IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 30, 2018, accepted October 27, 2018, date of publication November 22, 2018,
date of current version December 27, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2882878

Combining Convolution Neural Network and Bidirectional Gated Recurrent Unit for Sentence Semantic Classification

DEJUN ZHANG[1], LONG TIAN[2], MINGBO HONG[2], FEI HAN[2], YAFENG REN[3], AND YILIN CHEN[4]

[1]Faculty of Information Engineering, China University of Geosciences, Wuhan 430074, China
[2]College of Information and Engineering, Sichuan Agricultural University, Ya'an 625014, China
[3]Guangdong Collaborative Innovation Center for Language Research and Services, Guangdong University of Foreign Studies, Guangzhou 510420, China
[4]School of Computer, Wuhan University, Wuhan 430072, China

Corresponding author: Dejun Zhang (zhangdejun@cug.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61702350 and Grant 61702121.

ABSTRACT Many keywords in a sentence that represents the semantic propensity of the sentence. These words can exist anywhere in the sentence, which poses a great challenge to sentence semantic classification. The current sentence semantic classification methods usually tackle this problem by the use of attention mechanism, and most of them utilize softmax function to calculate each word's weight. According to the observation that a word with higher score carries more valuable information in sentence modeling, this paper presents a novel low-complexity model termed as CNN-BiGRU by integrating both convolution neural network (CNN) and bidirectional gated recurrent unit (BiGRU). Both the contextual representations and the semantic distribution are obtained through BiGRU, and the latter is constrained to a Gaussian distribution. In addition, the proposed model utilizes a shallow word-level CNN to obtain intermediate representations, and the score of each word is denoted as the Euclidean distance between the intermediate representations and the semantic distribution. Then, the final representations are obtained by the combination of the contextual representations and the score of each word, and thus, the model learns a compact code for sentence sentiment classification and can be trained end-to-end with limited hyper-parameters. In conclusion, the proposed model is able to focus both the keywords and the underlying semantics of the words. Comprehensive experiments are conducted on seven benchmarks. Compared with the state-of-the-art models, our model has excellent performance.

INDEX TERMS Semantic distribution, sentence classification, natural language processing, convolution neural network, bidirectional gated recurrent unit.

I. INTRODUCTION

Deep learning models have achieved remarkable performance in computer vision, speech recognition and Natural Language Processing (NLP). Text classification is one of the most important fields of NLP and is the basis for many other applications, such as web searching, information filtering and sentiment analysis [1]–[3].

In the early stage, the Bag-of-Words has been commonly used for the feature expression which is really critical and important for text classification. As for feature selection, there are several methods, such as MI [4] and pLSA [5]. Traditional models often suffer from the data sparsity problem, which heavily affects the accuracy of classification. On the other hand, those models ignore the context. A word usually has several different meanings, and according to the context, the specific meaning of the word can be finalized. For instance, "*the spring can be used to weigh objects*." When only the word "*spring*" is analyzed, its specific meaning cannot be well understood. The word "*spring*" can denote a season of the year, besides it can also represent a mechanical device that stores energy and so on.[1] But when the context is analyzed, it is easy to understand what "*spring*" means here. Thus the meaning of the word can be understood through context.

As one of the most salient models in sentence modeling, Recurrent Neural Networks (RNN) [6] can capture the

[1] https://en.wikipedia.org/wiki/Spring

long-term dependency of sentences. However, the traditional RNN faces the following two main problems: (i) The keywords in sentences play a decisive role in expressing the meaning of a sentence. However, the traditional RNN ignores the keywords in sentences. (ii) In language expression, there are some statements that have certain complex structures. To better understand the semantics of words, it is necessary to read sentences in different directions (forward or backward). However, the traditional RNN ignores the differences in semantic expression in different directions.

Zhou *et al.* [7] proposed C-LSTM to solve the first problem, which combines the Convolution Neural Network (CNN) [8] with Long Short-Term Memory (LSTM) [9]. It takes advantage of CNN to extract a sequence of higher-level phrase representations and feeds them into LSTM to represent sentences. In addition, attention mechanism [10] is widely used in neural networks, and through attention mechanisms, models can find the words that contribute more semantic information to sentences. Both of them solve the first problem very well. To tackle the second problem, Lai *et al.* [11] proposed the Bidirectional Recurrent Neural Network (BiRNN) to capture the semantic information in different directions for sentence modeling, and achieved remarkable results.

Inspired by C-LSTM, the attention mechanism and BiRNN, in this paper, a novel model is proposed in which CNN is adopted for sentence modeling and obtaining the intermediate sentence representation. In contrast to Lai *et al.* [11], BiGRU is employed to learn the contextual information and obtain the context-based sentence representation. In addition, the semantic distribution of each sentence can be obtained through training. According to the semantic distribution, the model calculates the score for each word. The score directly affects the final representation of the sentence.

The proposed model has a few hyper-parameters that directly improve the computational efficiency. The experimental results demonstrate that the proposed model outperforms the state-of-the-art approaches on seven benchmark datasets. Furthermore, word scores in some sentences are visualized in section V-G, which further proves the proposed model can learn the appropriate representations. Our contributions can be summarized as follows:

- We find that the semantic distribution of each sentence follow a Gaussian distribution. The mean value indicates the overall semantic tendency of the sentence. In a Gaussian distribution, the closer the words are to the mean value, the greater the effect on the overall semantic expression of the sentence is.
- We stack CNN and BiGRU in a unified architecture for semantic sentence modeling so the model can extract the feature information of the sentence more accurately and improve the sentence classification accuracy.

The outline of the paper is as follows: Section II discusses related work. The framework of the model is introduced in Section III. The algorithms and datasets are given in Section IV. The experimental study is shown in Section V.

