IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 23, 2019, accepted September 16, 2019, date of publication September 27, 2019, date of current version October 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2944136

Stance Detection of Microblog Text Based on Two-Channel CNN-GRU Fusion Network

WENFA LI[1], YILONG XU[2], AND GONGMING WANG[3]
[1]Robotics College, Beijing Union University, Beijing 100101, China
[2]Smart City College, Beijing Union University, Beijing 100101, China
[3]Inspur Software Group Company Ltd., Jinan 250000, China

Corresponding author: Gongming Wang (gongmingwang@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61841601, in part by the Science and Technology Projects of Beijing Municipal Education Commission under Grant KM201711417011, and in part by the Premium Funding Project for Academic Human Resources Development in Beijing Union University under Grant BPHR2018EZ01.

ABSTRACT In recent years, stance detection has become an important topic in the field of natural language processing. In earlier work, researchers have used feature engineering for stance detection but they need to define and extract appropriate features according to the particular application. This leads to poor generalization and a complex modeling process. Other researchers have applied deep learning methods. However, the popular convolutional neural network (CNN) method has the problem of information loss and a single-size CNN filter cannot accurately extract features that have different lengths from text, and so cannot deal with the diverse nature of features. In order to address these problems, we propose a two-channel CNN-GRU fusion network. First, a convolution layer with two filters with different window sizes is used to extract local features within the topic content and text content. Then, a gated recurrent unit (GRU) network is used to extract their timing characteristics. After that, the intermediate features are spliced and input to a classifier to complete the stance detection. Our method is validated using data from NLPCC 2016. The experimental results show that ACC and average F1 score of this method are 13.1% and 15.6% better than SVM method, 6.2% and 11.6% better than CNN method, 5.6% and 3.3% better than GRU method, and 1.1% and 2.2% better compared with hybrid model proposed by Nanyu, respectively, which is used as a baseline with no increase in run-time, and achieves the same accuracy with less run-time than another baseline of a semantic attention-based model proposed by Zhou. In addition, our method allows better classification than the single channel model. Finally, we find that the operation time of a multi-channel CNN-GRU increases gradually with increasing number of channels, but the classification accuracy does not improve, so a two-channel CNN-GRU is the most appropriate choice.

INDEX TERMS Stance detection, natural language processing, deep learning, CNN, GRU.

I. INTRODUCTION

With the rapid development of the Internet, people can conveniently express their opinions in a variety of ways. Consequentially, a large amount of text data has been generated, which provides a huge corpus and application domain for research into, and application of, text mining technology. The main goal of text mining is to automatically extract valuable information from massive amounts of textual data [1]. Among them, stance detection, also viewed as a subtask of opinion mining and somewhat similar to sentiment analysis, is an important task and has gradually become the primary application in the field of natural language processing [2].

Stance detection involves the analysis of text on a specific topic to determine whether the stances expressed in it are 'favor', 'against', or 'neither' [3]. Its core aim is to detect a theme and explore the polarity of opinions. The main difference from sentiment analysis is that, in stance detection, systems are to determine the author's favorability towards a given target and the target may not even be explicitly mentioned in the text [2]. This means that the relevant feature towards the target in the text needs extracting.

Stance detection has high application value, such as detecting a user's stance on people or things in aspects of politics,

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

economy and society, and understanding the user's information dissemination behavior, etc. It is an important method for conducting public opinion analysis, public opinion surveys, and many other application scenarios.

Feature engineering is a common method of stance detection but how to accurately extract features according to actual requirements is an important and pressing problem. In recent years, deep learning has been shown to enable the automatic extraction of features and has also become widely used in stance detection. However, the single-size CNN filter cannot accurately extract features with different lengths from the text, especially when dealing with complex problems (like stance detection), which cause the problem of information loss. Under this condition, the scope of application for deep learning in stance detection has been restricted. To address these problems, this paper proposes a two-channel CNN-GRU fusion network, which combines multiple CNNs that contain filters of different sizes through the multi-channel with a GRU network. The performance of the proposed network is evaluated using data from the 2016 Natural Language Processing and Chinese Computing (NLPCC 2016) conference. The experimental results show that the new method is more effective than existing methods in terms of average, supported and opposed F1 score.

II. RELATED WORK

Stance detection in user-generated content has been a long-standing problem [4]. Its goal is to determine a user's attitude (affirmative, negative, neutral) on a certain topic based on user-generated content. In recent years, natural language processing researchers have done a great deal of work on stance detection [5], [6], which can be divided into the following two categories [7].

