Multimedia Tools and Applications
https://doi.org/10.1007/s11042-019-7678-1

# Dialogue intent classification with character-CNN-BGRU networks

Yufan Wang [1] · Jiawei Huang [1] · Tingting He [1] · Xinhui Tu [1]

Received: 4 March 2018 / Revised: 30 March 2019 / Accepted: 24 April 2019
Published online: 11 June 2019


**Abstract**
Dialogue intent classification plays a significant role in human-computer interaction systems. In this paper, we present a hybrid convolutional neural network and bidirectional gated recurrent unit neural network (CNN-BGRU) architecture to classify the intent of a dialogue utterance. First, character embeddings are trained and used as the inputs of the proposed model. Second, a CNN is used to extract local features from each utterance, and a maximum pooling layer is applied to select the most crucial latent semantic factors. A bidirectional gated recurrent unit (BGRU) layer architecture is used to capture the contextual semantic information. Then, two feature maps, which are the outputs of the two architectures, are integrated into the final utterance representation. The proposed model can utilize local semantic and contextual information to recognize and classify the user dialogue intent in an efficient way. The proposed model is evaluated based on a social media processing (SMP) data set and a real conversational data set. The experimental results show that the proposed model outperforms the corresponding traditional methods. In addition, compared to the CNN and BGRU methods, the classification accuracy of the proposed model is 1.4% higher for the SMP data set.

**Keywords** Dialogue intent classification · CNN · BGRU · Character neural embeddings

## 1 Introduction

Intelligent dialogue systems are core technologies in artificial intelligence and human-computer interactions. Building a dialogue system usually requires great effort because the training of

✉ Tingting He
tthe@mail.ccnu.edu.cn

Yufan Wang
yufan_wang@mails.ccnu.edu.cn

Jiawei Huang
huangjava@mails.ccnu.edu.cn

[1] School of Computer Science, Central China Normal University, Wuhan 430079, China

automatic intent detection engine, spoken language understanding (SLU) and speech generation modules is essential for having effective conversations [13]. The dialogue system framework is shown in Fig. 1 [2]. Intent classification plays a significant role in the dialogue system. The objective of intent classification is to attribute one word of predefined intent to each given utterance. Therefore, this task is often considered a sequence classification problem.

Due to the characteristics of the dialogue text, many challenges must be addressed. For instance, speech variability exists among different speakers, and user dialogues are generally ultrashort text sequences with abbreviations and stronger contextual dependence than ordinary text sequences. Thus, intent classification is usually limited to certain domains. There are some traditional intent classification methods based on supervised machine learning methods, such as the naïve Bayes and support vector machine methods. The feature extraction of artificially marked text is the main goal of these methods. However, these traditional methods cannot learn deep semantic information. Therefore, the accuracy of intent classification may not improve when these methods are applied.

The rapid development of deep learning techniques has led to new methods of addressing various natural language processing (NLP) tasks. Word embeddings are distributed representations of words or characters that largely alleviate data sparsity problems [22, 24, 29, 40]. In addition, deep learning can be used to improve the performance of intent classification. With a multilayer network, the model can convert low-level text information to abstract high-level text representations [14]. A recurrent neural network (RNN) analyses dialogues word by word and saves the semantics of all the historical information in the hidden layer [36]. As a biased model, an RNN may decrease the importance of previous words because later words are more dominant. To solve this problem, Kim introduced the convolutional neural network (CNN) in the NLP field. CNNs can accurately assess discriminative phrases with a maximum pooling layer. As a model based on a sliding filter of fixed size, CNNs can also reduce the difficulty of extracting features and effectively capture semantic information [17]. In NLP fields such as sentence modelling, language modelling, and grammar analysis, neural network models have been successfully applied.

In this paper, we propose a model that can benefit from CNN and bidirectional gradient recurrent unit (BGRU) methods and pretrained character embeddings derived from the Chinese Wikipedia data set. The model is called Character-CNN-BGRU. In experiments, character-level features are used as model inputs. The results show that the character-level CNN-BGRU model of intent classification is more effective than both the widely used traditional machine learning methods and the standard feedforward neural networks. With the CNN-BGRU method, the intent classification accuracy increased by more than 1.4% for SMP data sets.

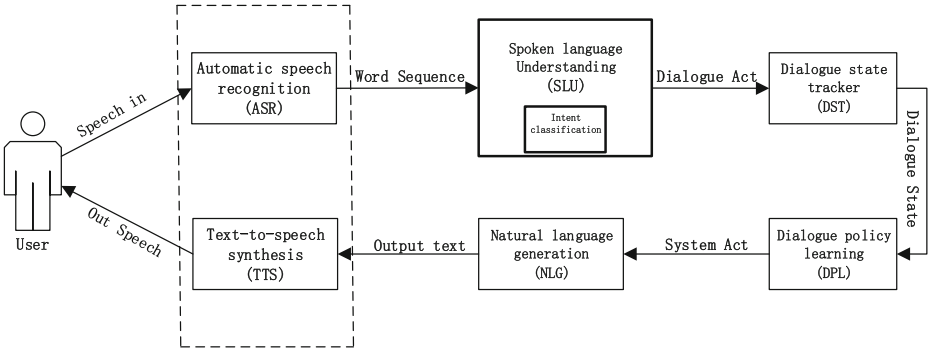


**Fig. 1** Dialogue system module framework

The remainder of this paper is organized as follows. Sect. 2 introduces the related work. The details of the model and the proposed methods are described in Sect. 3. The experimental setup is given in Sect. 4. In Sect. 5, experimental results and analysis are presented. Finally, in Sect.6, we conclude the proposed work and discus potential future work.

## 2 Related work

Previous work in dialogue intent classification has mainly focused on supervised learning methods. The popular traditional approaches include the naïve Bayes [16], hidden Markov [31], support vector machine, and decision tree [1] methods, as well as other rule-based methods [25]. Mercan Karahan et al. [15] introduced a method of combining different statistical classifiers for intent classification. Liu Ting et al. [20] proposed a method of graph ranking to study the problem of detecting consumer intent in microblogs. These methods can be applied to cases with large amounts of data or the amount of labelled data is relatively small, and all data can be involved in the learning process of the graph-ranking algorithm. Based on similarity in topic models, Asli Celikyilmazet et al. [3] exploited user intent detection in the domain of movies. However, the strong reliance on the size of the training data set makes it difficult for traditional methods to recognize different contexts for the same text dialogue.

