

Dual-Channel Attention Model for Text Sentiment Analysis

Hui Li, Yuanyuan Zheng^{*}, and Pengju Ren

School of Physics and Electronic Information, Henan Polytechnic University, Jiaozuo, 454000, China

Abstract

Focused on the issue that text information cannot be fully extracted by the single-channel neural network model, the Dual-Channel Attention Model (DCAM) is proposed for text sentiment analysis. Firstly, text is represented in the form of a matrix using a word vector trained by Word2Vec. Secondly, the matrix is used as input data and sent to Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) networks for feature extraction. Thirdly, an attention model is introduced to extract important feature information. Finally, the text features are merged, and the classification layer is used to classify the sentiment. The model is evaluated on a Chinese corpus. According to the experimental results, the accuracy of the proposed model can reach 92.7%, which is obviously superior to other single-channel neural network models.

Keywords: convolutional neural network; long short-term memory; attention model; sentiment analysis; dual-channel

(Submitted on October 20, 2018; Revised on November 21, 2018; Accepted on December 25, 2018)

© 2019 Totem Publisher, Inc. All rights reserved.

1. Introduction

In recent years, with the rapid development of the Internet industry, there have been many emerging media that are constantly impacting and changing people's lifestyles. The rise of various e-commerce platforms has made online shopping simple and popular. The only basis for deciding whether a potential consumer will purchase is the experience comment left by the buyers, which is the only feedback from online shopping. Therefore, sentiment analysis of texts, especially some Chinese complex sentences, is a useful work for e-commerce platforms and consumer groups.

Sentiment analysis is also known as opinion mining [1]. The main task of text sentiment analysis is to analyze the text information, extract features, and make polarity judgments. At present, the sentiment lexicon and machine learning are the two main methods of text sentiment analysis. The method based on sentiment lexicon records the sentiment tendency and intensity of words or phrases by constructing a sentiment lexicon and then combines the sentiment words in a certain way, and the sentiment polarity of the text is finally obtained [2]. However, with the development of society, many emerging words cannot be completely included in the sentiment lexicon, so the method is not ideal in the face of increasingly diverse texts. In addition, the methods of machine learning require a large number of data features to be artificially designed. With the increase in text datasets to be processed, traditional machine learning has been unable to learn the deep information features of text quickly and well.

With the rapid development of deep learning in the fields of computer vision [3] and speech recognition [4], many research works begin to use deep learning algorithms to solve natural language processing problems [5-6]. Deep learning is a part of machine learning that is dedicated to solving complex tasks by selecting abstract feature representations from massive original data, and it has stronger learning ability than traditional machine learning.

Commonly used deep learning models for text sentiment analysis tasks include Convolutional Neural Network (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM). CNN has three characteristics of local receptive field, parameter sharing, and down sampling, which makes it have great advantages in capturing local feature information of text. RNN has a strong ability to capture context information, and it has great advantages in sequence feature

^{*} Corresponding author.

E-mail address: 13273919602@163.com

processing. Kim first applied CNN to short-text sentiment analysis, which proved its feasibility in the emotional classification task but only experimented with English text. Kalchbrenner et al. [7] proposed a K-MAX dynamic pooling model, which is different from the traditional pooling method and retains K important text features. Yangsen et al. [8] applied RNN to the sentiment analysis of Chinese microblog corpus. By training the features vector with word and sentence information, it was finally proven that calculating the sentence vector can help learn the deep structure of the sentence. Although RNN can learn sequence features well, gradient explosion or gradient disappearance occurs as time nodes increase. In order to solve this problem, the LSTM network was introduced, which can capture the sequence feature of the text and the context dependencies. In addition, it can alleviate the gradient explosion or gradient disappearance, and the performance is better than the standard RNN. CNN and LSTM were combined to classify the English text emotionally, and a good result was achieved [9]. The CNN pooling layer was canceled, and the text features learned by CNN were directly used as input to the LSTM network to learn more about the feature of text sequences [10]. This connection is similar to image annotation in computer vision tasks [11].

The importance of individual words is not considered when using traditional methods to extract text information features, but it is worth noting that each word in the text contributes differently to the sentiment polarity of the entire text. In some cases, one or several words can directly determine the sentiment polarity of the text.

