1w_2\cdots w_{n-1}$ and the state $w_2w_3\cdots w_n$ is $P\left(w_n|w_1w_2\cdots w_{n-1}\right)$. The steady-state distribution for this transition matrix assigns a probability to each $(n-1)$-gram which we denote $S\left(w_1^{n-1}\right)$. We say that an n-gram language model is consistent if, for each string w_1^{n-1}, the probability that the model assign to w_1^{n-1} is $S\left(w_1^{n-1}\right)$. Sequential maximum likelihood estimation does not, in general, lead to a consistent model, although for large values of T, the model will be very nearly consistent.

III. Short Text Classification Method Based on ACCN and N-Gram

In our work, the model input unit we adopted is the word vector expression of each word, and we utilized three kinds of convolution kernel size, namely $3 * dim$, $4 * dim$, and $5 * dim$, then in the pooling layer we employed the concentration mechanism in order to extract the feature representation effectively. Finally, fully connection layer and softmax is included, the frame is shown in Fig.1.

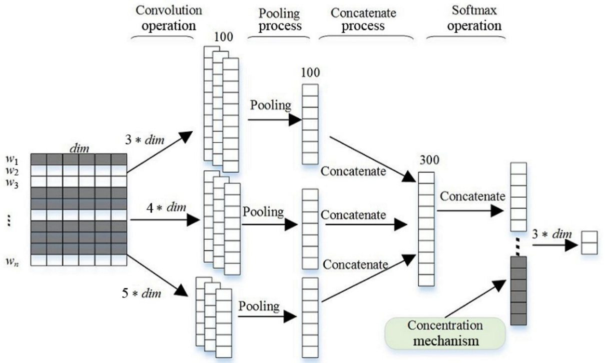

Fig. 1. Short text classification course of ACCN Model

N-gram, which is also known as the N meta model, is a very important concept in the course of natural language processing. Assume there is a string S, then the N-gram expression of string S indicates the word segmentations which divided the original word into the parts according to the length N, namely all of the substrings length is N in the string S. In order to get the abundant and effective text features, a sliding window model based on N-gram is designed in this paper, which utilized the non-linear combination way in the window. For example, there is a sentence as follows: ‘that man likes to watch movies’. To design a sliding window which size is 5, model is 4-gram, interval space is 1 and the steps of sliding windows equals 1, the detail is described as follows.

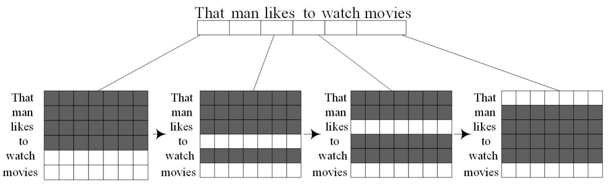

Fig. 2. Sliding process of 4–gram model windows

1) The content of current window is ‘that man likes to watch movie’, move a step length of word in the 4-gram model one by one, different text order is in turn obtained respectively, that is, ‘that man likes to’, ‘that man likes watch’, ‘that man to watch’, ‘that likes to watch’, ‘man likes to watch’.

2) The whole sliding window moves forward with a step length.

3) If the current window content is ‘that man likes to watch movies’, move forward a step length of word one by one in 4-gram model, just like the detail discussed above, the text order is respectively, ‘man likes to movies’, ‘man likes watch movies’, ‘man to watch movies’, ‘likes to watch movies’.

4) The rest operation conducts the repeated operation in the same way, sliding forward in turn.

If only using traditional sliding model, the text features are obtained as following, namely ‘that man likes to’, ‘man likes to watch’, ‘likes to watch movies’, while the sliding model based on the N-gram can obtain the more abundant text features, ‘that man likes to’, ‘that man likes watch’, ‘that man to watch’, ‘that likes to watch’, ‘man likes to watch, man likes to movies’, ‘man likes watch movies’, ‘man to watch movies’, ‘likes to watch movies’.

The traditional sliding windows model only focused on the linear sliding operation on the near words in a specific window, without considering the relationships between non-adjacent words. The sliding windows model based on N-gram, which presented in this paper, can obtain the more plentiful word vector of text by none continuous way, discarding the stop word, for example, prepositions, adverbs and so on, at the same time, we also get some important backbone of sentences, for example, just like Fig.2 shows, ‘man likes watch movies’.

Concentration mechanism achieved good results in the image recognition task[9], therefore it applied to the natural language processing field, and utilized this mechanism on the course of machine translation, the effect is very obvious and outstanding. According to these advantages, based on the windows sliding method, we applied the concentration mechanism on the course of windows sliding in order to extract some important and essential text features. The method we designed is similar to the conventional layer theory, and utilized the n-gram sliding window to move forward one by one, while it’s different from the conventional kernel, the weight value of every window does not share each other, the course of process is shown in Fig.3.