Recently, deep neural networks have been developed in artificial intelligence research. These networks have exhibited remarkable performance in the NLP field, including in dialogue intent classification tasks. Suman Ravuri et al. [28] compared feedforward networks and RNN, Long Short-Term Memory (LSTM), and gated recurrent unit (GRU) methods for classification tasks and found that the LSTM and GRU methods performed better than the feedforward network and RNN. Ding Xiao et al. [8] proposed an unsupervised cross-domain deep learning model for consumer intention detection tasks. Liu et al. [21] explored strategies in utilizing explicit alignment information in attention-based encoder-decoder neural network models. In addition, this approach provided additional information for intent classification. Liu et al. [21] further proposed an attention-based bidirectional RNN model for intent detection. Grabes et al. [11] applied a deep bidirectional LSTM (DBLSTM) network for the intent classification problem and obtained good results. Lian Meng et al. [19] achieved excellent results in intent classification by proposing a hierarchical LSTM model that considers both word-level features and sentence-level features. Qian et al. [27] treated consumer travel intention recognition tasks as a classification problem and used a convolutional LSTM (CLSTM) neural network model to identify the consumer travel intentions.

Thus, deep learning techniques have been successively applied in intent classification [4]. Zhou et al. [41] combined the strengths of two mainstream architectures, namely, CNNs and RNNs. However, some methods only consider a single CNN or RNN. In addition, other methods that use some variants of CNNs and RNNs fail to fully utilize the advantages of each. To use as many of the sentence features obtained by CNNs and RNNs as possible, we proposed the Character-CNN-RNN model.

The model processes an input sequence of characters that can capture subword information. Xiang Zhang et al. [39] proposed a text classification model based on character-level convolutional networks, and the experimental results were excellent when the training set size was sufficiently large. Xiao et al. [34] studied consumer travel behaviour and intent recognition by establishing an LSTM network model based on convolution combining the advantages of CNNs and LSTM. Since dialogue data sets are often informal oral text sets with many out-of-vocabulary words, character-level inputs can provide more semantic information than

traditional methods and have several other benefits, such as the natural incorporation of characters and better handling of rare words [34], especially in Chinese.

## 3 Methodology

The proposed model was inspired by the works of S Lai et al. [18], Qian et al. [27] and Wang et al. [32]. Instead of using the common feedforward neural network, the more powerful CNNs and RNNs are utilized in the proposed model. In particular, we choose the BGRU network, which has been successfully used in the NLP field [19]. The architecture of the Character-CNN-BGRU model is shown in Fig. 2, which includes the main structures, such as the input layer, convolutional layer, maximum pooling layer, window feature sequence, BGRU layer and combined feature layer. Every Chinese character is converted into a character embedding. The final output of the network is the probability for each category, as calculated by the Softmax function.

### 3.1 Embedding layer

In the first step, we use semantic feature vectors to represent dialogue texts. In the typical approach, each word is converted into a word embedding vector. However, we choose character embedding in this task. The reason why we choose character-level embedding will be explicitly discussed in the experimental analysis section. By using the Python or another vector embedding tool (e.g., Word2vec [33] by Google), we can obtain the character embedding vectors that are needed. Currently, based on Wikipedia, the character embeddings are trained to obtain pretrained embedding vectors.

### 3.2 Convolutional layer

The CNN model is commonly used in classification tasks. The convolutional structure can extract important semantic information and category features from the input text.

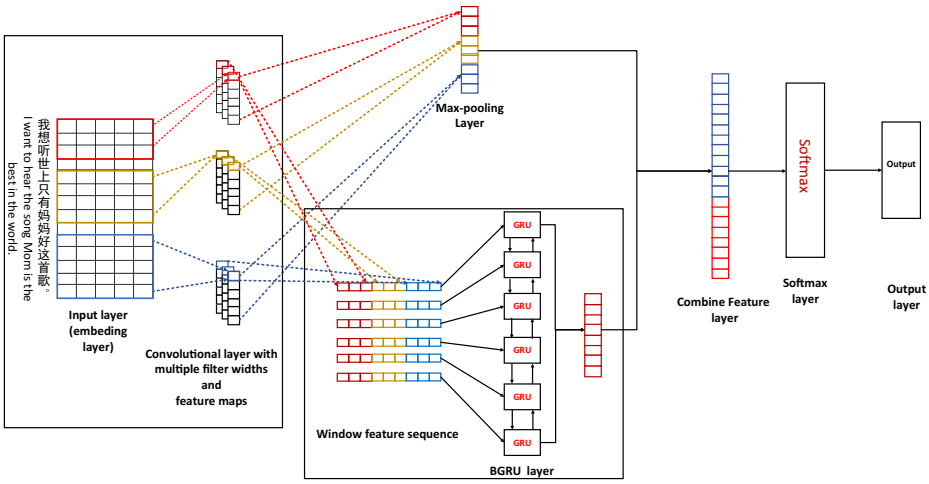


**Fig. 2** The architecture of the Character-CNN-BGRU model

In addition, because of the characteristics of weight sharing [9], there are fewer parameters required for training than in other deep learning structures. As a result, the training speed of a CNN is particularly fast. Studies in the field of computer vision have suggested that general multilayer convolutional networks are necessary to achieve better results than with single layers in deep architectures [12]. However, a single convolutional layer can achieve good classification performance in sequence classification tasks [17]. Because discrete and sparse input text may result in "overfitting" problems in intention classification, increasing the number of convolutional layers will not significantly improve the classification accuracy. Additionally, as the number of convolutional layers increases, the computational complexity and training time of the model will also increase [35]. A convolutional layer is equivalent to a sliding window that can obtain the contextual characteristics within the local window of text. Each filter slides through the sentence matrix and perform convolution operations to obtain different feature maps. The convolutional operations are formulated as follows:

$$s_i = h\left(W_s^T \cdot x_{s:s+n-1} + b_s\right) \tag{1}$$

where $W_s^T$ denotes the parameter of the convolutional filter, $x_i \in \mathbb{R}^d$, $d$ is the dimension of character embedding, $x_{i:i+n-1}$ is a convolutional filter of $n$ words from the $s$th word to the $(s+n-1)$th word and obtains local semantic information from the sentence, $b_s$ is the bias variable and $h(\cdot)$ is an activation function, which, in this case, is a leaky rectified linear unit (LeakyReLU) [23].

The model uses l LeakyReLU function as the activation function. Compared with ReLU, LeakyReLU allows a small gradient instead of a zero gradient because the ReLU function is directly set to a negative value. LeakyReLU can improve the learning efficiency of a deep learning model. Convolution operations can be classified into narrow convolution, wide convolution and equal-width convolution tasks. The lengths of the input and output are identical in an equal-width convolutional layer. In the proposed model, zero padding is required at the boundary of the equal-width convolutional layer. The structural diagram of the equal-length convolutional layer [9] is shown in Fig. 3.