Therefore, the attention mechanism is first applied to natural language processing tasks [12]. In essence, it allocates more attention to the important parts, so as to optimize the extracted features and improve the classification effect. The attention mechanism is inspired by human attention. When people browse pictures or texts, they can quickly get the whole picture, focus on several important parts, ignore other irrelevant information, and improve reading speed. The attention mechanism in deep learning initializes the attention matrix, associates it with existing or processed features, and finally obtains the probability matrix of the target, called the attention matrix. The larger the value, the greater the influence of the part. Yin et al. [13] proposed that attention models can be combined with different parts of the CNN. Feng et al. combined the CNN model with the attention model on the Chinese dataset and considered the correlation between text sentences and results, which improved the classification effect.

At present, most of the sentiment classification is based on the construction of a single-channel neural network model for text feature extraction. However, as the number of network layers increases, the performance of the single-channel model will be affected, and the features of the text cannot be fully extracted. Commonly used neural networks, such as CNN and LSTM, cannot distinguish the location of important information in a sentence when extracting features, so text classification results are greatly affected by irrelevant information. Therefore, in order to solve the above problems, inspired by [14], we propose a network model for text sentiment analysis of Chinese datasets. The main work is as follows:

(1) A Dual-Channel Attention Model (DCAM) is proposed to combine the advantages of CNN with LSTM. Besides that, the attention model [15] is introduced to extract important text features.

(2) By comparing experiments with the general network, it is proven that the Dual-Channel Model and attention mechanism can really improve the correct rate of text sentiment analysis.

2. DCAM Model

In this paper, the CNN and LSTM deep learning network are used to construct a Dual-Channel Attention Model, and attention mechanisms are also introduced in the model. The schematic diagram of the model structure is shown in Figure 1.

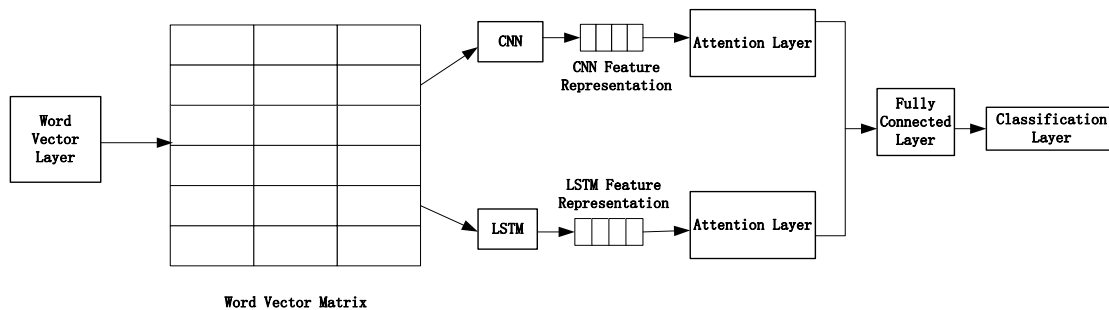


Figure 1. Schematic diagram of the dual-channel attention model

2.1. Word Vector Representation Layer

Due to the particularity of the text, it cannot be directly used as input data of the model like an image. Therefore, part of the important work of text sentiment analysis is training the word vector. The text is transformed into features that can be recognized by the computer in order to learn the semantic information contained in the text [16].

Word vector is a kind of technology that associates word vectors with semantics through the unsupervised training method. It is extremely important in data mining. At first, many studies use the one-hot word vector to represent the text, that is, the length of the vector is the size of the dictionary. Only one dimension in the vector is 1, its position corresponds to the position of the word in the dictionary, and the rest of the dimensions are all 0. The word vector represented by the one-hot method is very sparse, and when the dictionary capacity is too large, it is easy to cause a “dimension disaster”. In addition, there is no internal connection between word vectors, and the similarity between words cannot be expressed. Hinton proposed a word vector representation method for word embedding that maps words distributed in low-dimensional space. In 2013, Mikolov et al. [17] proposed the Word2Vec model, which convert words into the form of word vectors quickly. There are two training algorithms, continuous bag-of-words (CBOW) and Skip-gram. The difference between the two algorithms is that CBOW predicts target words by context, while Skip-gram uses target words to predict context. We use the Skip-gram model to train Chinese word vectors. The model structure is shown in Figure 2.

