IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 29, 2019, accepted July 22, 2019, date of publication August 1, 2019, date of current version August 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2932619

A Hybrid Bidirectional Recurrent Convolutional Neural Network Attention-Based Model for Text Classification

JIN ZHENG AND LIMIN ZHENG

College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
Beijing Laboratory of Food Quality and Safety, China Agricultural University, Beijing 100083, China

Corresponding author: Limin Zheng (zhenglimin@cau.edu.cn)

This work was supported by the National Key Research and Development Program of China under Project 2017YFC1601800.

ABSTRACT The text classification task is an important application in natural language processing. At present, deep learning models, such as convolutional neural network and recurrent neural network, have achieved good results for this task, but the multi-class text classification and the fine-grained sentiment analysis are still challenging. In this paper, we propose a hybrid bidirectional recurrent convolutional neural network attention-based model to address this issue, which named BRCAN. The model combines the bidirectional long short-term memory and the convolutional neural network with the attention mechanism and word2vec to achieve the fine-grained text classification task. In our model, we apply word2vec to generate word vectors automatically and a bidirectional recurrent structure to capture contextual information and long-term dependence of sentences. We also employ a maximum pool layer of convolutional neural network that judges which words play an essential role in text classification, and use the attention mechanism to give them higher weights to capture the key components in texts. We conduct experiments on four datasets, including Yahoo! Answers, Sogou News of the topic classification, Yelp Reviews, and Douban Movies Top250 short reviews of the sentiment analysis. And the experimental results show that the BRCAN outperforms the state-of-the-art models.

INDEX TERMS Attention mechanism, bidirectional long short-term memory, convolutional neural network, fine-grained sentiment analysis, multi-class text classification.

I. INTRODUCTION

Text classification is an essential component in many natural language processing applications, such as topic classification [1] and sentiment analysis [2], [3]. With the rapid development of news media and social networks, a large number of news reports or user-generated texts appears on the Internet. The topic classification of news reports makes it easy to recommend relevant content according to the user's interest and improve users' reading experiences. The texts of online reviews are usually subjective and semantically oriented. Correctly distinguishing the semantics of these texts has important research value for understanding users' intentions and opinions. The goal of text classification is to assign single or multiple predefined labels to a sequence of text. Traditional approaches of text classification generally consist of a feature extraction stage and a classification stage, such as the bag-of-words (BOW), unigrams, bigrams, the Term Frequency-Inverse Document Frequency (TF-IDF), Support Vector Machine (SVM) and Naive Bayesian(NB) probability model, which focus on the design of hand-crafted features [4], [5]. Nevertheless, they often ignore the contextual information or word order in texts, and have the problem of data sparsity, which affects the classification accuracy. More recently, deep learning approaches have been shown to outperform traditional approaches, such as Convolutional Neural Network [6] (CNN) and Recurrent Neural Network (RNN) based on long short-term memory [7] (LSTM). They can effectively extract relevant features without requiring complex artificial feature engineering.

The CNN and the RNN have shown different capabilities in representing a piece of text. RNN is particularly good at modeling sequential data, and capable of building effective text representation by learning temporal features and long-term

The associate editor coordinating the review of this manuscript and approving it for publication was Ugur Guvenc.

dependencies in sentences, and successfully used in NLP tasks [8]. CNN has been proved to be able to learn local features from words or phrases [6], it uses windows to acquire the most prominent features in sentences, and attempts to extract effective text representation by identifying the most influential n-grams of different semantic aspects. It usually trains faster than RNN, but its ability to capturing features over long distances is poorer. To exploit the full advantages of CNN and RNN, some existing methods combine them together for text classification [9], [10]. However, these methods give each word the same status in the sentence, and it is difficult to distinguish the keywords which play a greater role in the classification task against the common words.

Recently, neural networks based on the attention mechanism can assign different weights to words in sentences according to their importance to the classification, which can alleviate the above problems. HAN [11], Att-BLSTM [12] and HCAN [13] can achieve state-of-the-art performance for relation classification and multiple sentences classification. However, these attention mechanisms are assigned to individual CNN or RNN, and the role of context is not prominent.

To address the limitation of the above models, we propose a Hybrid Bidirectional Recurrent Convolutional Neural Network Attention-Based Model (BRCAN) for text classification. Firstly, word2vec is used to segment words and generate word vectors. A bidirectional long short-term memory (Bi-LSTM) captures the contextual information to the greatest extent possible when learning word representations, and reserves a larger range of the word order when learning representations of texts. Second, a maximum pooling layer of CNN is employed to judge which word play the key role in text classification through contextual information. Using the attention mechanism to give words higher weights to capture the key components in texts. Finally, the logical regression classification layer realizes the fine-grained text classification.

The contributions of this paper can be summarized as follows:

1) This paper proposes a hybrid framework, which utilizes Bi-LSTM to capture the contextual information and long-term dependence of sentences, picks the useful local features from the sequences generated by the Bi-LSTM according to the convolution and maximum pooling operations, and assigns different weights according to its importance by the attention mechanism to realize text classification effectively.
2) We validate BRCAN on four text classification tasks, including topic classification and sentiment analysis. Compared with the state-of-the-art models, BRCAN achieves excellent performance on these tasks. Specifically, it achieves the highest accuracy on fine-grained sentiment analysis datasets.
3) In order to achieve the best classification performance of BRCAN, we propose to use the bilinear attention function that realizes the interaction between vectors in the attention mechanism, and capture the local semantic features of n-grams with different granularity by using multiple convolution filters. A large number of experiments have been conducted to verify the effect of BRCAN on text classification, analyze the performance of BRCAN on different datasets, the influence of attention mechanism on the model, and conduct a sensitivity analysis of convolutional layer, filter, and maximum pooling size.

II. RELATED RESEARCH

In recent years, neural network models based on deep learning have achieved great improvement on topic classification and sentiment analysis. We group them into the following categories for a brief review.

A. CONVOLUTION NEURAL NETWORKS

CNN is able to extract local and deep features from natural language and has achieved good results in sentence classification. In 2014, Kim proposed the first convolutional neural network for text classification which has a simple and powerful architecture [6]. The Sogou News was proposed by Wang *et al*. in 2008 [14]. In 2015, Zhang proposed a character-level convolution neural network, and achieved 95.61% and 42.13% accuracy on Sogou News and Yelp Reviews [15]. Meanwhile, Zhang first proposed Yahoo! Answers and achieved an accuracy of 68.03% on this dataset. In 2016, Conneau A *et al*. only used very small convolution and pooling operations in character-level CNN, and achieved 96.39%, 72.83% and 35.74% accuracy on Sogou News, Yahoo! Answers and Yelp Reviews respectively [16]. In 2017, Quispe O *et al*. proposed a method that used the CNN with the latent semantic index for feature extraction, which achieved high accuracy on Yahoo! Answers [17]. Johnson R *et al*. proposed a CNN model using shallow word-level sequences instead of deep character-level sequences as input can improve the performance on Yahoo! Answers, Sogou News and Yelp Reviews [18]. In 2017, Johnson R proposed to use the deepening of word-level CNN to improve classification performance [38]. After that, in 2018, Le *et al*. found that the long-term correlation in sentences can be captured and the performance of character-level CNN can be improved by increasing the depth of CNN [19], [9].