Assuming that the sentence length is $n$, the output of the convolutional layer $S_i$ can be represented as follows:

$$S_i = [s_1, s_2, s_3, \ldots, s_i, s_n] \tag{2}$$

where $s_i$ denotes the local information of the sentence.

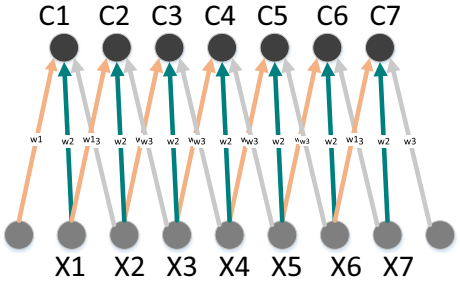


**Fig. 3** The structure diagram of the equal-length convolutional layer. The size of the filter is 3, as in the sample above, and different colours represent different weights ($w_1$, $w_2$, $w_3$), which are shared across different filters

### 3.3 The maximum pooling layer

Pooling is an important concept in CNN and generally applied after convolution operations [17]. The pooling layer which analyses the output of the convolutional layer, combines the features of various feature maps obtained with different filter sizes in the convolutional layer and reduces the dimension of the output vector. This layer can also effectively reduce the number of parameters required by the subsequent layers, mitigate the overfitting problem, and retain the important features of the text. The key feature information in a sentence can be captured by the maximum pooling layer. Additionally, the pooling layer has another function, an sentences of various length are converted to fixed-length vectors [38]. The average pooling layer sums the feature vectors and preserves the feature information to the greatest extent possible. The maximum pooling layer uses the largest feature vector to capture the most salient features of the text. In dialogue text, salient features are useful for capturing the key information in classification. Therefore, the maximum pooling layer is selected in the proposed model. The temporal complexity of the pooling layer is $O(n)$. The maximum pooling operation is formulated as follows:

$$s_{max} = max s_i, i\epsilon[1, n] \tag{3}$$

where $s_i$ is the feature map output of convolutional layer $S_i$, $s_{max}$ represents the result of the maximum pooling layer, and $n$ denotes the sentence length.

### 3.4 Window feature sequence

Another channel after convolutional operation is the BGRU. However, to retain temporal information form a sentence, a structure, namely, a window feature sequence layer, must be added before the convolutional result is input into the BGRU. In general, pooling occurs after the convolutional layer results are obtained in a deep learning model, but discontinuous feature selection sampling will destroy the temporal information associated with a sentence. To learn this temporal information, the window feature sequence layer is connected by elements corresponding to the $i$th dimension of each feature map after convolution [42]. The rearrangement operation is formulated as follows:

$$S_i = [s_1, s_2, \cdots s_n] \tag{4}$$

$$S_a = F_a^1 \oplus F_a^2 \oplus F_a^3 \oplus \cdots F_a^{n-1} \oplus F_a^n \tag{5}$$

$$v_b = F_1^b \oplus F_2^b \oplus F_3^b \oplus \cdots F_{n-1}^b \oplus F_n^b \tag{6}$$

where $S_i$ is the result of the convolutional layer, consisting of several feature maps, and $S_a (a = 1, 2, 3 \ldots n)$ represents the $a$th feature map obtained after convolution operations. Each feature map is a vector. $F_a^b$ represents the value of the element corresponding to the $b$th dimension of the $a$th feature map in the convolution result in formulas (4) and (5). $v_b$ is the vector corresponding to the rearrangement of $S_a$. All values of $v_i$ are combined to obtain the final rearrangement result $V$.

$$V = [v_1, v_2, \cdots, v_i, \cdots, v_n] \tag{7}$$

Each vector $v_i$ preserves the temporal information associated with a sentence according to the order of convolution., and $n$ is the sentence length.

### 3.5 Bidirectional gated recurrent unit layer

Pooling is performed after the convolutional layer results are obtained in a deep learning model, but discontinuous feature selection sampling will destroy the important temporal information associated with a dialogue text. Fortunately, RNNs specialize in capturing the chronological features of sentences [10]. In an RNN, each step performs the same operation, and only the input is different; therefore, in each step, each layer can share the same parameters: $u$, $v$, and $w$. Unlike in other standard neural networks, new parameters do not need to be trained in every step [5]. Thus, RNNs can greatly reduce the need for learning parameters.

Many variants of RNNs have been proposed, and we adopt the GRU in this work. An adaptive gating mechanism is introduced to help the GRU maintain the previous state and remember the extracted features from the input data. The rearranged feature maps, which were generated in the previous step, are fed into the BGRU to obtain sentence representations with temporal characteristics. A standard BGRU network [6] is shown in Fig. 4.

The basic GRU neural network was proposed by Cho et al. [6]. This paper uses a BGRU neural network, which can capture the word features of input sequences in both directions. Similar to the GRU, the BGRU contains a forward GRU and a backward GRU in the hidden layer. Compared to CNN models, which only use a fixed window (i.e., they only use some of the information from dialogue texts), bidirectional neural networks can eliminate ambiguity by using contextual information. According to the GRU correlation formula, we can deduce the formulas of the forward and backward GRUs. For each $x_t$ of the input sequence, we obtain two different representations $\overleftarrow{h_t}$, $\overrightarrow{h_t}$ for an utterance. The final characteristic output of the BGRU layer is obtained by connecting the final outputs of the forward GRU and backward GRU.

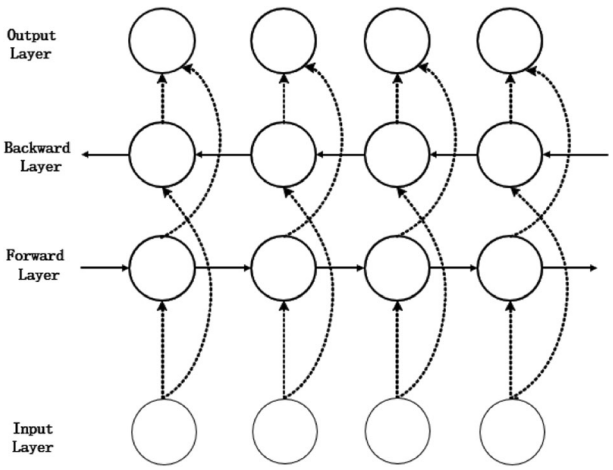


**Fig. 4** Illustration of a standard bidirectional gated recurrent unit network

### 3.6 Softmax layer

We combined the output of the maximum pooling layer and the output of the BGRU layer to obtain the final vector representation of the dialogue text. Instead of using a complex feature engineering method, the proposed model classifies the result according to the sentence feature vector representation. The Softmax classifier is widely used in multiclassification tasks. Specifically, the Softmax function maps a sentence feature vector to the corresponding category with conditional probability, as shown in the following equations:

$$y_i = g(W \cdot O_0 + b) \tag{8}$$

$$p_i = \frac{\exp(y_i)}{\sum_{j=1}^{k} \exp\left(y_j\right)} \tag{9}$$

where $O_0$ denotes the combination of the output of the BGRU layer and the output of the maximum pooling layer, $p_i$ is the category-based probability, $k$ is the class number, $y_i$ is the final vector representation of a sentence, and $g(\cdot)$ is the activation function.