II. RELATED WORK

As one of the hot research directions in NLP, text classification plays an important role in the era of big data. As branch tasks of text classification, outstanding achievements have been made in emotional information classification and spam classification in recent years. Continuous Bag-of-words (cBoW) is widely used in traditional text classification. Such models perform well on a variety tasks, but there is a very serious drawback: they lose the order information of the words, which is critical for semantic analysis. In addition, more complex features have been designed, such as part-of-speech tags, noun phrases [12] and tree kernels [13]. However, the performance of these models is less than satisfactory because of data sparsity.

In recent years, neural network has been increasingly used in NLP, and many models have achieved good performance in sentence classification using neural network. Moreover, neural network has led to new ideas for capturing the word order and solving the data sparsity problem. Kim [14] proposed a CNN with pre-trained word vectors for sentence classification, which uses CNN to construct non-linear interactions between words so they can learn sentence representations and can capture the semantics well. Although it achieved good results in sentence classification, it did not consider the meaning of the word in its context.

Socher *et al.* [15] proposed the Recursive Neural Network (RecursiveNN), which has been proven to be efficient in terms of constructing sentence representations. The semantics of sentences are captured by the RecursiveNN through the tree structure, its performance depends strongly on the analysis of textual tree structure. The composition procedure is recursively applied to child nodes in the parse tree in a bottom-up manner to generate the hidden representations of parent nodes until reaching the root of the tree. However, the tree construction can be very time-consuming, and constructing such a textual tree exhibits has a time complexity of at least $O(n^2)$, where n is the length of the sentence. Therefore, RecursiveNN is not suitable for long sentences.

Another model, which only exhibits a time complexity of $O(n)$, called the Recurrent Neural Network (RNN) [6] is special cases of the RecursiveNN. It can capture the long-term dependencies and learn the meaning of words from their context. The RNN can handle variable-length sequences with the memorys state, in which each time-steps output depends on the value at the previous time. It has received much attention due to its outstanding ability to save a sequence of information over time. However, one of the serious problems is that the gradient vanishing.

Hochreiter and Schmidhuber [9] proposed LSTM, which can overcome the problem of gradient vanishing through the gating mechanism and target information is automatically taken into account. The Gated Recurrent Unit (GRU) [16] is a variant of LSTM and has similar properties, including the gate of activation, candidate activation, update and reset. Chung *et al.* [17] compared the performance between GRU and LSTM. From their conclusion, GRU requires fewer

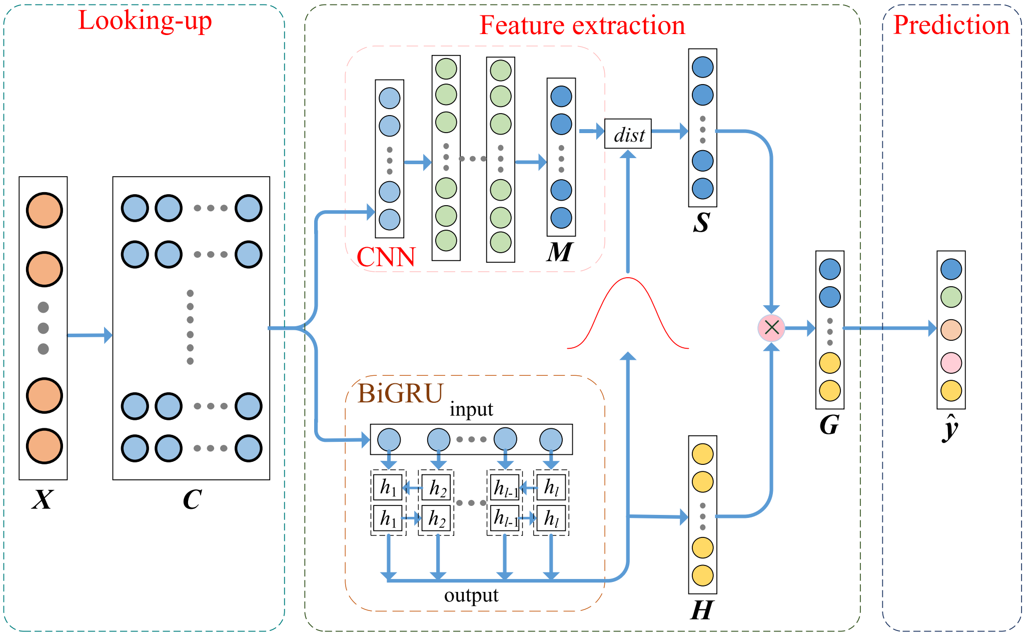


FIGURE 1. Architecture of CNN-BiGRU. The goal of the CNN-BiGRU is to learn the expression of sentences and use the expressions to achieve the classification of sentences. The input of the model is a sentence ***X*** through the looking-up layer and then get the word-embedding ***C***. ***M*** denotes the intermediate representation, while ***H*** denotes context representation. They are obtained from CNN and Bi-GRU respectively. ***S*** is vector in which each element denotes the score of the corresponding word. Combining ***S*** and ***H*** the final representation of the sentence can be obtained, through the fully connected layer, the model outputs the predicted label $\hat{y}$.

hyper-parameters than LSTM does, so GRU converges significantly faster in most tasks.

Lee and Dernoncourt [18] proposed sequential short-text classification, which utilized RNN and CNN to generate word embeddings, and then classified the short texts through a fully connected layer. Nevertheless, this method takes a lot of time to train word embeddings.

Pennington *et al.* [19] proposed Global Vectors for Word Representation (GloVe), which trains the model in a word-word co-occurrence matrix. The co-occurrence probabilities of words can reflect the correlation between the words, and this relationship can be described by a word-word co-occurrence matrix. In our model, pre-trained word vectors are adopted. Replacing the words in the text with pre-trained word vectors to form a text matrix, helps the model perform better and take less time.

III. THE MODEL DESCRIPTION

A. THE OVERALL FRAMEWORK

The model's architecture is shown in Fig. 1. The input of the model is a pre-trained word vector, and then it feeds the word embeddings to CNN and BiGRU. CNN is adopted to filter out the noise and extract the features of each word. By using BiGRU, it can fully consider the context of the text and learn the context distribution of sentences. Their derails are described in the section III-C and section III-D respectively. The final sentence representations are passed to a fully connected layer whose outputs are the probability distributions over labels.