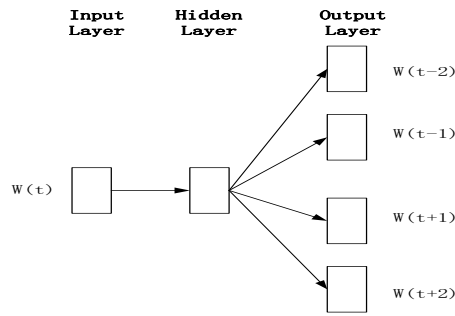


Figure 2. Schematic diagram of Skip-gram model

In the experiment of this paper, Word2Vec is used to train Chinese text as a d dimensions word vector in the following Equation (1):

$$x_i = [a_0, a_1, \dots, a_d] \quad (1)$$

The word vectors are spliced according to the order of the text words to form a word vector matrix as the input data of the model.

2.2. Feature Extraction Layer

2.2.1. CNN Feature Extraction Channel

Let $x_i \in R^d$ be the word vector corresponding to the i^{th} word in a text. The text is represented as a two-dimensional matrix of $l \times d$, where l represents the length of the sentence and d represents the word vector dimension. $X \in R^{l \times d}$ represents the input word vector matrix, which can be represented as Equation (2):

$$X = x_1 \oplus x_2 \oplus \dots \oplus x_l \quad (2)$$

Where \oplus is the concatenation operator. The CNN performs a convolution operation by using a linear filter on the input matrix, which is called the convolution kernel, and can be represented as a matrix of height h and width d , having $h \times d$ parameters. The convolution operation of the word vector matrix can be represented as Equation (3):

$$c_i = f(W \cdot X_{i+h-1} + b) \quad (3)$$

Where $b \in R$ is the bias term and f is the nonlinear activation function. In this experiment, ReLu is used as the activation function, which is represented as Equation (4):

$$f(x) = \max(0, x) \quad (4)$$

A convolution operation on the input sentence produces a feature map of C , which is represented as Equation (5):

$$C = [c_1; c_2; \dots; c_l] \quad (5)$$

Performing the maximum pooling operation will lose a lot of important information about the text. For longer texts, the features are not fully represented by one or several feature values. Therefore, after the convolution operation, the attention layer is added, which can learn the more important features in the text, ignore the unimportant features, and extract the features better. The working mechanism of the attention layer is as follows:

$$u_{ci} = \tanh(w_w c_i + b_w) \quad (6)$$

$$\alpha_{ci} = \frac{\exp(u_{ci}^T u_{cw})}{\sum_i \exp(u_{ci}^T u_{cw})} \quad (7)$$

$$s_c = \sum_i \alpha_{ci} c_i \quad (8)$$

Firstly, as described in Equation (6), the feature map c_i obtained by CNN is nonlinearly transformed into u_{ci} using the tanh function. Then, we initialize the attention matrix u_{cw} and multiply it by u_{ci} to obtain the normalized output weight of the CNN layer, which is represented as Equation (7). Finally, as described in Equation (8), the attention matrix is obtained, and the vector s_c is the text feature extracted by the CNN after the attention layer.

2.2.2. LSTM Feature Extraction Channel

LSTM is a special form of RNN. The LSTM used in this paper was proposed by Hochreiter and Schmidhuber in 1997. It is a special loop body structure that controls the transmission of information through the design of a “gate” to realize selective memory.

As shown in Figure 3, the LSTM controls the inflow and outflow of memory cell information through input gates, forget gates, and output gates. The LSTM repeating module has three inputs, namely the cell hidden layer state h_{t-1} at the previous moment, the cell memory state c_{t-1} at the previous moment, and the input x_t at the current moment, where t represents the current time and $t-1$ represents the previous moment. The converted word vector matrix is divided into l time steps, that is, each time step processes a word vector of a word. The specific working principle of LSTM can be expressed as the following Equations (9) to (14):