## 4 Experiments

### 4.1 Data set

For the sake of verifying the effectiveness of the proposed model in intent classification tasks for a dialogue system, selecting or building an appropriate data set is crucial for obtaining reliable experimental results. In this paper, the Chinese Wikipedia data set is used for character vector training, and two Chinese data sets are used to verify the validity of the model.

Wikipedia data set. Due to the large scale, high quality, openness and easy access of the Chinese Wikipedia data set, this paper uses the Word2vec tool to conduct training based on the Chinese Wikipedia data set and generate distributed vectors as character embeddings for character representation. The character embeddings are used in the input layer in the deep learning model. The data set used in this paper is approximately 1.2 GB of Chinese text.

SMP data set. This data set is the experimental data set that was provided for "The Evaluation of Chinese Human-Computer Dialogue Technology (SMP2017-ECDT)" [30] at the Sixth National Social Media Processing Conference. The data set, which consists of 2299 dialogue texts for the training set, 770 dialogue texts for the validation set, and 677 dialogue texts for the test set, is divided into 31 categories. In addition, this experiment only considers the classification of user intent for single-round conversation. At the character level, most of the samples have less than 40 Chinese characters, and the average length of all the samples is 17.05. At the word level, each sample has less than 20 words and symbols, with an average length of 10.07. "SMP "is used to represent this data set in this paper.

Real-world conversational data set. We annotated the real Chinese conversational data provided by Noah's Ark Laboratory [7, 37]. To create objective and consistent evaluation standards among different labelling teams, we organized two teams to label the data set into four categories based on the entertainment intention: the music intention category, game intention category, film and television intention category, and sports intention category. In this paper, the Pearson correlation coefficients of the two sets of labelling results were calculated to

verify the consistency of the indicators for the two labelling teams. If the correlation coefficient is close to 1 or − 1 [26], the correlation is strong. Similarly, if the correlation coefficient is close to 0, the correlation is weak. The final Pearson correlation coefficient was 0.89, indicating that the different labelling teams had a relatively consistent understanding of the entertainment intention categories. To avoid data contingency issues, the data set was divided into five parts at random, among which four parts were used as the training set and the remainder of the data formed the test set. This paper uses "RWC" to represent this data set. The details of the RWC data set are shown in Table 1.

### 4.2 Model training

**Experimental environment** The experiment was performed with a Linux system. The specific experimental environment is shown in Table 2.

**Data processing** For Chinese dialogue text, the usual practice is to first segment the text at the character level and then train the text to obtain character vectors. The Word2vec tool by Google can be used to train the character embedding vectors and obtain segmentation results. In this study, we use pretrained word vectors with dimensions of 50, 100, 200, and 300. Typically, in neural network experiments, the character embedding vectors trained with external data set perform better than those trained only on the current data set. In this experiment, a pretrained character embedding vector provided by Wikipedia was used as the input vector in the model.

**Parameter settings** In this step, we explore the effects of different hyperparameters in the proposed method, including the character embedding size, learning rate, number of filters, batch size, filter size for convolution, and the hidden layer size. The hyperparameters of the deep learning model may have important implications for the experimental results, especially the word embedding size. We assessed the four dimensions (50, 100, 200, and 300) of the word embedding vectors, and the dimension of 200 yielded the best results. The characters not included in the set of pretrained word embeddings were initialized randomly. The input data were processed according to the feature representation method described in the previous section and used to train the Character-CNN-BGRU model. The model implemented the stochastic gradient algorithm (Adam algorithm). The other hyperparameters are shown in Table 3.

### 4.3 Comparison of methods

Standard neural network models, such as CNNs, LSTM networks, and traditional machine learning SVM models, were selected for comparison. The specific method of each model was established as follows.

**Table 1** Details of the RWC data set

| Intent | Number |
|---|---|
| Game | 5888 |
| Music | 3742 |
| Sports | 2504 |
| Video | 3516 |

**Table 2** Hyperparameters of the proposed model

| Operating system | Ubuntu 12.04 |
|---|---|
| Development language | Python 2.7 |
| Deep learning platform | TensorFlow 1.0 |
| Python platform | Anaconda |
| Memory | 8 GB |
| Hard disk | 1 TB |

- CNN + word level: We select a CNN for comparison. The filter size of the convolutional layer is set to 3, 4 and 5 for convolution. In addition, the word-level embeddings are used as text input representations.
- CNN + character level: We select a CNN for comparison. The filter size of the convolutional layer is set to 3, 4 and 5 for convolution. In addition, the character-level embeddings are used as text input representations.
- LSTM + word level: We choose a standard LSTM network. In addition, the word-level embeddings are used as text input representations.
- LSTM + character level: We choose a standard LSTM network. In addition, the character-level embeddings are used as text input representations.
- GRU + word level: We choose a standard GRU. In addition, the character-level embeddings are used as text input representations.
- GRU + character level: We choose a standard GRU. In addition, the word-level embeddings are used as text input representations.
- BLSTM/BGRU + word level: We choose a standard BLSTM/BGRU. In addition, the word embedding vectors obtained by pretraining are used in this model.
- BLSTM/BGRU + character level: We choose a standard BLSTM/BGRU. In addition, the character embedding vectors obtained by pretraining are used in this model.
- CNN-BGRU + random: We choose the proposed model. In addition, the word embeddings are randomly initialized.
- CNN-BGRU + word level: We choose the proposed model. In addition, the word embedding vectors obtained by pretraining are used in this model.
- CNN-BGRU + character level: We choose the proposed model. In addition, the character embedding vectors obtained by pretraining are used in this model.