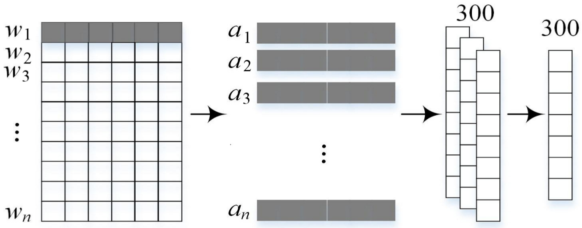

Fig. 3. Concentration mechanism process course

Under the sliding pattern mentioned in the above, we can obtain various phrase sets in every time of sliding. To assess the importance of each phrase sets under the same window, the formula 6 is adopted as following.

$$S_{i,j} = f\left(\sum W_i \times X_{j,j+1,\cdots,j+k-1} + b_i\right) \tag{6}$$

where $S_{i,j}$ represents the weight value of each phrase in the window, $X_{j,j+1,\cdots,j+k-1}$ denotes a set of word vector matrices in the current sliding window, W_i expresses the weight value of each phrase in the ith window, b_i is the offset value of the ith window, f is the activation function. In order to select the key phrases of each window, the expression shows in Eq.(7).

$$S_{i,m} = \max\left(S_{i,1}, S_{i,2}, \cdots, S_{i,l}\right) \tag{7}$$

It can be seen from the formula that the biggest phase set selected will be the key phrase set in the ith window. $S_{i,m}$ is the key phrase combination, $S_{i,l}$ represent the set of the lth phrase in the ith window. All of these parameters mean the key word vector set which can be obtained, namely a_i, and the expression shows as follows.

$$a_i = \left(w_{i,1}, w_{i,2}, \cdots\right) \tag{8}$$

where $w_{i,1}$ represents the first primitive word vectors in the ith window, after that conduct the convolution operation of keyword vectors, which the number of convolution kernels is 300, then we carry out the pooling operation and obtain the output result of feature vectors, and the expression shows in Eq.(9).

$$c_a = pooling\left(f\left(\sum w_a \cdot a_{1:n} + b_a\right)\right) \tag{9}$$

where c_a represents the output result of feature vector by the attention mechanism, $a_{1:n}$ denotes the keyword vector set, b_a is bias value, f is the activation function, pooling function denotes pooling operation.

Finally, we can obtain the output feature vector, namely $\hat{c}$, by using the different filter window, then carry out the concentrate operation with c_a, and add the bias value to activate output, the expression shows as follows.

$$h = f\left(\sum w \times \left[\hat{c} \oplus c_a\right]_{i:i+r-1} + b\right) \tag{10}$$

where h represents the feature vector of whole model output, w represent convolution kernel weight matrix, b is bias value and f is activation function.

To natural language processing field, the pooling operations generally include the max pooling operation and the average operation when using the convolution neural network to deal with the text classification. Pooling operations can compress data, reduce the number of parameters and avoid the fact of over-fitting and so on. There exists some limitations of these methods, therefore we adopt a pooling technology integrated with the above methods advantages, the expression is shown in Eq.(11).

$$\hat{c} = \frac{\sum_{i=1}^{k} \max_i \left\{c_1, c_2, \cdots, c_{n-h+1}\right\}}{k} \tag{11}$$

where $\hat{c}$ represents the output result after k-Max average pooling operation, $\sum_{i=1}^{k} \max_i \left\{c_1, c_2, \cdots, c_{n-h+1}\right\}$ represents the maximum output result after the first k convolution operation.

Max pooling operation is to make the maximum value of result as the output of sampling, while average pooling operation is to make the average value of result as the output of sampling. K-Max average pooling is to fetch the K largest values during the course of sampling, then calculate the mean value of them and regard as the final output result, which this approach avoids the fact that only one of the largest feature is selected during the max pooling operation course, thereby it can focus on the other feature effect. Moreover, it also prevents the phenomenon that the small features with the large distance weaken the entire feature strength during the course of average pooling. Therefore, K-Max average pooling method can retain strong feature strength and extract the sampling feature of word vectors effectively.

IV. Experimental Result and Analysis

In this section, we conduct the classification experiments on the various short text data sets to test the classification result which we proposed, while the course was carried out on Intel (R) i7-6500U 2.59GHz processor with 12GB memory, Python 3.5 programming environment and Theano 0.9 deep learning framework.

The experimental data we adopted are the short text datasets which widely used for text categorization in recent years, and the basic dataset includes the following kinds.