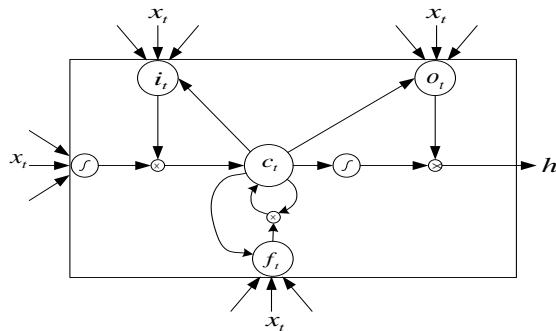


Figure 3. LSTM internal structure diagram

$$f_t = \sigma(w^f x_t + u^f h_{t-1} + b^f) \quad (9)$$

$$i_t = \sigma(w^i x_t + u^i h_{t-1} + b^i) \quad (10)$$

$$o_t = \sigma(w^o x_t + u^o h_{t-1} + b^o) \quad (11)$$

$$\tilde{c}_t = w^c x_t + u^c h_{t-1} + b^c \quad (12)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (13)$$

$$h_t = o_t \odot \tanh(c_t) \quad (14)$$

Where σ is the sigmoid activation function whose output is between [0,1] and determines how much information can pass. Tanh is a hyperbolic tangent function with an output is between [-1,1]. The \odot operator symbol represents the matrix multiplication operation. The forgot gate f_t decides to forget information that is not important in backpropagation. The input gate i_t determines how much information to be updated. The output gate o_t determines the content output from the current cell state c_t to the hidden layer state h_t . Finally, the current hidden layer state h_t is obtained. The current cell state c_t is the combination of the previous cell state c_{t-1} and the new memory. $w^{(i)}$ and $u^{(i)}$ represent the weights in the data processing process.

After learning the text feature information with LSTM, the attention mechanism is also introduced to extract important text features. The formula is expressed as the following Equations (15) to (17):

$$u_{li} = \tanh(w_{lw} h_i + b_{lw}) \quad (15)$$

$$\alpha_{li} = \frac{\exp(u_{li}^T u_{lw})}{\sum_i \exp(u_{li}^T u_{lw})} \quad (16)$$

$$s_t = \sum_i \alpha_{li} h_i \quad (17)$$

2.3. Classification Layer

The feature s_c extracted by the CNN layer that introduces the attention mechanism is merged with the feature vector s_l extracted by the LSTM layer, which also introduces the attention mechanism. The dropout technique is used to prevent over-fitting. When calculating, a part of the neurons is inactivated by random selection and then connected to the fully connected layer to obtain an output vector.

2.4. Model Training

The training of the model proposed in this paper uses the backpropagation method to update the parameters. The objective function is the cross-entropy cost function. y represents the real emotional category value of the text, and \hat{y} represents the emotion category value predicted by the model. The objective function is defined as Equation (18):

$$loss = -\sum [y^{(i)} \ln(\hat{y}^{(i)}) + (1 - y^{(i)}) \ln(1 - \hat{y}^{(i)})] \quad (18)$$

If the training text label is 1, the objective function is simplified to $loss = -\ln(\hat{y})$. The closer the predicted category is to 1, the closer the objective function value is to 0. Similarly, if the training text label is 0, the objective function is simplified to $loss = -\ln(1 - \hat{y})$. The closer the predicted category is to 0, the smaller the objective function. It can also be considered that the closer the objective function loss is to 0, the closer the model prediction value \hat{y} is to the true value y .

3. Experiment

3.1. Dataset and Pretreatment

To evaluate the performance of the proposed model, we use the dataset of Chinese shopping reviews from the Internet. There are 10696 positive comments (labeled 1) and 10428 positive comments (labeled 0). In order to eliminate the influence of irrelevant words in training word vectors, some words in the text are listed as stop words, such as punctuation mark, preposition, adverb, conjunctions, etc. We use jieba to segment the corpus and filter out the stop words. Then, the Word2Vec tool is used to train the word vector for the processed text. Finally, the trained word vector is stitched as the input data of the model.

3.2. Model Parameters

The text length is set to 60 words in order to ensure the uniformity of the size of the network input word vector matrix. The word vector dimension is set to 128. For the CNN channel, in order to extract different local information of the text, convolution kernels of different sizes are used and set to (3, 4, 5), and the number of convolution kernels per size is set to 128. For the LSTM channel, the number of hidden layer neurons is set to 128. To prevent overfitting, we set the Dropout rate to 0.5 (only when training the model). The batch size is 64 and the number of iterations (epoch) is 20. The optimizer used in the experiment is Adam.