## 5 Experimental results and analysis

Table 4 shows the model results based on the SMP and RWC data sets. The proposed model is competitive with other standard neural networks and outperforms other models according to the

**Table 3** Hyperparameters of the proposed model

| Embedding size | 200 |
|---|---|
| Hidden layer size | 40 |
| Learning rate | 0.001 |
| Loss regularization | 0.001 |
| Window size | (3, 4, 5) |
| Number of filters | 128 |

**Table 4** Results for all models

| Model | SMP data set | | | RWC data set | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 |
| SVM | 0.835 | 0.790 | 0.793 | 0.884 | 0.852 | 0.853 |
| Naïve Bayes | 0.804 | 0.763 | 0.763 | 0.852 | 0.793 | 0.812 |
| LSTM (Word) | 0.823 | 0.823 | 0.821 | 0.932 | 0.911 | 0.919 |
| LSTM (Character) | 0.847 | 0.847 | 0.846 | 0.951 | 0.949 | 0.950 |
| BLSTM (Word) | 0.862 | 0.846 | 0.846 | 0.945 | 0.920 | 0.925 |
| BLSTM (Character) | 0.886 | 0.886 | 0.885 | 0.950 | 0.954 | 0.953 |
| GRU (Word) | 0.866 | 0.866 | 0.865 | 0.933 | 0.921 | 0.927 |
| GRU (Character) | 0.871 | 0.874 | 0.872 | 0.947 | 0.946 | 0.946 |
| BGRU (Word) | 0.885 | 0.883 | 0.882 | 0.936 | 0.931 | 0.933 |
| BGRU (Character) | 0.893 | 0.893 | 0.892 | 0.947 | 0.955 | 0.954 |
| CNN (word) | 0.895 | 0.895 | 0.894 | 0.947 | 0.930 | 0.935 |
| CNN (Character) | 0.916 | 0.914 | 0.912 | 0.958 | 0.956 | 0.956 |
| CNN-BGRU (Word) | 0.871 | 0.871 | 0.864 | 0.955 | 0.951 | 0.952 |
| CNN-BGRU (Character) | 0.930 | 0.925 | 0.925 | 0.971 | 0.959 | 0.965 |

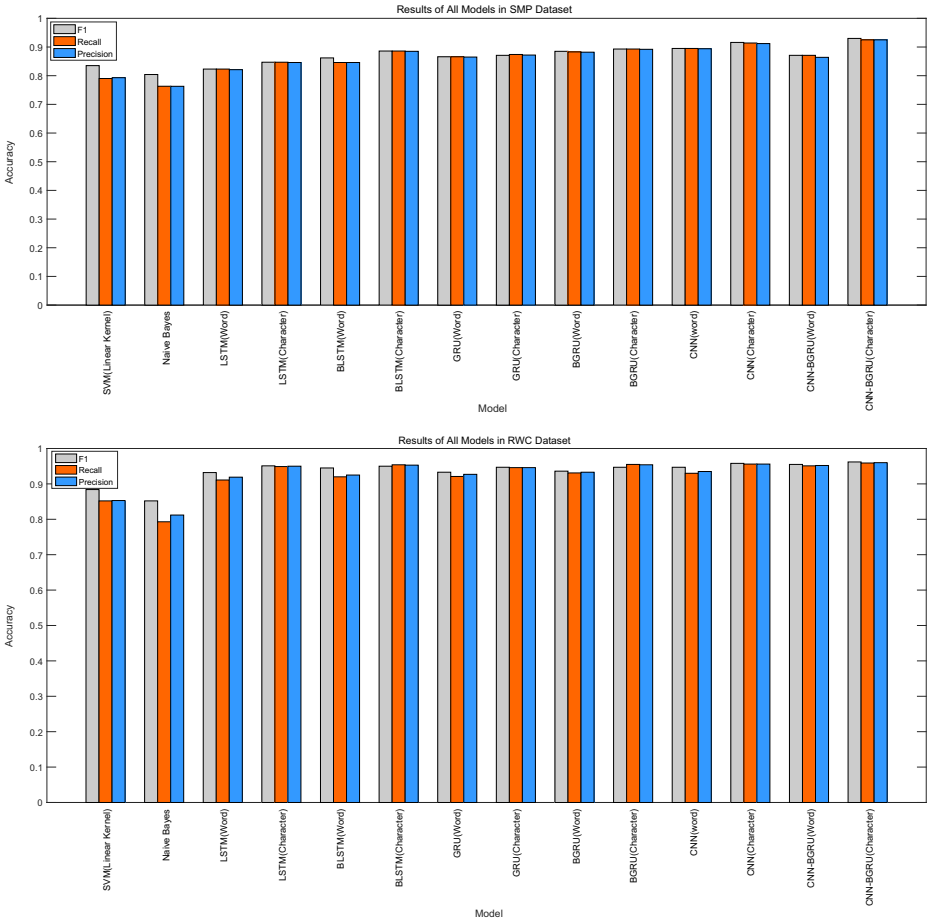


**Fig. 5** Experimental results for all models

**Table 5** Results of the comparative experiments

| Word embeddings | emb 50d | emb 100d | emb 200d | emb 300d | Random |
|---|---|---|---|---|---|
| BGRU-CNN + pre | 0.918 | 0.923 | 0.930 | 0.924 | 0.892 |

accuracy, recall and F1 score. As is shown in Fig. 5, the results of the experiments clearly indicate the superior performance of the Character-CNN-BGRU model compared to other models.

### 5.1 Randomly initialized *vs.* pretrained character embeddings

The results in Table 5 show that compared to randomly initialized character embeddings, a significant improvement is obtained using the pretrained character embeddings, which results from the fact that the character embeddings pretrained by the Word2vec tool contain contextual semantic information. The experiments show that in NLP tasks with neural network models, using pretrained embeddings in an unsupervised way is very important.

### 5.2 Character-level *vs.* word-level features

Zhang et al. constructed several large-scale data sets to show that character-level models can achieve superior or competitive results in sentence classification tasks compared to other model s [39]. Table 6 compares the model performance for different feature levels based on the SMP data set. According to the above experiments using character embeddings, we choose 200 dimensional pretrained word embedding vectors. Char represents the character-level embeddings, and Word denotes the word-level embeddings. The results indicate that for the Chinese data set, using the character-level embeddings is better than using the word-level embeddings as original features, mainly because the granularity of character-level features is smaller than that of word-level features. Additionally, the character-level features are more specific than the word-level embeddings.

Another possible reason for this result is that many similar words have one or more of the same characters in Chinese, such as “行政” and “政府”, as well as “姨妈” and “姨夫”. This phenomenon is less common in English. Notably, in Chinese, every character has independent semantics, and a word in classical Chinese (ancient Chinese) may have consisted of only one character. In addition, character-level features avoid the errors associated with the word segmentation accuracy, which are difficult to limit for Chinese text. The word-level approach fails in processing the out-of-vocabulary words that can only be mapped to an unknown word category. However, out-of-vocabulary characters are much less common. For these reasons, the character-level embedding vectors encompass more knowledge of the relevant information.