B. MODEL INPUT

For the proposed model, the input is a sentence. Each word in the sentence is replaced by the corresponding word vector, and it then forms a word embedding. Without loss of generality, let $\boldsymbol{x}_i$ denotes the i-th word in the sentence and $\boldsymbol{X}$ represents the input sentence. Let $\boldsymbol{c}_i \in R^d$ be the d-dimensional word vectors for the word $\boldsymbol{x}_i$, and $\boldsymbol{C} \in R^{l \times d}$ represents the word-embedding matrix, where l is the *maxlen* of the sentence, and *maxlen* represents the length of padding. The specific setting of the *maxlen* of the sentence and the way to initialize the word vector are described in sections IV-B and IV-C, respectively.

C. INTERMEDIATE REPRESENTATION

CNN is a feed-forward neural network. It can encode the important information contained in the input data with far fewer parameters than are used in other deep learning frameworks. In computer vision, multiple convolutional layers are frequently used to achieve good performance. However, as the number of layers increases, so does the amount of calculations. Yoon Kim [14] only used one convolutional layer for sentence classification and achieved remarkable results. In this paper, we also use only one convolution layer, as shown in Fig. 2.

Without the loss of generality, let $\boldsymbol{c}_i$ indicates the i-th word corresponding to the word vector in the sentence. $\boldsymbol{m}_i$ denotes the characteristics of the word $\boldsymbol{c}_i$ extracted by CNN:

$$\boldsymbol{m}_i = f(w \cdot \boldsymbol{c}_i + b), \tag{1}$$

where b is a bias term and f denotes a non-linear function.

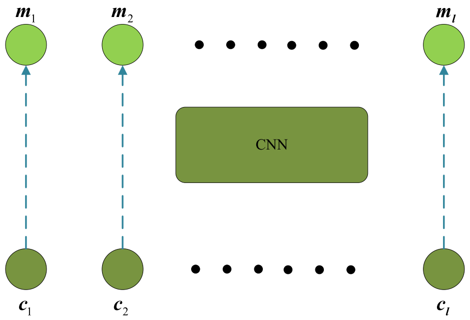


FIGURE 2. Illustration of Convolution Neural Network. The window size of the filter is 1 and weights are shared across different windows.

Let $\boldsymbol{M}$ denotes intermediate sentence representation, where $\boldsymbol{M} = [\boldsymbol{m}_1, \ldots, \boldsymbol{m}_l]$.

D. CONTEXTUAL REPRESENTATION

As a variant of LSTM, GRU has fewer hyper-parameters and effectively solves the problem of the vanishing gradient in RNN. Moreover, BiGRU is widely used in many fields such as text analysis, especially in the context semantics [20]. There are two gates in GRU: an update gate z and a reset gate $\boldsymbol{r}$. They modulate whether information is updated or forgotten, as shown in Fig. 3. To be specific, the update gate determines how many memories in the previous cell can survive, and the reset gate determines how to combine the new cell with the previous memory.

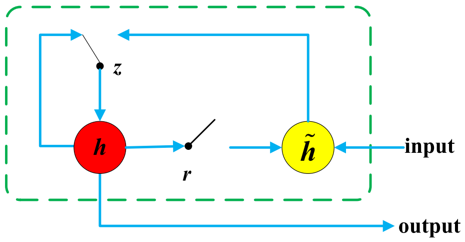


FIGURE 3. Illustration of the Gated Recurrent Unit, which is the basis of Bidirectional Gated Recurrent Unit. There are four very important parts in GRU: reset gates ***r***, update gates ***z***, activation ***h***, and candidate activation ***h̃***.

The activation of h_t^j of the GRU at time t is a linear interpolation between the previous activation h_{t-1}^j and the candidate activation $\tilde{h}_t^j$, which can be computed by

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_i^j \tilde{h}_t^j, \tag{2}$$

note that, how much the cell updates its activation is determined by the update gate z_t^j, which is computed by

$$z_t^j = \sigma(W_z \boldsymbol{x}_t + U_z h_{t-1}), \tag{3}$$

where σ is a nonlinear function such as logistic sigmoid function; $\boldsymbol{x}_t$ denotes a vector of the sequences at time step t, W_z and U_z are weights that can be trained to update z_t^j.

Similar to the traditional recurrent cell, the candidate activation $\tilde{h}_t^j$ can be computed by

$$\tilde{h}_t^j = \tanh(W\boldsymbol{x}_t + U(r_t \odot h_{t-1})), \tag{4}$$

where r_t denotes a set of reset gates and $\odot$ is an element-wise multiplication. The reset gate is portrayed as the decision maker, which means that it determines how many of the previous states h_{t-1} can survive. When the reset gate is zero, $\tilde{h}_t^j$ forgets all previous states. Following the update gate, the reset gate r_t can be computed by

$$r_t = \sigma(W_r \boldsymbol{x}_t + U_r h_{t-1}). \tag{5}$$

BiGRU includes two sequences: one forward and one backward. To fully consider the expression of information in different directions, we use element-wise summation to combine forward and backward sequences:

$$\boldsymbol{h}_i = \boldsymbol{h}_i^F \oplus \boldsymbol{h}_i^B, \tag{6}$$

where $\boldsymbol{h}_i^F$ represents the forward sequence, while $\boldsymbol{h}_i^B$ denotes the backward sequence.

In a nutshell, the word-embedding $\boldsymbol{C}$ passes through the BiGRU output μ and $\boldsymbol{H}$. Where μ denotes the mean value of the sentence of the semantic distribution, and $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_l]$ denotes the context sentence representation. The semantic information distribution of the sentence is constrained to a Gaussian distribution [21], where the closer the word is to the mean value, the greater the semantic contribution to the sentence is, and vice versa.