1) MR is the English movie reviews data sets, there is the 10662 recorders of data in total, the number of category cases has two kinds, namely positive and negative, and the average length of sentences is 20.

2) SST-1 is the English data set which is an extension of the MR data sets, there is the 11855 recorders data in total, the number of category cases has five kinds, and the average length of sentences is 18.

3) SST-2 is also the English data set which is similar to SST-1. Besides the general reviews, there are 9613 recorders data in total and two categories, the average length of sentences is 19.

4) ChnSentiCorp is the Chinese categorized data set, which collected from the Ctrip company and included three kinds of comments data, namely about books, hotels

and personal computers. We choose two classes of data, that is the books and personal computers, and there is the 8000 recorders data in total.

Word2Vec tool is used to train word vectors, and the value range of initialization sets by [–0.5, 0.5] when the word without appearing initialized randomly. The scale of convolution kernels is the $3*dim, 4*dim, 5*dim$ respectively, the number of each convolution kernels is 100, the value of batch size is 64, and the learning rate is 0.001. To prevent the fact of over-fitting, the dropout mechanism was introduced in the course of training, and the dropout rate sets by 0.5. During the attention mechanism, the jumping interval of the N-Gram model is 2, and the ten-fold cross validation is used in the course of experiment. Attention mechanism is added to the original CNN model, which can focus on the more important text features and keep them. In the pooling stage, the optimized operation mode adopts the K-MAX average pooling operation. In order to test its effectiveness, a set of comparative experiments was designed, while the evaluation parameter, namely precision, recall and F_1, are used to assess the effect of text categorization. Expressions formula is listed as following.

$$Precision = \frac{TP}{TP+FP}, \quad Recall = \frac{TP}{TP+FN} \tag{12}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{13}$$

where the TP represents the number of positive classes actually while being a positive one is predicted, FP represents the number of negative classes actually while being a positive one is predicted, FN represents the number of negative classes actually while being a negative one is predicted.

In order to demonstrate the better performance of convolution neural network model in the course of short text classification, this paper uses ChnSentiCorp Chinese dataset, which compared with traditional machine learning methods, such as NB(namely Naive Bayes), Random-forest and Linear-SVM also including the CNN model proposed by Kim. Giving the influence of different text feature representations, we used the word vectors technology in this paper when comparing with traditional machine learning methods. The comparison results are shown in Table 1.

From Table 1, we can see that the classification model based on convolution neural network has fairly better performance effect than traditional machine learning algorithm does in text classification field. The reason is that the deep learning model can considerably reflect classification features and retain more classification features, while the CNN-random precision is close to the CNN-static does, and the latter, namely 4-gram-ACNN precision, is higher than the former, the reason is that the former word vector model is randomly initialized and modified during the course of training, the latter is a word vector which obtained by word2vec training in advance, can fairly express text semantics. In this model, we adopt the attention mechanism to optimize the feature extraction. At the same time better pooling method is adopted in order to improve the classification effect. In order to explore the classification effect of model on different data sets, we use various English classification data sets to test and compare with the typical CNN-static method, then the classification effect of model is tested according to result, the accuracy of each classification is shown in Table 2.

Table 1. Precision comparison

Name of model	Precision(%)
NB	82.8
Random-forest	85.1
Linear SVM	88.5
CNN random	89.3
CNN static	91.2
4-gram-ACNN	92.6

Table 2. Different data sets precision ratio comparison(%)

Name of model	MR dataset	SST1 dataset	SST2 dataset
CNN static	80.6	45.3	86.2
4-gram-ACNN	81.2	48.6	86.9

From the experimental results, we can see that the method we proposed, namely 4-gram-ACC, has a good performance effect on each data set, and improves the classification accuracy obviously, it also shows that this model has good generalization ability on different data sets. For SST-1 data set, the model is improved greatly, which shows that the space of model optimization is larger in text multi-classification.

3) In order to select the N value of N-gram model which is the most ideal one, using different N-values conduct experiment operation on ChnSentiCorp data set, and the final classification results are shown in Table 3.

Table 3. Different N value of classification result comparison

Name of model	Precision(%)	Recall(%)	F_1(%)
1-gram-ACNN	91.8	93.2	92.5
2-gram-ACNN	92.2	94.3	93.2
3-gram-ACNN	92.6	94.9	93.7
4-gram-ACNN	92.4	93.7	93

As we can see from the Table 3, this model has the obvious effect on text classification in the certain of range with the N value increasing, When N is 4, the classification accuracy is the highest, because the trunk of a sentence can be well described and the vector features of the text words are more accurate, while N is 1, the classification accuracy is the worst, because the single word vector cannot represent the semantic features well, not taking the relationship between words into account.