**Table 6** Different feature levels

| Feature level | Precision | Recall | F1 |
|---|---|---|---|
| CNN-BGRU + pre + Char | 0.871 | 0.871 | 0.864 |
| CNN-BGRU + pre + Word | 0.930 | 0.925 | 0.925 |

**Table 7** Different recurrent units in the proposed model

| Feature Level | Precision | Recall | F1 |
|---|---|---|---|
| CNN-BRNN | 0.918 | 0.913 | 0.913 |
| CNN-BLSTM | 0.922 | 0.918 | 0.916 |
| CNN-BGRU | 0.930 | 0.925 | 0.925 |

### 5.3 GRU *vs.* LSTM

The conclusions of a previous paper [6] indicated that both the LSTM and GRU methods perform better than the traditional RNN. However, we cannot make a concrete conclusion about which of the two gating units is better [6]. Therefore, in this paper, we empirically evaluated an RNN with three different recurrent units that are widely used: (1) a traditional tanh unit, (2) an LSTM unit and (3) a GRU unit. Our assessment focused on intent classification modelling with the SMP data set. In Table 7, we can observe that the performance of CNN-BGRU is better than that of the other models.

### 5.4 Analysis of the number of convolutional layers

Kim [17] demonstrated that the best classification results can be obtained through a single-layer convolutional network for several data sets in sentence classification tasks. To further verify the effect of the number of convolutional layers on the intent classification results, we compare convolution experiments with different number of layers.

As shown in the Table 8, single-layer convolutional network is more effective than the multilayer convolutional network. Because the size of the convolution filter is consistent with the dimension of sentence embedding, the features of an entire sentence can be obtained by the single-layer convolutional network, and the characteristics obtained by the upper layer are further compressed when convolutional operations are performed, which may lead to the loss of sentence features. In particular, if the conversation text is short, implementing convolutional operations several times may result in the severe loss of text features. In addition, multiple-layer convolutional networks will increase the time for model training.

### 5.5 Model efficiency analysis

Figure 6 shows the training results of the CNN, BGRU, and CNN-BGRU models based on the character embedding representations of the SMP data set. As shown in Fig. 6 a and b, although the training time of the proposed model is slightly longer than that for the CNN and BGRU methods per epoch, the number of training epochs required to achieve the optimal value is

**Table 8** The results of different numbers of convolutional layers

| Convolutional layer | Precision | Recall | F1 |
|---|---|---|---|
| Single-layer convolutional network (CNN) | 0.916 | 0.914 | 0.912 |
| Two-layer convolutional network (CNN) | 0.756 | 0.742 | 0.741 |
| Single-layer convolutional network (our model) | 0.930 | 0.925 | 0.925 |
| Two-layer convolutional network (our model) | 0.808 | 0.792 | 0.792 |

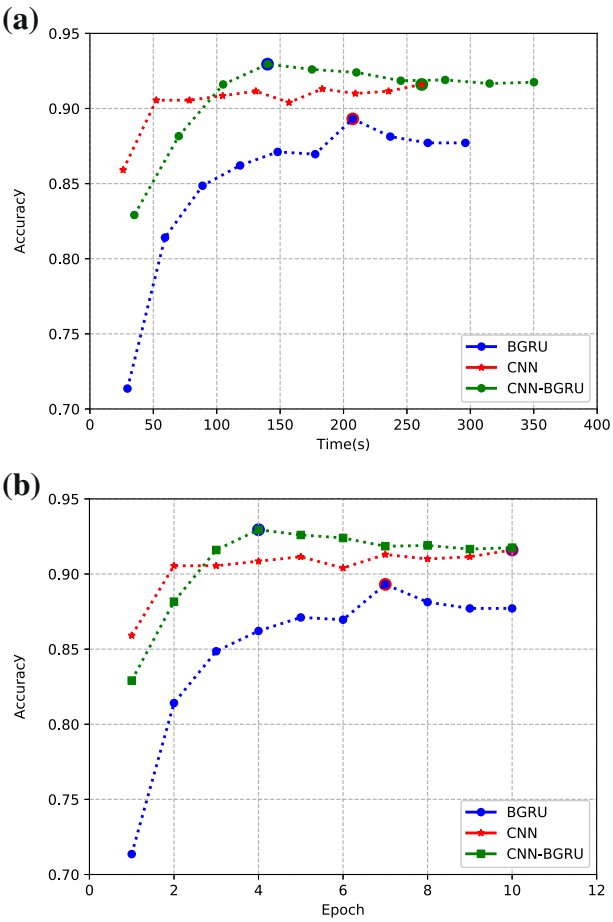


**Fig. 6** The efficiency of different models: **a** the relationship between the training time and accuracy of different models and **b** the relationship between the training epoch and accuracy of different models

lower than that for both the CNN and BGRU methods. Overall, among all the models, the model we proposed achieves the best result with a relatively short training time. Thus, this model has advantages regarding the overall training speed.

### 5.6 Sensitivity analysis

In the training of the proposed model, a number of model parameter values must be set. The number of filters and the filter window size are two key predefined parameters. During the training process, these parameters are not included in the updated hyperparameter set. Vieira et al. claimed that the number of filters and filter window size can have a significant impact on the results of the convolutional layer. Therefore, a sensitivity analysis of the key parameters was conducted in this study. These two parameters were set to 3 and 128 (filter size and the number of filters). When analysing the effect of one parameter, we fixed all other parameter values.