B. RECURRENT NEURAL NETWORKS

RNN is able to learn long-term dependencies in sequential data and is successfully applied to speech recognition and machine translation. Recent research has found that it is also applied to the text classification tasks [20], [7]. Yogatama D *et al*. proposed an LSTM-based generative classification model on Sogou News, Yahoo! Answers and Yelp Reviews. The experimental results showed that the model has strong robustness [21]. In 2018, Wang B *et al*. proposed to use the disconnected recurrent neural networks for Yahoo! Answers and Yelp Reviews, which captured key phrases and long-term dependencies [22]. RNN may explode or disappear in gradient. Hochreiter S *et al*. first proposed the

LSTM model to solve this problem by learning long-term dependencies [7].

In 2015, Huang Z et al. proposed various sequential marker network models based on LSTM, which were less dependent and robust to word embedding [23].

C. HYBRID CNN-RNN MODELS

CNN and LSTM are the most advanced semantic synthesis models for text classification. Xiao Y *et al*. proposed a combination model of CNN and RNN on Sogou News, Yahoo! Answers and Yelp Reviews which achieved the accuracy of 95.17%, 71.38% and 61.82% [9]. In 2017, Hassan A *et al*. proposed a model that relied on CNN and bidirectional RNN, replacing the pooling layer in CNN with the bidirectional layer, which had high performance [10]. In 2018, Hua Q *et al*. proposed a hybrid CNN and bidirectional RNN model to classify Sogou News and Yelp Reviews at the character level, which was more accurate than using a single CNN or RNN [24]. Meanwhile, Marinho W *et al*. proposed to use tensors to represent text and assign shorter codes to the most commonly used characters. Two variants CNN and LSTM models were used to train on Sogou News, Yahoo! Answers and Yelp Reviews, with high accuracy of 95.16%, 93.96%, 94.52%, 68.10%, 70.24%, 70.27% and 58.0%, 57.03%, 59.71% [25]. Tang D *et al*. proposed a network model which used CNN or LSTM to generate RNN to adaptively encode sentence semantics and their internal relations. The results showed that using the gated recurrent neural network in this model was significantly better than the standard CNN [26].

D. ATTENTION-BASED MODELS

The attention mechanism has become an effective strategy for dynamic learning the contribution of different features to specific tasks, which improves the performance of text classification [23]. Wang Y *et al*. designed an attention mechanism that captured key parts of sentences to respond to text classification [27]. Yang Z applied two levels of attention mechanism at the word and sentence, and the accuracy of Yahoo! Answers and Yelp Reviews reached 75.8% and 71.0%, respectively [11]. In 2018, Gao S *et al*. proposed a hierarchical convolutional attention neural network for text classification, which was more accurate than the most advanced classification models, and the training speed was twice as fast as the current [13].

Although these deep learning methods are effective on Sogou News, Yahoo! Answers, and Yelp Reviews, fine-grained text classification is still very challenging. Therefore, the text has designed a stronger neural network model. One basic motivation for using the recurrent layer is that it can effectively capture long-term dependencies between sentences in a text even if there is only a single layer. Further use of the bidirectional recurrent layer can alleviate the imbalance of information when capturing longer sentences. The recurrent layer has the advantage of better capturing context information and is conducive to capturing the semantics of long text. However, the recurrent layer is computed based on the whole input sequence. With the linear growth of the length of the input sequence, the time complexity is higher. This is in contrast to the convolutional layer for which computations can be efficiently done in parallel. The convolutional layer can learn to extract local features in sentences. By stacking multiple convolutional layers and using filters of different sizes, it can effectively higher-level local features from the input sequences. The maximum pooling layer can fairly determine discriminative phrases in texts. However, neither the recurrent layer nor the convolutional layer can give higher weight to those words that play a decisive role in text classification. Each word has different importance for text classification, especially for fine-grained text classification, individual keywords directly determine the classification results. Based on this motivation, we consider using the attention mechanism network that captures more valuable information in the text for classification. Considering the above motivation, we propose a hybrid bidirectional recurrent convolutional neural network attention-based model. In order to evaluate the performance of the model, we verify it on the four datasets. The results show that the proposed model achieves the optimal classification effect with smaller size and fewer parameters.

III. PROPOSED METHOD

This section describes the network architecture in our model, including word embedding and hybrid neural networks. The classified document consists of a list of sentences, and each sentence consists of a list of words. First, the words in sentences are entered into the input layer. In the embedding layer, word2vec model is used to learn the representations of word vectors. Second, input the represented vector into the Bi-LSTM layer to learn the long-term dependence between the sentences in the text. Third, the intermediate sentence feature representations generated by Bi-LSTM are input into CNN layer to capture the local features of sentences, a maximum pooling layer of CNN is employed to judge which words play the key role in text classification through contextual information. Forth, use the attention layer to give them higher weights to capture the key components in texts. Finally, the logical regression classifier of the output layer realizes the fine-grained text classification. The overall structure of the model is shown in Fig. 1 and the flowchart of the whole algorithm is shown in Fig.2. The components and structure of the model are described in detail in the following sections.

A. EMBEDDING LAYER

Word embedding usually needs to transform words into vectors with the low-dimensional distribution. In fact, it maps words from vocabulary to a corresponding vector of real values to capture the morphological, syntactic and semantic information of words. The bag of words model is also low-dimensional, but there is a lack of context between words. To better represent the text content, we use the skip-gram model [28] in word2vec [29] to train data and learn the

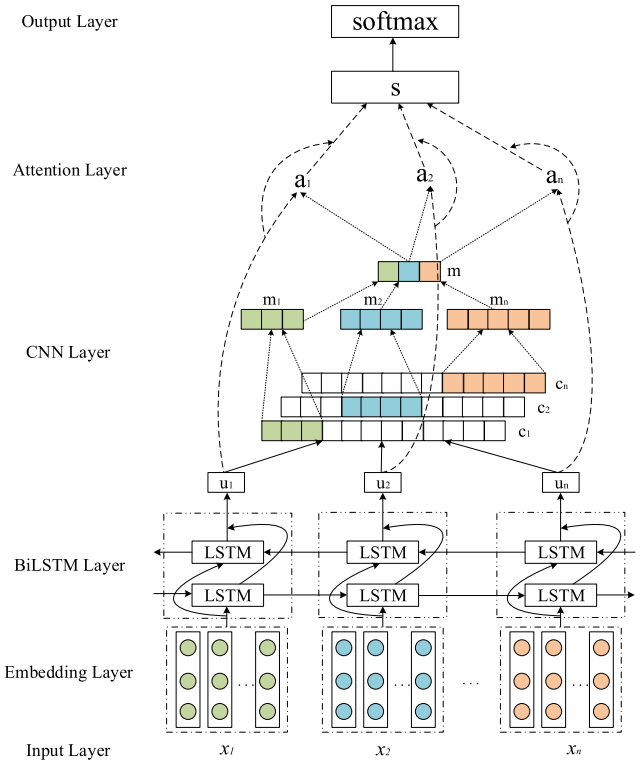


FIGURE 1. The architecture of a bidirectional recurrent convolutional neural network attention-based model.