E. FINAL REPRESENTATION

Inspired by the attention mechanism [10], our model employs the word-to-mean Euclidean distance to describe the importance scores of words. Specifically, the word's score reflects the semantic contribution of the word to the sentence. The higher the score is, the more it can influence the semantic expression of the sentence. The score of each word is computed by

$$s_i = sim < \boldsymbol{m}_i, \mu >, \tag{7}$$

where s_i denotes the score of the i-th word in the sentence. $sim\!<,>$ is a function used to calculate the Euclidean distance. We multiply s_i and $\boldsymbol{h}_i$ to get the expression of the word in the sentence:

$$\boldsymbol{g}_i = \boldsymbol{h}_i \odot s_i. \tag{8}$$

The final representation is expressed as $\boldsymbol{G} = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_l]$, which contains both the meaning of a single word and the semantic information of the word in the context. Then, the final representation is fed into the fully connected layer. The fully connected layer is followed by a softmax non-linear layer that predicts the probability distribution over classes. The parameters of the network are trained to minimize the cross-entropy of the predicted and true distributions. The entire learning algorithm of CNN-BiGRU is summarized as Algorithm 1.

Algorithm 1 Pseudo-Code for Training

```
Input: $X$
Output: $\hat{y}$
$\boldsymbol{C} \leftarrow \boldsymbol{X}$; //Obtaining $\boldsymbol{C}$ based on $\boldsymbol{X}$.
for $i$ in [1, epoch] do
    $\boldsymbol{M} \leftarrow \boldsymbol{C}$; //Extracting $\boldsymbol{M}$ through CNN.
    $\boldsymbol{H}, \mu \leftarrow \boldsymbol{C}$; //Obtaining $\boldsymbol{H}$ and $\mu$ through BiGRU.
    $\boldsymbol{S} \leftarrow sim\hat{I}ij < \boldsymbol{M}, \mu >$; //Calculating the score.
    $\boldsymbol{G} \leftarrow \mathbf{H} \odot \boldsymbol{S}$; //Obtaining the final representation.
    $\hat{y} \leftarrow \boldsymbol{G}$; //Feeding $\boldsymbol{G}$ into fully connected layer.
    Minimize cross entropy between $y$ and $\hat{y}$ with the
Adadelta update rule.
end for
```

IV. TRAINING PROCEDURE AND DATASETS

A. ALGORITHM

First, we remove all the unused symbols, including punctuation, and replace the words in the sentence with pre-trained word vectors to form a text matrix. Then, we feeding the text matrix to CNN and BiGRU, respectively. The noise can be filtered out through CNN, word features can be learned, and the intermediate representation $\boldsymbol{M}$ can be obtained. Further, the context information $\boldsymbol{H}$ and the mean value μ can be obtained through BiGRU.

After that, according to μ, the sentence semantic is constrained to a Gaussian distribution. By calculating the Euclidean distance of each word to the mean value. The length of the distance indicates how much the word contributes to the overall semantics. The shorter the distance is, the higher words score is. We combining the score of each word with the contextual information $\boldsymbol{H}$ and finally get the representation of the sentence, which is expressed by $\boldsymbol{G}$.

Next, we feed the final representation $\boldsymbol{G}$ into the fully connected layer, and obtain the predicted label $\hat{y}$ Then, we minimize the cross entropy between the given label y and the predicted label $\hat{y}$, and all parameters are updated with the Adadelta [22] update rule.

B. PADDING

The input of the algorithm is N variable-length sentences. The lengths of the sentences in each dataset are analyzed, a *maxlen* is set for each dataset, and padding operation is adopted. When the lengths of sentences are shorter than the *maxlen*, the end of them are padded with zeros, which makes them equal in length. For sentences longer than *maxlen*, the ends of the sentences need to be subtracted, so the sentences lengths are equal to *maxlen*.

C. WORD VECTOR INITIALIZATION

Each sentence is constituted by words that are represented by vectors. One-hot vectors were widely used in traditional word representation, and they performed well in the document classification task [23]. However, the problem of data sparseness becomes prominent when the sentence length increases. Currently, Word2vec [24] and GloVe [19] are the two most widely used pre-trained word embedding matrices, although they may result in slightly different performances. In this paper, GloVe is employed and it has been trained by Pennington [19] on 6 billion tokens of Wikipedia 2014 and Gigaword 5. GloVe can be a good display of the relationship between words and can improve the efficiency and accuracy of the model. Replacing the word with the corresponding word vector. For words that are not in the corpus, they are randomly initialized with a uniform distribution [-0.1, 0.1]. There are four different dimensions of GloVe, and we perform an experimental analysis of the effects of the different dimensions on the results of the model in section V-F2.

D. DATASETS

The model is tested on various benchmarks. The summary statistics of these datasets are in Table 1. The first column is for all data names, the second column is the total number of categories included in the data, the third column is the average length of the data, the fifth column is the size of the datasets, and the last column represents test set size(CV means there was no standard train/test split and thus 10-fold CV was used).

- MR: This dataset is about using one sentence to comment on the movie, and the task is to determine the emotional polarity of sentence. Using the sentence polarity from [25], there are 5,331 positive and 5,331 negative reviews in the dataset.[2]
- Subj [26]: The task of the dataset's subjectivity is to classify a sentence as being subjective or objective. Roughly speaking, Subj contains 5,000 subjective sentences and 5,000 objective sentences.
- CR [27]: This contains customer evaluations of goods (cameras, MP3s etc.). The dataset is divided into

[2] http://www.cs.cornell.edu/people/pabo/movie-review-data

TABLE 1. Characteristics of datasets.