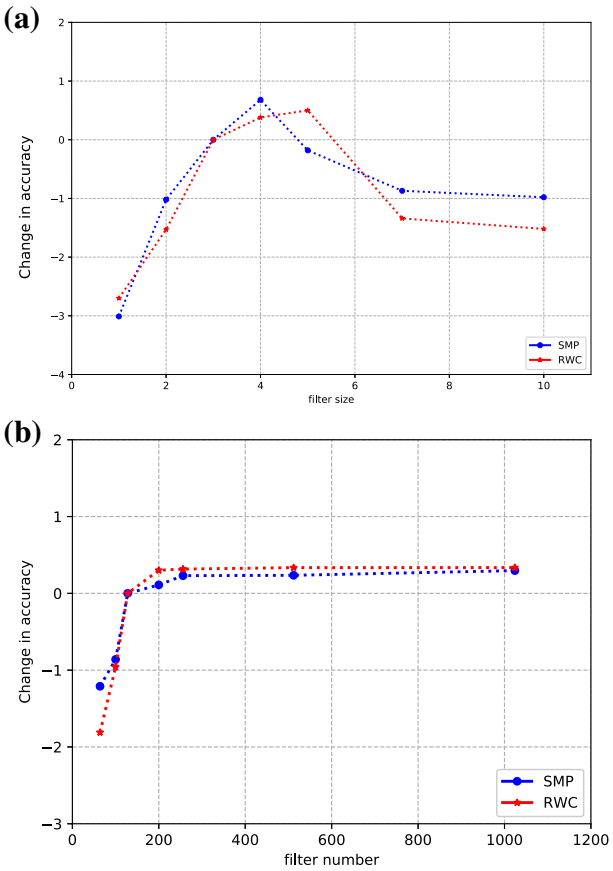


**Fig. 7** Sensitivity analysis of the filter size and number of filters

As shown in in Fig. 7(a), a filter of size 5 is the optimal value for obtaining best performance based on the RWC data set. However, a filter of size 4 is the optimal value for the SMP data set. Through experimental comparison, it can be found that as the filter size increases, the performance of the proposed model for both data sets tends to increase first and then decrease. Additionally, the model accuracy tends to stabilize when the filter size ranges between 3 and 5. To make the model more robust, we do not use a single feature window size but instead a combination of feature window sizes (3, 4, and 5). As shown in Fig. 7(b), when the number of filters is very small, the accuracy of the model will be very low. As this number increases, the performance improves and eventually stabilizes between 200 and 400. In

**Table 9** Predictions for different categories

| Categories | Accuracy |
|---|---|
| app | 0.800 |
| video | 0.793 |
| email | 0.989 |
| news | 0.950 |

addition, the computational complexity of the model increases rapidly with increasing number of feature maps, and too many feature maps will slow the training process of the model.

Therefore, we set the number of filters to 128 in the experiment based on the size of each filter. In addition, since the feature window sizes are 3, 4 and 5, the number of feature maps in the model is actually 384.

### 5.7 Analysis of performance for different categories

As shown in Table 9, the two intent categories in the SMP data set, namely, "video" and "app", are associated with relatively poor performance. In contrast, the accuracy of "email" and "news" classification is highest. We found that the results of some categories are relatively poor, potentially because the text in these categories required additional contextual information in the classification. For example, "我想看电影台北飘雪" ("I want to see the movie named Snow in Taipei") can be classified into the "video" or "cinemas" category. "打开搜狐" ("open the sohu") can be classified into the "website" or "app" category. For a single sentence, these user intent cannot be fully assessed without contextual information, even if assessed by other humans. In addition, some words cannot be well understood, possibly because the training set is not large enough, resulting in the absence of words in the training data set or in a certain domain. For instance, the model cannot determine whether "三安光电" ("three safety photoelectric") is a stock or a video. One way to alleviate this problem is to increase the number of training samples. Introducing domain-related knowledge is another good approach to solving this problem.

However, taking "email" as an example, such as "帮我发邮件给XXX" ("help me send an email to XXX"), these semantic features can be easily captured by the proposed model with remarkable performance.

### 5.8 CNN-BGRU *vs.* other models

The individual CNN and BGRU models can achieve good results. The difference between the two models is that they use different structures to capture contextual information. Fixed-window character representations are used as contextual semantic information by the CNN, and a recurrent structure is used by the BGRU to capture a wide range of contextual semantic information. The CNN performance is strongly affected by the filter size and pooling layer processing sequence. A small filter size may cause the loss of some long-distance semantic information, and a large size might result in data sparsity. The single-layer CNN ignores the historical contextual semantic information of the dialogue text. However, the historical contextual semantic information is significant in NLP tasks. The BGRU model incorporates the chronological features of sentences but has some limitations in mining the deep semantic information of dialogue text compared with CNN methods. The experimental analysis indicated that the performance of the Character-CNN-BGRU model is better than that of other models. The proposed model effectively integrates the CNN and BGRU structures to extract high-level semantic information and local information from user dialogues using the CNN. Then, the most important features of the sentences are extracted through the maximum pooling layer, and the rearranged feature maps are input into the BGRU network to obtain the corresponding contextual information. In particular, the features of a colloquial short text present difficult problems for Chinese dialogue intention classification. By considering that the characteristics of Chinese characters, we chose character-level embeddings as the network

inputs. The experimental results suggest that based on standard data, the new model is superior to other intent classification models.

## 6 Discussion

This paper studies the classification of intention in dialogue systems considering the features of the dialogue text and Chinese characters and proposes a Character-CNN-BGRU model. The model uses character vector representations. A CNN is used to extract deep text information, and a BGRU is used to obtain contextual semantic information. Therefore, comprehensive and important information contained in user discourse can be summarized by effectively integrating the CNN and BGRU structures in the model. This model provides strong sentence representations and can improve the accuracy of intent classification compared to traditional methods. Through an experimental analysis of two Chinese data sets, the effectiveness of the intent classification model was verified. The model provides a feasible solution for intent classification tasks in dialogue systems. The model can also be applied to classification tasks in other fields, such as news classification, emotion classification, mail classification, topic tracking, and so on. Therefore, the CNN-BGRU model is of certain reference significance for subsequent text classification research.

In the future, the deep learning methods will be studied from the following perspectives. 1) the selection of a domain-relevant corpus to optimize the pretrained character vectors. Although the Wikipedia data set is of high quality, it does not highlight the features of specific dialogue data sets. The training of character and word vector representations can be performed based on large-scale conversational data set. 2) The identification of some categories in intent classification requires the introduction of external knowledge. The knowledge of the entities in short dialogues can be extended to ensure that the dialogues contain sufficient information and to improve the accuracy of classification. 3) The intent classification method in this paper is based on single-turn dialogue. In the future, the research can be extended to multi-turn dialogues, and the historical context of multi-turn dialogues can be used to improve the performance of intent classification.

**Acknowledgements** This research is supported by the Fundamental Research Funds for Central Universities (CCNU18JCK05), the National Natural Science Foundation of China (61532008), the National Science Foundation of China (61572223), and the National Key Research and Development Program of China (2017YFC0909502).