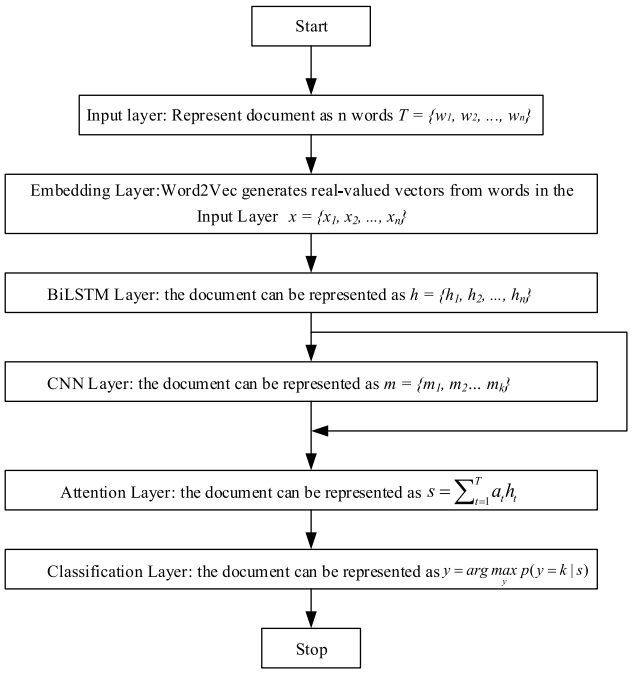


FIGURE 2. The flowchart of the whole algorithm.

context between words. In the model, T = $\{w_1, w_2, \ldots, w_n\}$ is used to represent the given document and n words, each word w_i is converted into a real-valued vector x_i. We first transform a word w_i into its one-hot encoding vector v_i, and then the embedding matrix W is used to transform v_i into its

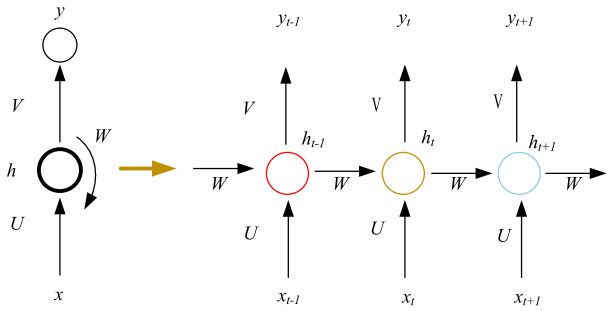


FIGURE 3. RNN expands into a complete network structure.

word embedding x_i. The calculation formula is as follows:

$$x_t = Wv_i \tag{1}$$

$W \in R^{d \times |V|}$, where $|V|$ is a fixed-sized vocabulary and d is the dimension of the word embedding vector. In this way, the document can be represented as real-valued vectors $x = \{x_1, x_2, \ldots, x_n\}$ and fed into the bidirectional recurrent neural network layer.

B. BIDIRECTIONAL RECURRENT LAYER

RNN is a kind of neural network that uses sequence information and maintains its characteristics through the middle layer. It can process any length of the sequence by using the mechanism of back propagation and memory. The variable length sentence vectors are mapped to fixed length sentence vectors, by truncating or filling the sequence. RNN introduces a time(state)-based convolution mechanism, which allows RNN to be regarded as multiple convolutions of the same network at different time steps. Each neuron transmits the currently updated results to the neurons at the next time step. So, the RNN layer is used to extract the temporal features and long-term dependencies from the text sequences. As shown in Fig. 3, the RNN is expanded into a complete network.

In Fig. 3, the word embedding vectors $\{x_1, x_2, \ldots, x_n\}$ are put into the recurrent layer step by step. The word vector x_t and h_{t-1} which presents the hidden state of the previous step are the input sequence of time t. The hidden state of time t, h_t is the output. U, W, and V are the weight matrices. The RNN is developed according to the input. The longer the input is, the deeper the expansion is. Deep network training often suffers from the problem of gradient disappearing or explosion. So we use LSTM to prevent the gradual disappearing gradient by controlling the information flow. The long-term dependence can also be more easily captured [30], [31]. LSTM has a complex structure in the recurrent layer, which uses four different layers to control information interaction. LSTM designs a 'gate' storage unit structure to remove or increase information.

First, the 'forget gate' determines which information should be discarded from the cell.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{2}$$

Then enter the 'input gate' to determine which information to be updated, and create a new candidate value vector G_t

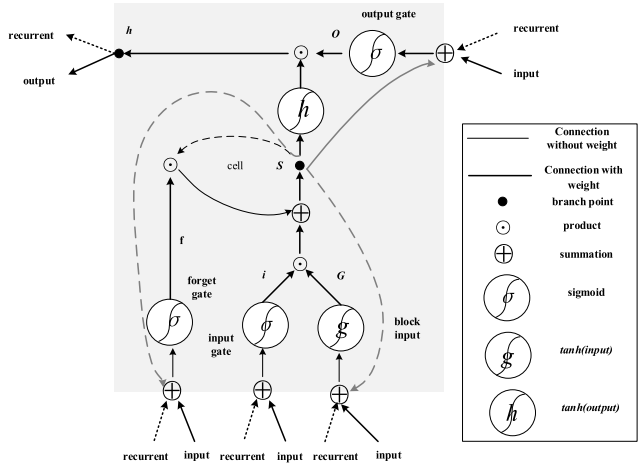


FIGURE 4. LSTM internal structure.

through the *tanh* layer.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i \tag{3}$$

$$G_t = tanh\,(W_G \cdot [h_{t-1}, x_t] + b_G) \tag{4}$$

While, the old cell state S_{t-1} is multiplied by f_t, useless information is discarded, and the product of i_t and G_t is added. The new candidate value is calculated to update the old cell state.

$$S_t = f_t \cdot S_{t-1} + i_t \cdot G_t \tag{5}$$

Finally, the final output value is determined by the cell state S_t. First, the sigmoid gate is used to determine which part of the cell state to be output, and then the cell state is processed through the *tanh* gate and multiplied by the output of the sigmoid gate. Finally, only the part to be output is determined.

$$O_t = \sigma\,(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = O_t \cdot tanh\,(S_t) \tag{7}$$

W represents the weight matrix, b represents the bias, f_t, i_t, O_t represent the weight values of the forget, input, and output of the LSTM, σ and *tanh* represent the sigmoid function and the hyperbolic tangent function, G_t and h_t represent the memory representation and the hidden layer state representation of LSTM at time t. The network structure of LSTM is shown in Fig. 4.

In different time steps of the recurrent layer, the amount of information obtained by the hidden state is unbalanced. The earlier hidden state obtains the less vector calculation, whereas the later hidden state obtains more vector calculation. The proposed model in this paper can be further extended to alleviate the problem of information imbalance by using the bidirectional recurrent layer. The bidirectional recurrent layer consists of two opposite recurrent layers which return two hidden state sequences from the forward and the backward directions:

$$h_{forward} = \left(\overrightarrow{h}_1, \overrightarrow{h_2}, \ldots \overrightarrow{h_n}\right) \tag{8}$$

$$h_{reverse} = \left(\overleftarrow{h_1}, \overleftarrow{h_2}, \ldots \overleftarrow{h_n}\right) \tag{9}$$

$$h_t = \left(\overrightarrow{h}_t, \overleftarrow{h}_t\right) \tag{10}$$

In this way, the document can be represented as $h = \{h_1, h_2, \ldots, h_n\}$.