Dataset	Number of target classes	Average sentence length	Maximum sentence length	Dataset size	Test set size
MR [25]	2	20	56	10,662	CV
Subj [26]	2	23	120	10,000	CV
CR [27]	2	19	105	3784	CV
MPQA [28]	2	3	36	10,606	CV
SST-1 [15]	5	18	53	11,855	2210
SST-2	2	19	53	9618	1821
TREC [29]	6	10	37	5952	500

two types: positive and negative, task is to predict reviews.[3]

- MPQA [28]: This is the opinion polarity subtask of the MPQA dataset.[4]
- SST-1: The Stanford Sentiment Treebank is an extension of MR, but it was re-labeled by [15], which means that it has fine-grained labels (very positive, positive, neutral, negative and very negative).[5]
- SST-2: It is the same as STT-1 but the neutral comments and binary labels are removed.
- TREC [29]: It is a dataset for question classification. It divides questions into 6 types (whether the question is about person, location, numeric information, etc.)[6]

V. EXPERIMENTAL RESULTS AND COMPARISON

A. REGULARIZATION

In some cases, the performance only increases marginally or even decreases because of over-fitting. In this paper, two effective technologies are employed: dropout and L2 weight regularization [30]. Dropout is a regularization technique that is commonly used to reduce over-fitting by preventing complex co-adaptation on training data. It is a very efficient way of performing model averaging for neural networks. Dropout works when all words have been replaced by the corresponding word vector, and L2 regularization is applied to the weight of the softmax layer. Both dropout and L2 regularization aim to alleviate over-fitting.

B. PARAMETER SETTING

The *maxlen* of MR, CR, MPQA and TREC are set to 30, and the *maxlen* of Subj, SST1 and SST2 are set to 60. For all tasks, the dimension of each word vector is 300, and the word vectors are fine-tuned during the training phase. The hidden layer for CNN and BiGRU is set to 300, and the output dimension of CNN and BiGRU are set the same. The training batch size for SST-1, SST-2 and TREC are set at 128, and those for the other datasets are set as 64. All experiments are conducted with 500 epochs. The learning rates for all datasets are set as 1.0. Training is done through the stochastic gradient descent over shuffled mini-batches with the Adadelta [22] updating rule. Adadelta gives similar results to Adagrad [31], but it requires fewer epochs. The model is developed based on tensorflow and keras. To benefit from the efficiency of the parallel computation of the tensors, all simulation studies are conducted with a NVIDIA 1050 GPU on a Windows PC.

C. CLASSIFICATION ACCURACY COMPARISON

The performance of the proposed method is compared with different basic baselines and state-of-the-art neural sentence models. The classification accuracy of CNN-BiGRU compared with other approaches is shown in Table 2. The spaces lacking scores indicate the model was not evaluated on this dataset.

- CNN-rand: Instead of using pre-trained word vectors, it randomly initializes all words and modifies the word vectors during training.
- CNN-static: Pre-trained word vectors are used to replace the word with the corresponding word vectors, and words that are not in the corpus are initialized randomly.
- NB-SVM (Naive Bayes Support Vector Machines) and MNB (Multinomial Naive Bayes): They input features into the SVM classifier and the Naive Bayes classifier, respectively [32]. Neither uses pre-trained word vectors.
- cBoW (continuous Bag-of-Words) [33]: This model uses the average or max pooling to compose a set of word vectors into a sentence representation.
- RAE (Recursive AutoEncoder) [34]: RAE relies mainly on recursive self-encoding to implement sentence classification and uses pre-trained word vectors.

[3] http://www.cs.uic.edu/liub/FBS/sentiment-analysis.html

[4] http://www.cs.pitt.edu/mpqa/

[5] http://nlp.stanford.edu/sentiment/

[6] http://cogcomp.cs.illinois.edu/Data/QA/QC/

TABLE 2. Comparison with the accuracy of other approaches on benchmark datasets.

approach	MR	Subj	CR	MPQA	SST-1	SST-2	TREC
CNN-rand	76.1	89.6	79.8	83.4	45.0	82.7	91.2
CNN-static	81.0	93.0	84.7	89.6	45.5	87.2	92.8
NB-SVM [32]	79.4	93.2	81.8	86.3	-	-	-
MNB [32]	79.0	93.6	80.0	86.3	-	-	-
cBoW [33]	77.2	91.3	79.9	86.4	42.8	81.5	87.3
RAE [34]	77.7	-	-	86.4	43.2	82.4	-
MV-RNN [35]	79.0	-	-	-	44.4	82.9	-
RNTN [15]	-	-	-	-	45.7	85.4	-
RNN [6]	77.2	92.7	82.3	90.1	47.2	85.8	90.2
Bi-RNN [36]	81.6	93.2	82.6	90.3	48.1	86.5	91.0
DCNN [37]	-	-	-	-	48.5	86.8	93.0
P.V. [38]	74.8	90.5	78.1	74.2	48.7	87.8	91.8
C-LSTM [7]	-	-	-	-	49.2	87.8	-
ATT-CNN [39]	-	-	-	-	49.2	-	89.8
CNN+RNN attention [40]	79.0	93.2	-	-	48.0	86.1	-
CRAN [40]	82.0	93.8	-	-	48.1	86.9	-
CNN+BiGRU	78.0	91.4	90.2	89.2	44.3	86.3	94.1
BiGRU+CNN	78.3	92.5	91.0	78.3	42.2	85.4	93.5
CNN-BiGRU	**79.4**	**93.8**	**93.4**	**90.6**	**49.3**	**87.5**	**95.1**