3.3. Experiment

To prove the validity of the model proposed in this paper, the following comparative experiments are set up:

- CNN: A model for text classification using the CNN with pre-trained vectors from Word2Vec.
- LSTM: A model for text classification using the LSTM with pre-trained vectors from Word2Vec.
- CNN-Attention: After the feature extraction of CNN, the attention mechanism is introduced, and the model is single channel.
- LSTM-Attention: Same as above, but the model use LSTM to extract text features.
- DC-CNN-LSTM: The word vector trained by Word2Vec is used as the input of CNN and LSTM. This model does not introduce the attention mechanism.
- DCAM (rand): This model uses the Dual-Channel Attention Model, and the word vector is randomly initialized.
- DCAM (Word2Vec): This is the model proposed in this paper with pre-trained vectors from Word2Vec.

4. Experimental Results and Discussion

Accuracy and *F-SCORE* are used as model evaluation indicators. *F-SCORE* is calculated by Recall and Precision. The formula is shown as Equations (19) to (21):

$$Precision = \frac{\sum_{c_i \in c} True(c_i)}{\sum_{c_i \in c} Doc(c_i)} \quad (19)$$

$$Recall = \frac{\sum_{c_i \in c} True(c_i)}{\sum_{c_i \in c} Response(c_i)} \quad (20)$$

$$F-SCORE = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% \quad (21)$$

Where $True(c_i)$ is the number of samples whose classification is c_i and the classification is correct, $Doc(c_i)$ is the number of all samples classified as c_i in the experiment, and $Response(c_i)$ is the number of texts actually classified as c_i in the text.

The experimental results are shown in Table 1. It can be concluded from Table 1 that the accuracy and *F-SCORE* of the DCAM model proposed in this paper are higher than other network models, which proves the feasibility and superiority of

the model in the text sentiment analysis task.

Table 1. Classification results of different models

Model	Accuracy	F-SCORE
CNN	0.863	0.860
LSTM	0.869	0.875
CNN-Attention	0.893	0.892
LSTM-Attention	0.897	0.895
DC-CNN-LSTM	0.913	0.902
DCAM (rand)	0.900	0.896
DCAM (Word2Vec)	0.927	0.922

Comparing CNN with the CNN-Attention model and LSTM with the LSTM-Attention model, it is found that the introduction of attention mechanism in the model can effectively improve the accuracy of text classification. This is because the attention layer can extract important features of the text and ignore the unimportant features, so that the classification results are affected by irrelevant words as little as possible, thereby improving the accuracy.

Comparing CNN and LSTM with DC-CNN-LSTM models, it is found that the dual-channel network structure improves the effect of text classification to some extent. This is because the single-channel network cannot extract features fully, and the model proposed in this paper not only uses CNN to learn local feature information but also uses LSTM to learn the sequence features of text.

Comparing the DCAM (rand) with DCAM (Word2Vec) models, it is found that the use of pre-trained Chinese word vectors is higher than that of random initialization, because more primitive data features are contained in the word vectors trained by the Word2Vec tool.

To show the change trend of the accuracy of different models, the accuracy of each iteration is plotted as a line graph as shown in Figure 4. The F-SCORE value is plotted as a line graph as shown in Figure 5. It can be concluded from the figures that the DCAM model has higher accuracy and better stability. The accuracy reaches the highest when the number of iterations is 13. It means that the model has converged and can stop training.