C. CONVOLUTIONAL LAYER

Since Bi-LSTM has access to the future context as well as the past context, h is related to all the other words in the text. The CNN layer extracts the most influential n-grams of different semantic aspects from the text. In the next step, the matrix composed of feature vectors will be processed effectively. The convolution and the max pooling operation in the convolutional layer will be utilized to capture more meaningful information.

A matrix $h = \{h_1, h_2, \ldots, h_n\}$, $h \in \mathrm{R}^{n*d}$, is obtained from Bi-LSTM Layer, where d is the size of word embeddings. Then the convolutional layer extracts local features over h, and it consists of two stages. A convolution operation involves a set of k filters at the first stage. Each filter $F \in \mathrm{R}^{l \times d}$, which is applied to a window of l words to produce a new feature c_i from a window of vectors $h_{i:i+l-1}$ as follows:

$$c_i = f\,(F \cdot h_{i:i+l-1} + b) \tag{11}$$

where b is the bias, and f is the nonlinear activation function *tanh*. The filter is applied to each possible window of the matrix h to produce a feature map $c = [c_1, c_2, \ldots, c_{n-l+1}]$. With k filters, the first stage produces k feature maps.

The pooling layer converts the text with various lengths into a fixed-length vector. With the pooling layer, we can capture the information throughout the entire text. For text classification tasks, only a few words and their combinations are useful for capturing the meaning of the document, and the maximum pooling layer enables the most uses latent semantic factors to be found in the document. So in the second stage, max pooling operation is applied to each feature map to extract the maximum value $m = max\{c\}$. After extracting k features from feature map, the pooling results are combined $m = \{m_1, m_2 \ldots m_k\}$ as the output of the CNN layer.

D. ATTENTION LAYER

Considering that not all words have the same effect on the representation of sentence meaning, and the contribution of each word to the text classification is different. The optimal feature dimension is obtained by adding attention mechanism and using weighted linear combination of vectors. The attention mechanism is used to train these weights so that more important features can get higher weights. Based on the attention mechanism, we extend our model with the attention layer. After the maximum pooling layer, we use the attention layer to combine the local feature representations generated by the convolution layer with the intermediate sentence feature representation generated by the recurrent layer to calculate the attention weights [32]. BRCAN first takes the last hidden layer state in the recurrent layer h_s as a discriminant reference and combines the current hidden layer state h_t to get u_t as the

hidden representation of h_t.

$$u_t = h_t^T w_a h_s \tag{12}$$

Then calculate the attention weight distribution u_t, use the output of the CNN layer m as the context vector of the attention model, locate the informative word from u_t, and use a softmax function to get a normalized importance weight a_t.

$$a_t = \frac{exp\left(u_t^T m\right)}{\sum_{t=1}^{T} exp\left(u_t^T m\right)} \tag{13}$$

After the attention weights are obtained, we compute the text vector s as weighted arithmetic mean based on the weights $a = \{a_1, a_2, \ldots, a_n\}$, the final sentence representation is given by:

$$s = \sum_{t=1}^{T} a_t h_t \tag{14}$$

Hidden representation function used in this paper is a bilinear attention function, which is different from the function used in the previous studies on attention mechanism. This function uses fewer parameters to obtain the interaction between hidden layer states. Hidden representation function studies previously to divide the hidden layer state vectors. The function is multiplied by different regions of the weight matrix. The activation function performs nonlinear transformation according to the elements, and the last point multiplication operation is scaled by the elements. There is no interaction between the two-state vectors.

$$u_t = v_a^T tanh\left(w_a \left[h_t : h_s\right]\right) \tag{15}$$

The context vectors in traditional attention mechanisms are the same to all the samples, each sample has a unique context vector in BRCAN, which provides more flexibility and potential to achieve better performance. The context vector contains useful information which can guide the attention model to locate informative words from the input sequences, and thus plays an important role in the attention mechanism. However, previous works either ignore the context vector or initialize randomly, which weakens the role of the context significantly [33]. According to the context vector generated by the final convolution layer, BRCAN picks useful local features from the intermediate sentence representation generated by the recurrent layer, and uses the attention layer to assign different weights to these local features, and thus reserves the merits of three models. The output of the attention layer is taken as the input of the output layer.

E. CLASSIFICATION LAYER

The classification layer is a logistic regression classifier. Given the fixed dimension input from the lower layer, the classification layer calculates the prediction probability of all categories by the *softmax* function [34].

$$p\left(y = k|s\right) = \frac{exp\left(w_k^T s + b_k\right)}{\sum_{k'=1}^{k} exp\left(w_{k'}^T s + b_{k'}\right)} \tag{16}$$

$$y = \underset{y}{argmax}\, p(y = k|s) \tag{17}$$

where w is the weight, b is the bias, and k is the number of target classes.

A reasonable training objective to be minimized is the categorical cross-entropy loss. The loss is calculated as a regularized sum:

$$J(\theta) = -\frac{1}{k} \sum_{i=1}^{k} \overline{y_i} \log(y_i) + \lambda ||\theta||_F^2 \tag{18}$$

where $\overline{y_i}$ is the ground truth label, y_i is the estimated probability for each class by *softmax*, k is the number of target classes, and λ is an L2 regularization hyper-parameter.

IV. EXPERIMENTS

A. DATASETS

The text uses four large-scale text classification datasets to evaluate the performance of the model. These datasets are divided into two types of text classification tasks, sentiment analysis and topic classification. The sentiment classification dataset includes Yelp Reviews full and Yelp Reviews polarity, Douban movie top250 short reviews full and Douban movie top250 short reviews polarity, and the topic classification dataset includes Yahoo! Answers and Sogou News.

1) Yelp REVIEWS FULL

The Yelp reviews dataset is obtained from the Yelp Dataset Challenge in 2015. Five classes represent the number of starts that users have given. The full dataset has 130000 training samples and 10000 testing samples in each star. Since the original dataset does not provide verification samples, we randomly select 10% of the training samples for verification.

2) Yelp REVIEWS POLARITY

The Yelp reviews dataset is obtained from the Yelp Dataset Challenge in 2015. The original data is transformed into a polarity problem. Rating of 1 and 2 stars are represented as Bad, 4 and 5 as Good. The polarity dataset has 280,000 training samples and 19,000 test samples in each polarity. Since the original dataset does not provide verification samples, we randomly select 10% of the training samples for verification.

3) Douban MOVIE TOP250 SHORT REVIEWS FULL

It is a self-defined dataset which is collected, cleaned and labeled according to its URL. It is divided into five grades: highly recommended, recommended, okay, poor and very poor. It manually sets the highly recommended as rating 5, recommended as rating 4, okay as rating 3, poor as rating 2, very poor as rating 1. Each rating obtains 100 pieces of data. There are total 125,000 pieces of data, of which 100,000 pieces of data are used as training set labeling instances. 12,500 pieces of data are used as test set labeling instances, and 10,000 pieces of data are used as a verification set labeling instances.

TABLE 1. Dataset partition.

Classification	Dataset	Number of categories	Training set	Verification set	Test set
Sentiment classification	Yelp.F	5	650000	65000	50000
	Yelp.P	2	560000	56000	38000
	DB.F	5	100000	10000	12500
	DB.P	2	80,000	8,000	10,000
topic classification	YA	10	140,000	140,00	5000
	SN	10	50000	5000	10000

4) Douban MOVIE TOP250 SHORT REVIEWS POLARITY

Douban movie top250 short reviews are transformed into a polarity problem. Rating of 1 and 2 stars are represented as poor, 4 and 5 as recommended. Each rating obtains 100 pieces of data. There are total 100,000 pieces of data, of which 80,000 pieces of data are used as training set labeling instances, 10,000 pieces of data are used as test set labeling instances, and 8,000 pieces of data are used as a verification set labeling instances.