- MV-RNN (Matrix-Vector Recursive Neural Network) [35]: It uses RNN and parse trees to realize the classification of sentences.
- RNTN (Recursive Neural Tensor Network) [15]: This model belongs to recursive neural network and recursively compose word vectors into sentence vector along a parse tree. Every word in the parse tree is represented by a tensor-based feature function.
- RNN (Recurrent Neural Network) [6]: The RNN composes words in a sequence from the beginning to the end into a final sentence vector.
- Bi-RNN (Bidirectional Recurrent Neural Network) [36]: It is a variant of RNN. It performs composition from both beginning to end and from end to beginning.
- DCNN (Dynamic Convolutional Neural Network) [37]: It uses the pre-trained word2vec and K-max pooling.
- P.V. (Paragraph Vector) [38]: It is the logistic regression on top of the paragraph vectors. It learns sentence or paragraph representations by learning word vectors using cBOW and skip-gram.
- C-LSTM [7]: C-LSTM learns the sentence representation by combining CNN and LSTM, and uses the learned sentence representation to achieve text classification.
- ATT-CNN (Attention based CNN) [39]: A convolutional neural network with attention mechanism is utilized to improve the performance of sentence classification. The use of attention-based CNN is able to capture long term contextual information for each word without any external features.
- LSTM+RNN attention [40]: It combines the CNN and RNN attention mechanisms to achieve sentence-level classification.
- CRAN (Convolutional Recurrent Attention Network) [40]: This model utilizes convolution operation to capture the attention signals, each signal representing local information of a word in its context; then RNN is used to model text with attention signals.
- CNN+BiGRU: It is a pipelining structure, that is, the output from the CNN is fed into the input of BiGRU, and the learned sentence representation is used to achieve sentence classification.
- BiGRU+CNN: It is also a pipelining structure. Contrary to CNN+BiGRU, the output from the BiGRU is fed into the input of CNN.

D. COMPARISON SYSTEMS

By comparing CNN-rand with CNN-static, it can be found that when using pre-trained word vectors, the model can have a better performance. We have tried to randomly initialize word vectors, as CNN-rand does, but they did not perform well on our model.

NB-SVM and MNB are based on two Bayesian models [32]. When the datasets have long sentences, the model performs well. However, they do not perform well on datasets with short sentences. This results from the sparsity of n-gram encoding for short sentences.

The performance of cBoW [33] is not good because it loses word order information in the sentences; this information is very important for the expression of the semantics.

RAE [34], MV-RNN [35] and RNTN [15] are recursive neural network structures. It can be seen that the accuracy of these models are not satisfactory. Their accuracies largely depend on the construction of the parse trees. The models are so complex that they easily overfitting.

Compared with the experimental results of the RNN [6] and BiRNN [36] models, it can be easily found that BiRNN has better results because it can understand sentence semantics from different directions. The bidirectional structure helps BiRNN extract more information and enhance the learning performance.

DCNN [37] uses multiple layers and K-max pooling to achieve sentence classification. Compared with methods that use only one-layer CNN, the accuracy of DCNN is not greatly improved. This result shows that using only one-layer is sufficient to extract complex information.

C-LSTM [7] is a really novel model that fully combines the advantages of CNN and LSTM and achieves state-of-the-art results.

ATT-CNN [39] utilizes attention mechanism to automatically capture long term contextual information and correlations among non-consecutive words. It achieves competitive performance as [7] without any external syntactic information.

CRAN [40] utilizes convolution operation to capture attention signals. It combines RNN with attention signals to model text, and finally achieves text classification.

Compared with previous works, the CNN-BiGRU is a parallel structure, which takes advantage of word-level CNN to extract the characteristics of each word. The model utilizes BiGRU to obtain the contextual information and the semantic distribution of the sentence, and the score of each word is calculated by Eq. (7) which directly affects the final feature expression. During the experiments, the higher the score, the greater the semantic contribution of the word to the sentence. Even with long sentences, our model can still pay close attention to each keyword. As a consequence, once the sentence contains more keywords (such as positive or negative emotional vocabulary) or has a longer length, our model will achieve better performance.

There are many different ways to combine CNN and BiGRU, such as the pipelining structure. Both the CNN+BiGRU and the BiGRU+CNN are pipelining structures. As shown in Table 2, both of them achieve unsatisfactory results, which is caused in the case of gradient explosion or vanishing when performing forward or back propagation, thus making the model unable to effectively learn feature information and leading the result unsatisfactory. As a contrast, our CNN-BiGRU can not only extract key information effectively, but can also significantly improve the accuracy of sentence classification.

E. STATISTICAL TEST

The proposed model has several randomization and initialization steps, therefore, we use paired comparison t-test to validate the performance improvement of the CNN-BiGRU over other methods. Specifically, the paired t-test is used to compare two different methods of measurement when the measurement is applied to the same subject.

Firstly, we set the null hypothesis which is denoted as the mean difference between two paired methods as zero. Then, the difference of each time for each method-pair is calculated. The t-statistic is defined as follows:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}}, \tag{9}$$

where $\bar{d}$ denotes the mean difference, s_d denotes standard deviation of differences and n is the number of pairs of observation, which is 10 in our case.

Under the null hypothesis, the t-statistic follows a t-distribution with n-1 degree of freedom. The p-value of the paired t-test can be found using a table of values from Student's t-distribution. The p-value indicates whether the differences of measurements of the two methods are statistically significant. If the calculated p-value is below the threshold chosen for statistical significance, then the null hypothesis is rejected in favor of the alternative hypothesis.

We find that the correlation p-value for each method-pair on all datasets are less than 0.01 (the complete p-value results are not shown). Therefore, compared to the baseline, we can conclude that our method yields better performance.

F. SENSITIVITY ANALYSIS

There are two very important parameters in the model. One is the dropout rate, and the other is the word embedding size. The effects of different parameters on the models results are analyzed in this section. When analyzing the effect of one parameter, the other parameters are held constant at their basic configuration values.

1) EFFECT OF DROPOUT RATIO

To mitigate over-fitting, dropout is widely used. Different ratios affect the precision of the model, as shown in Fig. 4. It can be found that in some datasets, the dropout ratio has great impacts. Further, when the dropout ratio is extremely large or small, the performance of the model is not satisfactory. However, when the dropout ratio is in the range of 0.4 to 0.6, the model performs well. Based on the above analysis of the results, we have reason to believe that dropout will be more helpful when the model is more complex.

2) EFFECT OF EMBEDDING SIZE

In this paper, GloVe is adopted, GloVe has 4 different dimensions: 50, 100, 200 and 300. To compare the effects of different dimensions on the performance of the model, a comparative experiment is conducted. As the results shown in Fig. 5, with the increasing of the size of word embedding, the expression of each word is getting more extensive. Our model performs best when the dimension is 300.