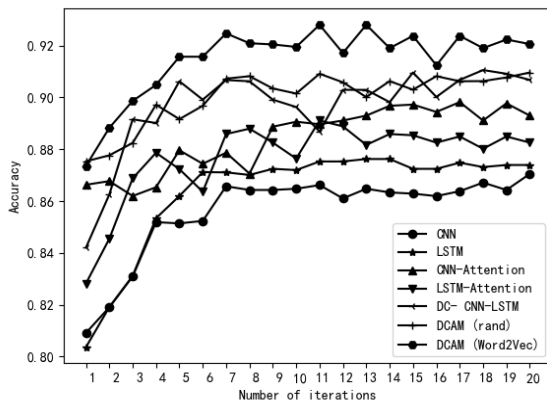


Figure 4. Change of accuracy of different models

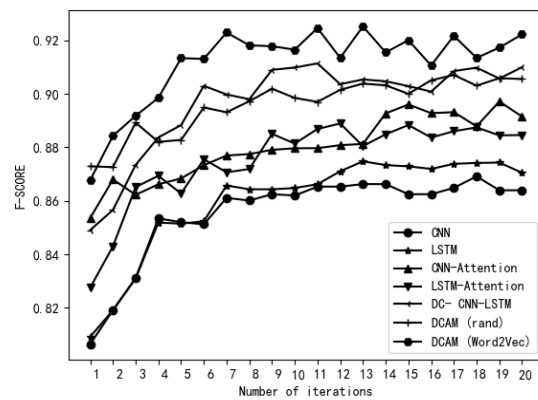


Figure 5. Variations of different models of F-SCORE

5. Conclusions

In this paper, we propose a Dual-Channel Attention Model to overcome the shortcomings of traditional neural networks. We propose to use the word vector trained by Word2Vec as the input feature of the CNN and LSTM networks and introduce the attention layer afterwards. Then, the features extracted by the two channels are merged. We evaluate the model with real shopping reviews from the Internet. The experimental results prove the feasibility of the proposed model and confirm the superiority of combining CNN and LSTM. In addition, our results also prove that pre-trained word vectors are significant features in text sentiment classification. Due to the complexity of the neural network structure, we will continue to study the model, optimize the network structure, and improve its application to text sentiment analysis tasks.

Acknowledgements

The project was supported by the Henan Provincial Department of Education Science and Technology Research Key Project (No. 13A510330) and the Basic and Advanced Technology Research Project of Henan Province (No. 152300410103).

References

1. F. T. I. I. Retrieval, "Opinion Mining and Sentiment Analysis," *Foundations & Trends in Information Retrieval*, Vol. 2, pp. 1-135, 2008
2. M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based Methods for Sentiment Analysis," *Computational Linguistics*, Vol. 37, pp. 267-307, 2011
3. S. Liu and W. Deng, "Very Deep Convolutional Neural Network based Image Classification using Small Training Sample Size," in *Proceedings of the 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pp. 730-734, 2016
4. A. Graves, A. R. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 38, pp. 6645-6649, 2013
5. R. Yin, P. Li, and B. Wang, "Sentiment Lexical-Augmented Convolutional Neural Networks for Sentiment Analysis," in *Proceedings of IEEE Second International Conference on Data Science in Cyberspace*, pp. 630-635, 2017
6. Y. Zhang, S. Roller, and B. Wallace, "MGNC-CNN: A Simple Approach to Exploiting Multiple Word Embeddings for Sentence Classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016
7. N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," *Eprint Arxiv*, Vol. 1, 2014
8. Y. Zhang, Y. Jiang, and Y. Tong, "Study of Sentiment Classification for Chinese Microblog based on Recurrent Neural Network," *Chinese Journal of Electronics*, Vol. 25, pp. 601-607, 2016
9. A. Hassan and A. Mahmood, "Efficient Deep Learning Model for Text Classification based on Recurrent and Convolutional Layers," in *Proceedings of IEEE International Conference on Machine Learning and Applications*, 2018
10. A. Hassan and A. Mahmood, "Convolutional Recurrent Deep Learning Model for Sentence Classification," *IEEE Access*, Vol. 6, pp. 13949-13957, 2018
11. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, et al., "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," *Computer Science*, pp. 2048-2057, 2015
12. D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *Computer Science*, 2014
13. W. Yin, H. Schütze, B. Xiang, and B. Zhou, "ABCNN: Attention-based Convolutional Neural Network for Modeling Sentence Pairs," *Computer Science*, 2015
14. Q. H. Vo, H. T. Nguyen, B. Le, and M. L. Nguyen, "Multi-Channel LSTM-CNN Model for Vietnamese Sentiment Analysis," in *Proceedings of International Conference on Knowledge and Systems Engineering*, pp. 24-29, 2017
15. M. T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," *Computer Science*, pp. 1412-1421, 2015
16. Y. Bengio and O. Delalleau, "On the Expressive Power of Deep Architectures," in *Proceedings of International Conference on Algorithmic Learning Theory*, pp. 18-36, 2011
17. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Computer Science*, 2013

Hui Li received his doctorate degree in information and communication engineering in 2008 from Nanjing University of Science and Technology. He is currently working as a teacher at Henan Polytechnic University. His research interests are chaotic communications.

Yuanyuan Zheng is a graduate student in the School of Physics and Electronic Information at Henan Polytechnic University. Her research interests are natural language processing and deep learning.

Pengju Ren is an undergraduate student in the School of Physics and Electronic Information at Henan Polytechnic University. His research interests are image processing and data mining.