5) Yahoo! ANSWERS

The dataset consists of 10 categories including society and culture, science and mathematics, health, education and reference, computer and internet, sports, business and finance, entertainment and music, family and relationships, and politics and government. Documents used in this article include the title of the question, the background of the question and the best answer. There are 140,000 training samples and 5,000 test samples. Since the original dataset does not provide verification samples, we randomly select 10% of the training samples for verification.

6) Sogou NEWS

Sogou News, which uses the Sogou CA and Sogou CS joint news datasets. It contains news articles from various thematic channels, totaling 2,905,551. We label each news by using its URL, manually classifying their domain names, and labeling their categories. The subset selected in this paper contains 10 categories: IT, Finance, Health, Education, Military, Tourism, Automotive, Sports, Culture, and Recruitment. Each category contains 6,500 pieces of data, each category contains 5000 pieces of data in the training set, 500 pieces of data in the verification set, and 1000 pieces of data the test set.

Although the Douban movie top250 short reviews and Sogou News are datasets in Chinese, we use pypinyin package combined with jieba Chinese segmentation system to produce Pinyin-a phonetic Romanization of Chinese, so that the proposed network model can train the two datasets [15].

The classification of the dataset used in this paper, the number of categories, the division of the training set, test set, and verification set are shown in Table 1.

B. EXPERIMENTAL SETUP

We use two different types of datasets, the topic classification dataset and the sentiment analysis dataset, to illustrate the applicability of our proposed model. Each sentiment analysis dataset includes full and polarity. In order to highlight the model we proposed is more effective for fine-grained sentiment analysis, and then we introduce polarity for comparison. We standardize each dataset. First, we convert the Chinese dataset to Pinyin, lowercase all characters in the text and delete symbols other than commas, periods, exclamation marks, and question marks (especially web emoticons). Since there are no official validation sets in these datasets, we randomly select 10% of the training samples as validation sets for each dataset. The evaluation metric of these datasets is accuracy, which is compared with the state-of-the-art work.

The embedding layer realizes the distributed representation of words, and each word is represented as a low-dimensional, continuous real-value vector. In order to highlight the advantages of the network structure we designed, we controlled the word embedding factor when compared with other network models. All the deep learning models used in the experiment used word2vec to train the word vector. In our experiment, we obtain the skip-gram algorithm in word2vec network model to get the pre-trained word vector, and update it during the training process. We set the word embedding dimension to 300.

When constructing the neural network for feature extraction, we invest extra time to debug the parameters considering that different combinations of parameters may achieve different results. The hyper parameters of the model are tuned when training the validation sets. We save the model parameters with the highest validation accuracy and use those parameters to evaluate on the test set.

In terms of the RNN layer, we use Bi-LSTM to effectively alleviate the problem of information imbalance and set the dimension of LSMT to 100 and thus the Bi-LSTM layer provides the intermediate sentence representation vector h_t with 200 dimensions. In terms of the CNN layer, by using stacked 3 convolutional layers and 3, 4, 5 filters with 256 feature maps each, higher-level local features can be effectively extracted from the sequence. The details are further introduced in Section V.

Dropout effectively regulates the deep neural networks [28], because it takes less time to prevent overfitting. Dropout weakens the joint adaptability of neuron nodes and enhances the generalization ability. We set the dropout as 0.3 and apply it after the CNN layer as well as after the Bi-LSTM layers. We set the mini-batch size as 256 and the learning rate of Adam [35] as 0.001. Other parameters in our model are initialized randomly.

C. BASELINE METHODS

We compare our method BRCAN with several baseline methods which are widely used text classification tasks.

1) FEATURE-BASED METHODS

We compare BRCAN with several powerful baseline models for text classification. These baselines mainly use machine learning methods with unigram and bigrams as features and SVM as classifiers [36].

This paper chooses the linear method of traditional methods, uses statistical data as features and uses the linear classifier of logistic regression to realize text classification, including Bag-of-words, Bag-of-words and its TFIDF(term-frequency inverse-document-frequency) [15] and Naive Bayes methods [21]. We compare whether the proposed model is better than the classifiers of strict feature engineering.

2) SINGLE-LAYER NEURAL NETWORKS

We choose convolutional neural networks for comparison, including word-based CNN [6] model and character-based CNN model. Word-based CNN model uses one convolutional layer for the word representation of the documents, while character-based CNN model uses six convolutional layers [15] and 29 convolutional layers [16] for the character representation of the documents.

We choose recurrent neural networks for comparison, including RNN, LSTM [15] and disc-LSTM [21]. Each hidden state of the recurrent neural network is calculated based on the whole input sequence. It treats the whole document as a single sequence and the average of the hidden states of all words are used as features for classification.

3) HYBIRD NEURAL NETWORKS

We choose some hybrid network models based on CNN and RNN, extract the local features of sentences through CNN, and capture the long-term dependence between words in sentences through RNN. These hybrid network models including CNN-RNN [9], CNN-LSTM [10] and RNN-CNN [37].

4) NEURAL NETWORKS BASED ON ATTENTION MECHANISM

We choose some network models based on attention mechanism, including Att-CNN [13], Att-RNN [11], Att-BLSTM [12] and Att-CRAN [33]. Att-BLSTM is similar to our BRCAN model without the CNN layer. Att-CRAN is similar to our BRCAN model, but the overall network structure is different.

V. RESULT AND DISCUSSION

A. OVERALL PERFORMANCE

Table 2 and Table 3 present the comparison of the accuracies of BRCAN model and other state-of-the-art models on four classification tasks. The BRCAN model achieves excellent performance on all tasks. The accuracies in Yahoo! Answers, Sogou News, Yelp Reviews full, Yelp Reviews polarity, Douban movie top250 short reviews full and Douban movie top250 short reviews polarity achieve 77.75%, 97.86%, 73.46%, 96.81%,75.05%, and 96.32%. Especially, the accuracies of Yelp reviews full and Douban movie top250 short reviews full for fine-grained sentiment analysis improved 2.46% and 5.22% compared with the best model.

TABLE 2. The accuracy(%) of topic classification models.

Model	Yahoo! Answers	Sogou News
SVM+Bigrams	71.57	93.05
SVM+Unigrams	70.36	92.91
BOW	68.90	92.85
BOW+TFIDF	70.14	93.45
Naive Bayes	68.70	86.30
CNN-word	68.03	95.61
CNN-char	70.45	95.12
VD-CNN	73.43	96.82
RNN	76.26	94.90
LSTM	70.84	95.18
Disc-LSTM	73.70	94.90
CNN-RNN	71.74	95.17
CNN-LSTM	71.38	95.18
RNN-CNN	72.80	95.73
Att-CNN	73.20	95.32
Att-RNN	75.80	96.10
Att-BLSTM	72.75	95.90
Att-CRAN	73.72	96.85
BRCAN	77.75	97.86

TABLE 3. The accuracy(%) of sentiment analysis models.