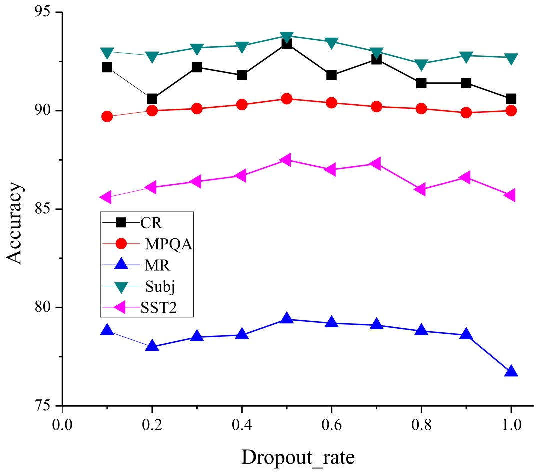


FIGURE 4. Illustration of the effects of different dropout ratios.

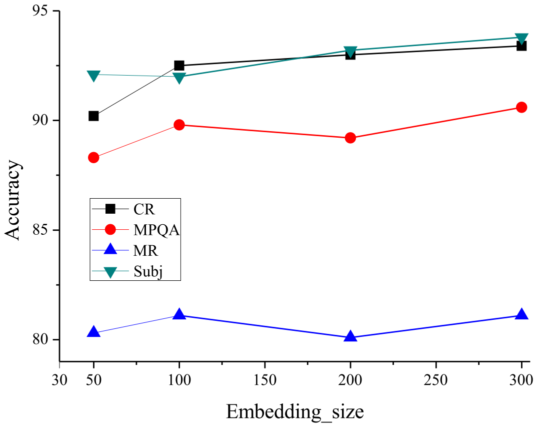


FIGURE 5. Illustration of effects of word embedding dimension.

G. VISUALIZATION OF WORDS SCORE

To further demonstrate the effectiveness of the model, several sentences from MR are selected for visualization analysis, as shown in Fig. 6. The darker the color is, the higher the score is, and the greater the semantic contribution to the sentence is. The first three sentences are positively expressed in red, and the last three sentences are negatively expressed in blue.

In the first three sentences, it can be found that the words "*enjoyable*", "*heartfelt*", "*comedy*" and so on have deeper colors, which means that they contribute more to the semantics of sentences. They are more able to express the positive emotions of the sentences. This is also in line with the actual situation. In the last three sentences, it can be observed that the colors of the "*truth*", "*brazenly*", "*misguided*", "*tiresome*" and so on are deeper than the others. The deeper colors imply that they are more important than the other words.

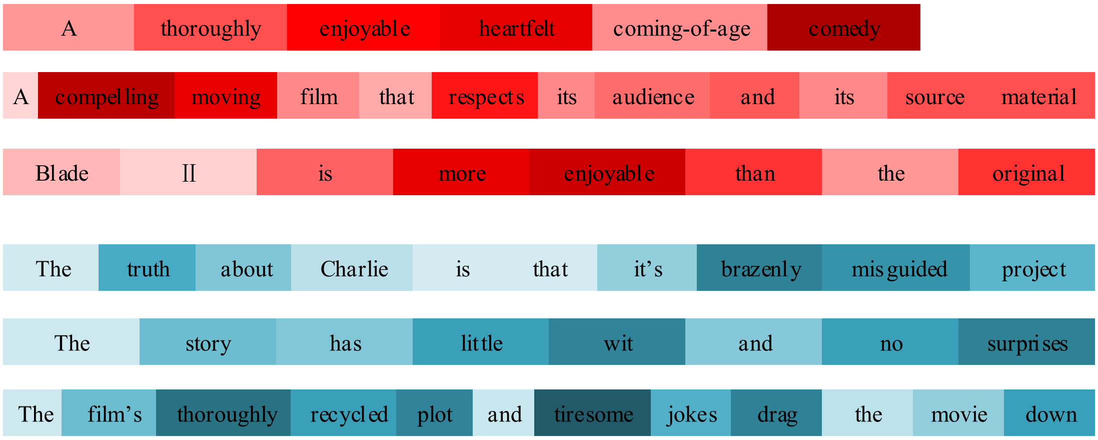


FIGURE 6. Illustration of visualization of the scores. They are selected from the Movie Review dataset. The first three sentences are labeled as positive in the dataset, while the last three sentences are labeled as negative. They are represented using different colors.

VI. CONCLUSION

In this paper, a novel model is proposed termed as CNN-BiGRU. The model innovatively integrates convolution neural network and bidirectional gated recurrent unit into CNN-BiGRU for sentence modeling. The use of convolution neural network can extract the semantic information of each word, while the use of Bidirectional Gated recurrent Unit (BiGRU) can extract the semantic information of words in the context. Furthermore, the combination of convolution neural network with the BiGRU enables the model to extract comprehensive information. In addition, the use of the word-to-mean Euclidean distance to represent the score of each word helps the model notice words that contribute significantly to the semantic information of sentences.

The proposed model can extract the comprehensive and significant information contained in a sentence with limited hyper-parameters. The final sentence representation is powerful, which makes the full-connected predict label very accurate. On seven benchmark datasets for sentence classification, our model achieves state-of-the-art results.

In this paper, we integrate CNN and BiGRU to get different expressions of sentences. For future work, we will consider other ways to get sentence expressions. In addition, how to reduce the amount of parameters to furtherly improve the calculation efficiency are also our future priorities.

ACKNOWLEDGMENT

(Dejun Zhang and Long Tian contributed equally to this work.)

REFERENCES

[1] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," in *Mining Text Data*. Boston, MA, USA: Springer, 2012, pp. 163–222.

[2] X. Liu, Q. Xu, and N. Wang, "A survey on deep neural network-based image captioning," *Vis. Comput.*, to be published, doi: 10.1007/s00371-018-1566-y.

[3] J. E. Meng, Y. Zhang, N. Wang, and P. Mahardhika, "Attention pooling-based convolutional neural network for sentence modelling," *Inf. Sci.*, vol. 373, pp. 388–403, Dec. 2016.