## References

1. Ali SA, Sulaiman N, Mustapha A, Mustapha N (2009) Improving Accuracy of Intention-Based Response Classification using Decision Tree. Inf Technol J 8(6)
2. Becerra A, Rosa JIDL, González E (2017) Speech recognition in a dialog system: from conventional to deep processing. Multimed Tools Appl 78(2):1–37
3. Celikyilmaz A, Hakkanitur D, Tur G, Fidler A, Hillard D (2011) Exploiting Distance Based Similarity in Topic Models for User Intent Detection. IEEE Automatic Speech Recognition & Understanding Workshop: 425–430
4. Chen H, Liu X, Yin D, Tang J (2017) A Survey on Dialogue Systems: Recent Advances and New Frontiers. Acm Sigkdd Explorations Newsletter 19(2)
5. Cho K, Van Merrienboer B, Bahdanau D, Bengio Y (2014) On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. Computer Science

6. Chung, J., Gulcehre, C., Cho, K.H., Bengio, Y (2014) Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. Eprint Arxiv
7. Deep Learning for Natural Language Processing - Communication Between Neural Network Systems and Human [EB/OL]: Deep Learning for Natural Language Processing - Communication Between Neural Network Systems and Human [EB/OL]. http://www.noahlab.com.hk/topics/DeepLearning4NLP.
8. Ding, X., Liu, T., Duan, J., Nie, J.Y.: Mining User Consumption Intention from Social Media Using Domain Adaptive Convolutional Neural Network (2015)
9. Er MJ, Zhang Y, Wang N, Pratama M (2016) Attention pooling-based convolutional neural network for sentence modelling. Inf Sci 373:388–403. https://doi.org/10.1016/j.ins.2016.08.084
10. Gallicchio C (2018) Short-term Memory of Deep RNN
11. Graves A, Jaitly N, Mohamed AR (2014) Hybrid speech recognition with Deep Bidirectional LSTMAutomatic Speech Recognition and Understanding, 2014273-278
12. H., Y.L.Y.B (2015) Deep learning. Nature.
13. Hashem IAT, Yaqoob I, Anuar NB, Mokhtar S, Gani A, Ullah Khan S (2015) The rise of "big data" on cloud computing: Review and open research issues. Inf Syst 47(0):98–115. https://doi.org/10.1016/j.is.2014.07.006
14. Jingxue Liu FMYZ (2017) Character-Level neural networks for short text classification. Paper presented at the International Smart Cities Conference
15. Karahan M, Hakkani-Tur D, Riccardi G, Tur G (2003) Combining classifiers for spoken language understandingAutomatic Speech Recognition and Understanding, 2003. ASRU '03. 2003 IEEE Workshop on, 2003589-594
16. Keizer S (2001) Dialogue act modelling using Bayesian networks
17. Kim Y (2014) Convolutional Neural Networks for Sentence Classification. Eprint Arxiv
18. Lai S, Xu L, Liu K, Zhao J (2015) Recurrent Convolutional Neural Networks for Text Classification
19. Lian Meng MH (2017) Dialogue Intent Classification with Long Short-Term Memory Networks. Paper presented at the The Sixth Conference on Natural Language Processing and Chinese Computing (NLPCC), Dalian, China
20. Liu T, F.B.C.Y (2015) Detecting consumption intention based on graph ranking in social media. Sci Sin Inform
21. Liu B, Lane I (2016) Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling
22. Luong T, Socher R, Manning CD (2013) Better Word Representations with Recursive Neural Networks for Morphology Conference, 2013104-113
23. Maas AL, Hannun AY, Ng AY (2013) Rectifier nonlinearities improve neural network acoustic models
24. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed Representations of Words and Phrases and their Compositionality. Adv Neural Inf Proces Syst 26:3111–3119
25. Niimi Y, Oku T, Nishimoto T, Araki M (2001) A rule based approach to extraction of topics and dialog acts in a spoken dialog system Euro speech 2001 Scandinavia, European Conference on Speech Communication and Technology, Interspeech Event, Aalborg, 20012185-2188
26. Pearson correlation coefficient: Pearson correlation coefficient. https://en.wikipedia.org/wiki/Pearson_correlation_coefficient
27. Qian Y, X.D.T.L. (2017) Identification method of user's travel consumption intention in chatting robot. Sci Sin Inform
28. Ravuri S, Stoicke A (2016) A comparative study of neural network models for lexical intent classification Automatic Speech Recognition and Understanding, 2016368-374
29. Socher R, Bauer J, Manning CD, Ng AY (2013) Parsing with Compositional Vector GrammarsMeeting of the Association for Computational Linguistics, 2013455-465
30. Social Media Processing Homepage: Social Media Processing Homepage. http://www.cips-smp.org/smp2017/
31. Surendran D, Levow GA (2006) Dialog act tagging with support vector machines and hidden markov modelsIn Proceedings of Interspeech/ICSLP, 20061-28
32. Wang J, Wang Z, Zhang D, Yan J (2017) Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification Twenty-Sixth International Joint Conference on Artificial Intelligence, 20172915-2921
33. Word2vec Homepage (2014) Word2vec Homepage. http://code.google.com/archive/p/word2vec/
34. Xiao, Y., Cho, K (2016) Efficient Character-level Document Classification by Combining Convolution and Recurrent Layers
35. Xu X, Li W, Ran Q, Du Q, Gao L, Zhang B (2018) Multisource Remote Sensing Data Classification Based on Convolutional Neural Network. IEEE Transactions on Geoscience & Remote Sensing PP(99):1–13
36. Yao K, Peng B, Zhang Y, Yu D, Zweig G, Shi Y (2014) Spoken Language Understanding using Long Short-Term Memory Neural Networks. IEEE – Institute of Electrical & Electronics Engineers:189–194

37. Yin J, J. X. L. Z.: Neural Generative Question Answering. Paper presented at the International Joint Conference on Artificial Intelligence
38. Zhang Y, Marshall I, Wallace BC (2016) Rationale-Augmented Convolutional Neural Networks for Text Classification, 2016795
39. Zhang X, Zhao J, Lecun Y (2015) Character-level Convolutional Networks for Text Classification, 649-657
40. Zheng X, Chen H, Xu T (2013) Deep learning for Chinese word segmentation and POS taggingConference on Empirical Methods in Natural Language Processing, 2013
41. Zhou C, Sun C, Liu Z, Lau FCM (2015) A C-LSTM Neural Network for Text Classification. Comput Therm Sci 1(4):39–44
42. Zhou C, Sun C, Liu Z, Lau FCM (2015) A C-LSTM Neural Network for Text Classification



**Yufan Wang** was born in 1993. He is a graduate student at the Central China Normal University, majoring in computerscience and technology. His research interests include natural language processing and deep learning.

**Jiawei Huang** was born in 1994. He is a graduate student at the Central China Normal University, majoring in computerscience and technology. His research interests include natural language processing and deep learning.

**Tingting He** is currently the President of the school of computer science. And she is a Full Professor in School of Computer Science, Central China Normal University, China. Her areas of research interests are natural language processing£¬computational intelligence and deep learning.

**Xinhui Tu** is a Full Professor in School of Computer Science, Central China Normal University, China. His areas of research interests are natural language processing£¬computational intelligence, information retrieval and deep learning.