Model	Yelp.F	Yelp.P	Douban.F	Douban.P
SVM+Bigrams	62.40	90.63	65.30	87.67
SVM+Unigrams	61.10	89.71	64.20	87.31
BOW	57.99	92.24	62.73	86.92
BOW+TFIDF	59.86	93.66	63.65	87.09
Naive Bayes	51.40	86.00	61.71	86.37
CNN-word	59.84	95.40	64.92	90.20
CNN-char	61.60	94.75	65.35	90.71
VD-CNN	64.72	95.72	66.07	91.38
RNN	57.34	93.16	64.82	91.43
LSTM	58.17	94.74	65.00	92.10
Disc-LSTM	59.60	92.60	65.13	92.03
CNN-RNN	61.82	94.49	66.54	93.40
CNN-LSTM	62.00	94.50	66.98	93.71
RNN-CNN	63.41	95.33	67.51	93.80
Att-CNN	68.30	95.70	69.10	95.60
Att-RNN	71.00	96.40	69.83	95.51
Att-BLSTM	63.23	95.02	67.60	94.24
Att-CRAN	65.31	96.00	68.43	95.70
BRCAN	73.46	96.81	75.05	96.32

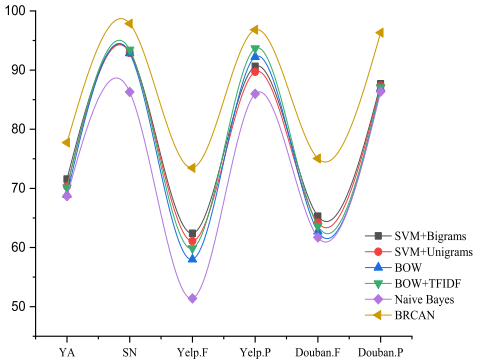


FIGURE 5. Comparison BRCAN with the machine learning methods.

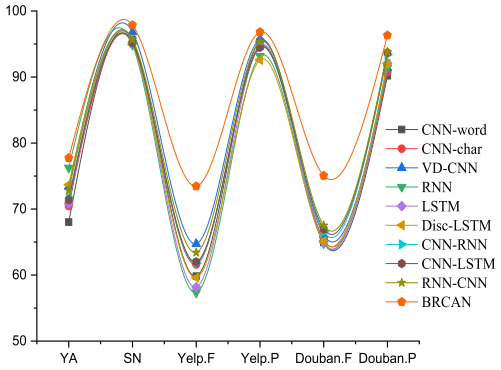


FIGURE 6. Comparison BRCAN with CNN and RNN.

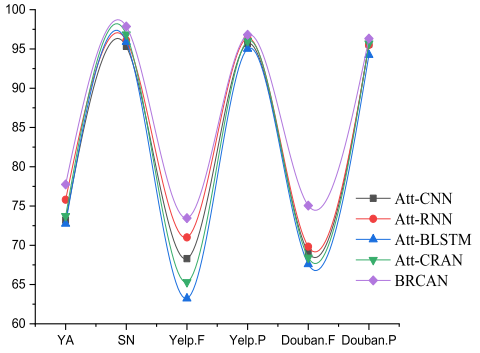


FIGURE 7. Comparison BRCAN with the attention-based methods.

B. COMPARISON WITH THE SIMILAR MODELS

1) COMPARISON BRCAN WITH THE TRADITIONAL MACHINE LEARNING METHODS

The BRCAN and the traditional machine learning method were compared on the above six datasets, and the experimental results are shown in Fig. 5. The deep learning method is superior to the traditional machine learning method in all datasets. It proves that the neural network can effectively compose the semantic representation of texts. Compared with traditional methods based on hand-crafted feature extraction, neural networks can capture more contextual information of features, and may suffer less from the data sparsity problem.

2) COMPARISON BRCAN WITH CNN AND RNN

The BRCAN and CNN, RNN and their variants were compared on the above six datasets, and the experimental results are shown in Fig. 6. BRCAN performs much better than the standalone CNN and RNN. CNN uses a fixed window to capture contextual information through the convolutional layer, and selects more discriminative features through the max-pooling layer. So, the performance of CNN is influenced by the window size. A small window may result in a loss of some long-distance patterns, whereas large windows will lead to data sparsity [37]. For text classification tasks, we need to preserve more detailed and complete information from the input text, and the convolution-only model probably loses detailed local features, so the performance is not as good as BRCAN. Even though VD-CNN increases its number of layers to 29, which can make up for the shortcoming that CNN is difficult to capture long-term dependence. However, this network structure is too complex and involves many parameters, and its performance is still inferior to BRCAN.

RNN can capture long-term dependencies in sentences and get more complete sentence representations. However, RNN pays attention to the time series in sentences, and treats each word in sentences fairly, so it can not extract words that contribute greatly to the text classification. BRCAN not only represents sentences completely, but also recognizes the words with large contributions, so its performance is better than RNN.

In particular, CNN-RNN, CNN-LSTM, RNN-CNN are rather coarse combination manners for unifying CNN and RNN. Although it can improve performance compared with CNN or RNN, the performance of BRCAN is better than them.

BRCAN is similar to the RCNN model, but it adds one more attention layer than RCNN, which is used to add weight to the context information captured by the CNN layer and obtain words that contribute more to text classification. The experimental results show that BRCAN has better classification performance than RCNN.

3) COMPARISON BRCAN WITH THE ATTENTION-BASED METHODS

The BRCAN and the attention-based methods were compared on the above six datasets, and the experimental results are shown in Fig. 7. Adding the attention mechanism to the neural network can significantly improve the accuracy of text classification. Attention mechanism applies on RNN to form a strong classification model, which performs better than most of the existing methods. However, because the context of Att-RNN is ignored, and the CNN layer can provide useful information for BRCAN to pick the important words from the sequences generated by the RNN layer, BRCAN model can perform better than Att-RNN.

BRCAN is also similar to the CRAN model, but the network architecture is different. CRAN takes CNN and RNN as the input of attention layer in parallel, while BRCAN generates intermediate sentence representation by RNN, the CNN layer picks context information based on intermediate sentence representation and inputs it into the attention layer.

The experimental results show that BRCAN has better classification performance than CRAN.

C. THE EFFECT OF BRCAN ON DIFFERENT TYPES OF DATASETS

The experimental results show that the model can train different datasets, and the accuracy of the model is improved compared with the previous best models. The model is effective for fine-grained text classification tasks.

The experiments found that the accuracy of text classification models based on machine learning and deep learning is not significantly different in Sogou News and Yahoo! Answers. The main reason is that the two datasets belong to the topic classification, the differences between the topics are large and the features in the different categories of the topic documents are more obvious and easier to distinguish. However, our proposed BRCAN can still improve the accuracy of the classification and achieve better performance than optimal machine learning and deep learning methods.

Yelp Reviews and Douban movie top250 short reviews belong to sentiment analysis. The comments in these two datasets tend to be more subjective. The emotional polarity of the sentence may be more obvious. But for the same positive five-star and four-star comments, the boundary is not very obvious. The similarity of sentiment words is higher, and the same is true for one star and two stars in negative comments, so it is not easy to distinguish. The languages of sentiment analysis datasets tend to be colloquial, and there may be ambiguous semantics. It is difficult to extract effective features and sentence expressions in classification. These reasons lead to the accuracy of fine-grained sentiment analysis in machine learning and deep learning methods is not high, but the effect of polarity is good. Finally, the experimental results show that the method of deep learning is superior to the traditional machine learning for sentiment analysis, because the methods based on deep learning can learn different features in sentences, capture more complex semantic relations [38] and achieve better classification effect. The BRCAN achieves the highest accuracy for sentiment analysis datasets in deep learning methods.