V. Conclusions and Future Work

In this paper, we explored the knowledge about convolution neural network of typical multi-size filters, and aimed at the solution method for short text classification task, finally an ACNN model based on N-gram is proposed through our endeavor, which made the word vector as the input of model trained by word2vec tool, utilized the skipping and interval method to obtain the more effective feature expression in the sliding window, focused on the key feature property and combines the two different pooling operation by the concentration mechanism. Experiments are conducted on the different models with the different data sets, and experimental results show that the model we proposed is especially appropriate and feasible on the task of short text classification, and the classification effect increase remarkably in the certain extent.

In the future, we plan to apply our methods to the other short text classification task, such as PubMed data, to check the feasibility applied on the medical text context, in particular, the usage of fastText, which provides pre-trained word embedding generated with word2vec using a considerable amount of texts. Additionally, the exploration of other text representation extensions of word2vec is also one goal for our research works.

References

[1] Marcin Micha Mirończuk and Jarosaw Protasiewicz, "A recent overview of the state-of-the-art elements of text classification", *Expert Systems with Applications*, Vol.106, No.15, pp.36–50, 2018.

[2] S. N. Bharath Bhushan and Ajit Danti, "Classification of text documents based on score level fusion approach", *Pattern Recognition Letters*, Vol.94, No.15, pp.118–120, 2017.

[3] Hinton G E and Salakhutdinov R R, "Reducing the dimensionality of data with neural networks, *Science*, Vol.313, No.5786, pp.504–507, 2006.

[4] Lei T, Barzilay R and Jaakkola T, "Molding CNNs for text: Non-linear, non-consecutive convolutions", *Indiana University Mathematics Journal*, Vol.58, No.3, pp.1151–1186, 2015.

[5] LIANG Bin, LIU Quan, XU Jin, *et al.*, "A special sentiment analysis based on multi-concentration CNN", *Computer Research and Development*, Vol.54, No.8, pp.1724–1735, 2017.

[6] Bahdanau D, Cho K and Bengio Y, "Neural machine translation by jointly learning to align and translate", *Computer Science*, Vol.18, No.2, pp.124–135, 2014.

[7] Guo J, Yue B, Xu G, *et al.*, "An enhanced convolutional neural network model for answer selection", *International Conference on World Wide Web Companion, International World Wide Web Conferences Steering Committee*, pp.789–796, 2017.

[8] YANG Zhen, FAN Kefeng, LAI Yingxu, *et al.*, "Short text classification through reference document expansion", *Chinese Journal of Electronics*, Vol.23, No.2, pp.315–323, 2014.

[9] Zhang Y and Wallace B, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification", *Computer Science*, Vol.16, No.4, pp.874–880, 2015.

[10] LUO Fan and WANG HouFeng, "Chinese text sentimental classification combined the RNN with CNN hierarchical network", *Beijing University* (*Natural Science Version*). https://doi.org/10.13209/j.0479-8023, 2017.

[11] Mandelbaum A and Shalev A, "Word embeddings and their use in sentence classification tasks", *Machine Learning*, Vol.26, No.10, pp.1–15, 2016.

[12] FENG Xingjie, ZHANG Zhiwen and SHI Jinchuan, "Text sentimental analyses based CNN and concentration model", *Computer Application Research*, Vol.35, No.05, pp.1434–1436, 2018.

[13] ZHANG Jiang, SUN Qigan, LI Xue, *et al.*, "A novel feature selection method based on probability latent semantic analysis for Chinese text classification", *Chinese Journal of Electronics*, Vol.20, No.2, pp.228–232, 2011.

WANG Haitao was born in 1974, and received the Ph.D. degree from College of Computer Science and Technology, Jilin University, China, in 2016. Since 2011, he has been an associate professor of College of Computer Science and Technology in Henan Polytechnic University. His research interests include cloud computing, text data mining and Web info process. (Email: jz_wht@hpu.edu.cn)

HE Jie was born in 1987, and received his bachelor degree at Henan University of Science and Technology in 2018. His research interests include big data process and data mining. (Email: hejie_2018@126.com)

ZHANG Xiaohong (corresponding author) worked in Henan Polytechnic University now, received the Ph.D. degree in computer architecture from University of Chinese Academy of Sciences. She did one year of post-doc research on cloud computing in Wayne State University. Her main research interests include cloud computing and big data analysis. (Email: 1760778431@qq.com)

LIU Shufen was born in 1950, professor, Ph.D. tutor, worked in College of Computer Science and Technology of Jilin University, researched on computer collaboration process, network software, simulation and model and so on. (Email: Liusf@jlu.edu.cn)