[4] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.

[5] L. Cai and T. Hofmann, "Text categorization by boosting automatically extracted concepts," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2003, pp. 182–189.

[6] K.-I. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural Netw.*, vol. 6, no. 6, pp. 801–806, 1993.

[7] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau. (2015). "A C-LSTM neural network for text classification." [Online]. Available: https://arxiv.org/abs/1511.08630

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489.

[11] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. AAAI*, vol. 333, 2015, pp. 2267–2273.

[12] D. D. Lewis, "An evaluation of phrasal and clustered representations on a text categorization task," in *Proc. 15th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 1992, pp. 37–50.

[13] M. Post and S. Bergsma, "Explicit and implicit syntactic features for text classification," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2013, pp. 866–872.

[14] Y. Kim. (2014). "Convolutional neural networks for sentence classification." [Online]. Available: https://arxiv.org/abs/1408.5882

[15] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1631–1642.

[16] K. Cho *et al.* (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation." [Online]. Available: https://arxiv.org/abs/1406.1078

[17] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. (2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling." [Online]. Available: https://arxiv.org/abs/1412.3555

[18] J. Y. Lee and F. Dernoncourt. (2016). "Sequential short-text classification with recurrent and convolutional neural networks." [Online]. Available: https://arxiv.org/abs/1603.03827

[19] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors forWord representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[20] A. Chakrabarty, O. A. Pandit, and U. Garain, "Context sensitive lemmatization using two successive bidirectional gated recurrent networks," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1481–1491.

[21] D. Zhang, F. He, S. Han, L. Zou, Y. Wu, and Y. Chen, "An efficient approach to directly compute the exact Hausdorff distance for 3D point sets," *Integr. Comput.-Aided Eng.*, vol. 24, no. 3, pp. 261–277, 2017.

[22] M. D. Zeiler. (2012). "ADADELTA: An adaptive learning rate method." [Online]. Available: https://arxiv.org/abs/1212.5701

[23] R. Johnson and T. Zhang. (2014). "Effective use of word order for text categorization with convolutional neural networks." [Online]. Available: https://arxiv.org/abs/1412.1058

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean. (2013). "Efficient estimation of word representations in vector space." [Online]. Available: https://arxiv.org/abs/1301.3781

[25] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics*, 2005, pp. 115–124.

[26] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. 42nd Meeting Assoc. Comput. Linguistics (ACL)*, 2004, pp. 271–278.

[27] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 168–177.

[28] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Lang. Resour. Eval.*, vol. 39, no. 2, pp. 165–210, May 2005.

[29] X. Li and D. Roth, "Learning question classifiers," in *Proc. 19th Int. Conf. Comput. Linguistics*, vol. 1, 2002, pp. 1–7.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[31] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.

[32] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2012, pp. 90–94.

[33] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[34] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 151–161.

[35] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, 2012, pp. 1201–1211.

[36] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[37] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. (2014). "A convolutional neural network for modelling sentences." [Online]. Available: https://arxiv.org/abs/1404.2188

[38] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.

[39] Z. Zhao and Y. Wu, "Attention-based convolutional neural networks for sentence classification," in *Proc. INTERSPEECH*, 2016, pp. 705–709.

[40] J. Du, L. Gui, R. Xu, and Y. He, "A convolutional attention model for text classification," in *Proc. Nat. CCF Conf. Natural Lang. Process. Chin. Comput.* Cham, Switzerland: Springer, 2017, pp. 183–195.

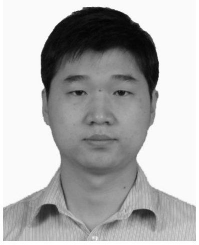
DEJUN ZHANG received the Ph.D. degree from the Department of Computer School, Wuhan University, China, in 2015. He is currently a Lecturer with the Faculty of Information Engineering, China University of Geosciences, China. He has conducted a considerable amount of research in digital geometric processing and computational photography. His research areas include computer graphics, computer-aided design, computer vision, and image and video processing. He has published over20 papers in journals and conferences. Since 2015, he has been serving as a Senior Member of the China Society for Industrial and Applied Mathematics (CSIAM). He is also a Committee Member of the geometric design and computing with the CSIAM.

LONG TIAN was born in Jianyang, China, 1994. He is currently a pursuing the degree with the College of Information and Engineering, Sichuan Agricultural University, China. He has been involved in scientific research for two years. He has authored three articles in referred conferences and proceedings in the areas of natural language processing. He is a member of the China Society for Industrial and Applied Mathematics.

MINGBO HONG was born in Quanzhou, China, 1997. He is currently pursuing the degree with the College of Information and Engineering, Sichuan Agricultural University, China. He has been involved in scientific research for two years. He has authored one article in referred journals and proceedings in the areas of natural language processing and computer graphics. He is a member of the China Society for Industrial and Applied Mathematics.

FEI HAN was born in Chengdu, China, 1993. She is currently pursuing the degree with the College of Information and Engineering, Sichuan Agricultural University, China. She has been involved in scientific research for two years. She has authored three articles in referred conferences and proceedings in the areas of natural language processing. She is a member of the China Society for Industrial and Applied Mathematics.

YAFENG REN received the Ph.D. degree with the Computer School of Wuhan University, China, 2015. He was a Post-Doctoral Research Fellow with the Singapore University of Technology and Design from 2015 to 2016. He is currently an Associate Professor with the Guangdong University of Foreign Studies. He has published over 10 papers in journals at conferences, including AAAI, EMNLP, and COLING. His research interests include natural language processing, machine learning, and data mining.



YILIN CHEN was born in Ji'an, China, 1989. He received the M.S. degree from the Department of Computer School, Wuhan University, China, in 2016, where he is currently pursuing the Ph.D. degree with the School of Computer Science. His current research interests include pattern recognition, computer vision, and computer graphics. He is a member of the China Computer Federation.