D. EFFECT OF CONVOLUTIONAL LAYERS NUMBER

The convolutional layer captures contextual information and picks useful local features from the intermediate sentence representation generated by the RNN layer. The intuitive impression is that the more convolutional layers, the better the performance of the model, such as the 29-layer CNN [16]. However, this is not the case in hybrid models. The accuracy of the model does not always increase with the number of convolutional layers. The performance peaks at two or three convolutional layers and decreases if we add more to the model. As more convolutional layers produce longer characters n-gram, this indicates that there is an optimal level of local features to be fed into the attention layer. In BRCAN, we choose three convolutional layers to achieve the best performance.

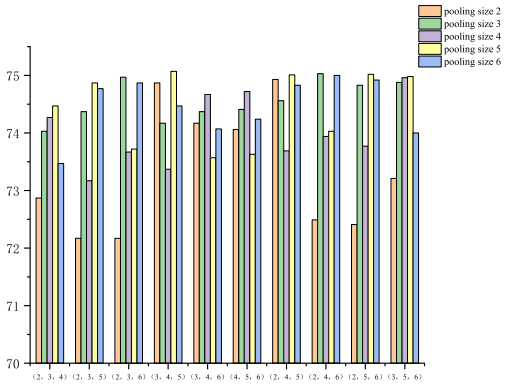


FIGURE 8. Effect of the convolutional filter and max pooling size on BRCAN accuracy.

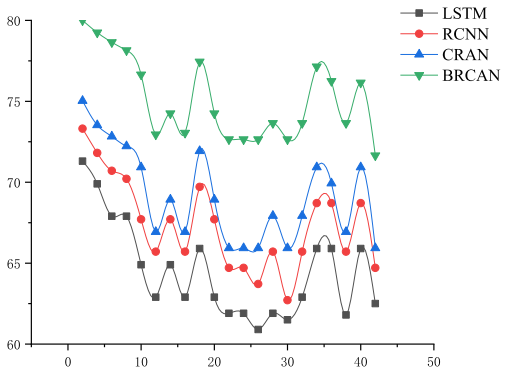


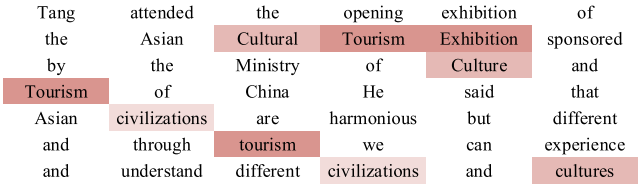
FIGURE 9. Training results of different models for sentence length.

E. EFFECT OF CONVOLUTIONAL FILTER AND MAX POOLING SIZE

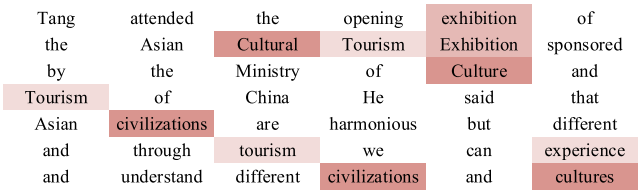
In the process of modeling, it is found that the size of convolutional filters and max pooling will affect the performance of the model while they are either too small or too large. In order to get better performance, we select the most suitable convolutional filters and max pooling size for BRCAN. We conduct experiments on Yelp reviews full dataset with BRCAN and set the number of feature maps to 256. The effect of different convolutional filters and max pooling size on the accuracy of BRCAN as shown in Fig. 8. For the horizontal axis, c means convolutional filter size, and the five different color bar charts on each c represent different max pooling size from 2 to 6. The experimental results show that if a larger filter is used, the convolution can detector more features, and the performance may be improved, too. However, the networks will take up more storage space, and consume more time. Considering comprehensively, we set the size of convolutional filters to be 3, 4, 5 and the max pooling size to be 5.

F. EFFECT OF SENTENCE LENGTH

The different lengths of sentences can also affect the performance of the model. We randomly select 100 sentences from Douban movie top250 short reviews full for training, and the experimental results are shown in Fig. 9. In the figure, the x-axis represents sentence lengths and the y-axis is accuracy.

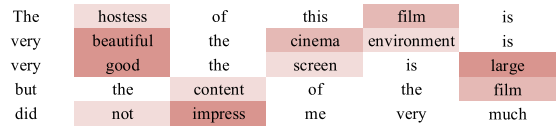


(a) the visual effect map of Att-LSTM predicts the document type is "tourism"

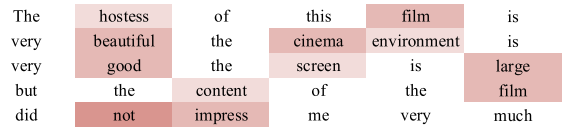


(b) the visual effect map of BRCAN predicts the document type is "culture"

FIGURE 10. The visual effect map of Att-LSTM and BRCAN on Sogou News.



(a) the visual effect map of RCNN predicts the document type is "positive"



(b) the visual effect map of BRCAN predicts the document type is "negative"

FIGURE 11. The visual effect map of RCNN and BRCAN on Douban Movies Top250 short reviews.

The sentences collected are no longer than 45 words, and the value of each data point is a mean score over 5 runs. The results show that the performance of BRCAN is better than LSTM, RCNN and CRAN, which can encode semantically-useful structural information. At the same time, it shows that these models can handle short or moderately long sentences well. If the sentences are too long, the accuracy of the models will decline.

G. EFFECT OF ATTENTION MECHANISM

To further show the effectiveness of our model in selecting informative words in a document, the classification effect of BRCAN is compared with two similar models Att-LSTM and RCNN. Fig. 10 and Fig. 11 are visualized effect maps. We pick two typical examples from the Sogou News and Douban Movies Top250 short reviews, respectively. We group the words into three categories according to their importance for classification. The deeper the color, the greater the weight.

Fig. 10 shows a document picked from the Sogou News. Att-LSTM predicts the document type is ''tourism'', BRCAN predicts the document type is ''culture'', and the real label of the document is ''culture''. The reason that BRCAN can predict the correct label is that it chooses the word ''civilization'' with strong information highly related to ''culture'', while Att-LSTM only focuses on the choice of superficial words such as ''Tourism Exhibition'' and ''Tourism'' which are more related to ''Tourism'' and mislead the judgment.

Fig. 11 shows a comment picked from Douban Movies Top250 short reviews. RCNN predicts a positive comment of 4 stars and BRCAN predicts a negative comment of 2 stars. The comment is actually a negative comment of 1 star. Although neither RCNN nor BRCAN predicts correctly, BRCAN is closer to the correct label. ''not impress '' obtains a rather high weight in BRCAN, while ''beautiful'', ''good'' and '' not impress '' are equal in RCNN. So RCNN leads to the wrong prediction.

From the above examples, we can see the importance of adding an attention mechanism. Each word contributes to text classification differently, so it is necessary to give them different weights. CNN can choose meaningful contextual information, help the attention mechanism to focus on the correct information vocabulary, and help the model make the right decisions.

During the experiment, we can find that there are many factors that affect the performance of the model, including the size of the dataset, the setting of the parameters, the architecture of the network, and the selection of different optimization methods, etc. Therefore, there is no specific model suitable for all types of datasets.

VI. CONCLUSION

In this paper, we propose a hybrid bidirectional recurrent convolutional neural network attention-based model (BRCAN), which combines the Bi-LSTM and CNN effectively with the help of the word2vec model and attention mechanism for fine-grained text classification. As we all know, the proposed model has many advantages: it captures the contextual information and the semantics of long text by Bi-LSTM to alleviate the problem of information imbalance and save the time-step information; it picks higher-level local features useful for classification from the intermediate sentence representation generated by Bi-LSTM according to the context generated by CNN; and fewer parameters are used to obtain the interaction between hidden layer states by applying a bilinear attention function in the attention layer and assign different weights to features according to their importance to text classification. Thus our model reserve the merits of three models in representing a piece of text. We validate the proposed model on multi-topic classification and fine-grained sentiment analysis tasks, and compare it with state-of-the-art classification models based on traditional machine learning and deep learning methods. The experiments results demonstrate that BRCAN not only outperforms traditional machine learning models, but also works better than CNN, RNN or directly combines CNN and RNN. BRCAN achieves state-of-

the-art performance on most of the datasets, especially on the sentiment analysis dataset. In addition, the model proposed in this paper is not only limited to text classification tasks but also can be applied to other applications [39], which is of great significance for future research. By adding different categories of sentiment dictionaries, we can effectively obtain sentiment words in sentences for fine-grained sentiment analysis. Trying to change the way of word vector generation or gradually using the attention mechanism in the network structure will also be a worthy exploration direction in the future.

REFERENCES

[1] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," *Assoc. Comput. Linguistics*, vol. 2, pp. 90–94, Jul. 2012.
[2] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting Assoc. for Comp. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2011, pp. 142–150.
[3] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, nos. 1–2, pp. 1–135, 2008.
[4] G. Paltoglou and M. Thelwall, "A study of information retrieval weighting schemes for sentiment analysis," in *Proc. 48th Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, Jul. 2010, pp. 1386–1395.
[5] L. Qu, G. Ifrim, and G. Weikum, "The bag-of-opinions method for review rating prediction from sparse text patterns," *Assoc. Comput. Linguistics*, vol. 1, pp. 913–921, Aug. 2010.
[6] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP*, vol. 2014, pp. 1746–1751.
[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[8] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," 2016, *arXiv:1605.05101*. [Online]. Available: https://arxiv.org/abs/1605.05101
[9] Y. Xiao and K. Cho, "Efficient character-level document classification by combining convolution and recurrent layers," 2016, *arXiv:1602.00367*. [Online]. Available: https://arxiv.org/abs/1602.00367
[10] A. Hassan and A. Mahmood, "Efficient deep learning model for text classification based on recurrent and convolutional layers," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 1108–1113.
[11] 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.*, Jun. 2016, pp. 1480–1489.
[12] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, Aug. 2016, pp. 207–212.
[13] S. Gao, A. Ramanathan, and G. Tourassi, "Hierarchical convolutional attention networks for text classification," in *Proc. 3rd Workshop Represent. Learn. NLP*, Jul. 2018, pp. 11–23.
[14] C. Wang, M. Zhang, S. Ma, and L. Ru, "Automatic online news issue construction in Web environment," in *Proc. 17th Int. Conf. World Wide Web*, Apr. 2008, pp. 457–466.
[15] X. Zhang, J. Zhao, and Y. Lecun, "Character-level Convolutional Networks for Text Classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 649–657.
[16] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," 2016, *arXiv:1606.01781*. [Online]. Available: https://arxiv.org/abs/1606.01781
[17] O. Quispe, A. Ocsa, and R. Coronado, "Latent semantic indexing and convolutional neural network for multi-label and multiclass text classification," in *Proc. IEEE Latin Amer. Conf. Comput. Intell.*, Nov. 2017, pp. 1–6.
[18] R. Johnson and T. Zhang, "Convolutional neural networks for text categorization: Shallow word-level vs. deep character-level," 2016, *arXiv:1609.00718*. [Online]. Available: https://arxiv.org/abs/1609.00718
[19] H. T. Le, C. Cerisara, and A. Denis, "Do convolutional networks need to be deep for text classification?" in *Proc. Workshops 32nd AAAI Conf. Artif. Intell.*, Jun. 2018, pp. 1–20.
[20] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 3, pp. 517–529, Mar. 2015.
[21] D. Yogatama, C. Dyer, and W. Ling, "Generative and discriminative text classification with recurrent neural networks," 2017, *arXiv:1703.01898*. [Online]. Available: https://arxiv.org/abs/1703.01898
[22] B. Wang, "Disconnected recurrent neural networks for text categorization," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2018, pp. 2311–2320.
[23] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*. [Online]. Available: https://arxiv.org/abs/1508.01991
[24] Q. Hua, S. Qundong, J. Dingchao, G. Lei, Z. Yanpeng, and L. Pengkang, "A character-level method for text classification," in *Proc. 2nd IEEE Adv. Inf. Manage., Communicates, Electron. Autom. Control Conf. (IMCEC)*, May 2018, pp. 402–406.
[25] W. Marinho, L. Martí, and N. Sanchez-Pi, "A compact encoding for efficient character-level deep text classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Oct. 2018, pp. 1–8.
[26] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2015, pp. 1422–1432.
[27] Y. Wang, M. Huang, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Austin, TX, USA, Nov. 2016, pp. 606–615.
[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: https://arxiv.org/abs/1301.3781
[30] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. 9th Int. Conf. Artif. Neural Netw.*, 1999, pp. 850–855.
[31] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Oct. 2013, pp. 6645–6649.
[32] M. J. Er, Y. Zhang, N. Wang, and M. Pratama, "Attention pooling-based convolutional neural network for sentence modelling," *Inf. Sci.*, vol. 373, pp. 388–403, Dec. 2016.
[33] L. Guo, D. Zhang, L. Wang, H. Wang, and B. Cui, "CRAN: A Hybrid CNN-RNN Attention-Based Model for Text Classification," in *Proc. Int. Conf. Conceptual Modeling*. Cham, Switzerland: Springer, 2018, pp. 571–585.
[34] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*. Berlin, Germany: Springer, 1990, pp. 227–236.
[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[36] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.
[37] S. Lai, L. Xu, K . Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification,"Twenty-ninth AAAI conference on artificial intelligence. Austin Texas, USA, Jan. 25-30, 2015.
[38] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp: 562-570.
[39] X. Guo, H. Zhang, H. Yang, L. Xu, and Z. Ye, "A single attention-based combination of CNN and RNN for relation classification," *IEEE Access*, vol. 7, pp. 12467–12475, 2019.
[40] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. Apr. 2015, pp. 4580–4584.

JIN ZHENG was born in Henan, China, in 1994. She received the bachelor's degree in computer science and technology from Xinxiang University, in 2016. She is currently pursuing the master's degree in computer science and technology with China Agricultural University. She is dedicated to the study of artificial intelligence. Her research focuses on the use of deep learning for natural language processing.

LIMIN ZHENG was born in Liaoning, China, in 1962. She received the B.Sc. degree in computer science from Shenyang Ligong University, in 1984. She is currently a Professor with the College of Information and Electrical Engineering, China Agricultural University. Her research interests include artificial intelligence and computer vision.