The first kind of stance detection is based on feature engineering. This method relies on designing and extracting features according to the application domain, and then determines the user's stance through a feature training model. Some prominent examples include Somasundaran and Wiebe who designed a stance detection method by establishing a debate corpus, which achieved a better accuracy than the distribution-based baseline [8]. Anand et al. enhanced n-gram feature detection, based on a dictionary and reliant features, and then demonstrated an improvement in detection accuracy [9]. Wojatzki and Zesch used a classifier based on a support vector machine (SVM) to perform stance detection and achieved a higher accuracy than previous methods [10]. Tutek et al. combined machine learning methods with genetic algorithms to carry out stance detection, which ranked 3rd in the Stance Classification Task of SemEval-2016 [11]. In addition, Zhang and Lan used a two-stage method for stance detection. The first stage determines whether the given text is related to the target subject, while the second stage finds the stance of the given text in relation to the target subject [12]. Xu et al. describe an ensemble framework that integrates various feature sets and classification methods, which achieved fourth place in position detection of Nlpcc2016 [13].

However, all these methods need to extract semantic, morphological, and other characteristics, according to the actual requirements and knowledge of the particular field, which make the problem more complicated. In addition, as the selected features are specific to the problem domain, they lack generality. When faced a new topic, it is necessary to re-select the features which requires significant human intervention.

Another kind of stance detection is based on deep learning. This method can automatically extract features and train the model through deep learning, which addresses the problems of high complexity, poor generalization and long operation time in feature extraction. As a result, it has been widely used in recent years. For example, Augenstein et al. used an improved bidirectional long short-term memory (Bi-LSTM) for stance detection [14]. Zarrella and Marsh completed stance detection by combining word2vec and LSTM, and achieved the best results in the stance detection task of SemEval-2016 [15]. Vijayaraghavan et al. proposed a classifier for stance detection based on a word-level or character-level CNN model [16]. Wei et al. trained multiple CNN networks for different target themes for stance detection [17]. In addition, Bai Jing et al. performed stance detection by combining an attention mechanism with LSTM and CNN [7]. Yan et al. show a novel deep-learning-based, fast stance detection framework in bipolar affinities on Twitter to predict the stance of the election-related tweets, and demonstrate the effectiveness of the proposed framework [18]. Wei et al. propose a dynamic memory-augmented network for multi-target stance detection, which uses external memory to capture and store stance-indicative information of multiple targets in text to improve the effect of stance detection [19]. Zhou et al. propose to embed a novel attention mechanism at the semantic level in the bi-directional GRU-CNN structure, and demonstrate its advantages in stance detection tasks [20]. Yu et al. proposed a model based on CNN and LSTM to determine the stance of Weibo automatically, which takes the stance detection task as a classification problem [21]. Lin et al. propose a topic-based approach to detecting multiple standpoints in Web texts that uses the standpoint-related topic-term distributions to enhance a generative standpoint classifier, and contains an adaptive method to determine parameter values. Then, they prove the effectiveness of this method through experiments [22]. Igarashi et al. first collected external resources (e.g., reptile text), and then used a CNN to solve the stance detection task at SemEval-2016, at which the application effect of CNN in position detection was initially discussed [23]. Rajendran et al. compared and discussed the performance of LSTM and GRU in stance detection whose categories are 'Agree', 'Discuss', 'Disagree', and 'Unrelated'. They found that bidirectional LSTM performed best [24]. Sobhani et al. proposed an attention-based encoder-decoder framework that shows better results than other methods in multi-target stance detection [25]. Sun et al. proposed a joint neural network model to predict the stance and sentiment, and proved its effectiveness through experiments [26].

Although this method can address some of the problems encountered in the feature extraction process, it also has some shortcomings that limit its scope of application. The CNN used in the above methods only has a single size filter to extract features from text, which limits the flexibility of mining information and may lead to partial loss of information. To solve this problem, we propose a two-channel CNN-GRU fusion network model to address this problem, which is described in the next section.

III. TWO-CHANNEL CNN-GRU FUSION NETWORK

First, the local convolution features of the subject content and text content that have different lengths are extracted by two convolution layers with different filter window sizes, then the timing features are extracted by the GRU. Second, the output of each GRU is input to a pooling layer where the results of the two channels are fused by concatenation, and a dropout layer is used to reduce overfitting. Finally, the output is sent to a softmax classifier to complete the stance classification. The entire processing flow is shown in Figure 1.

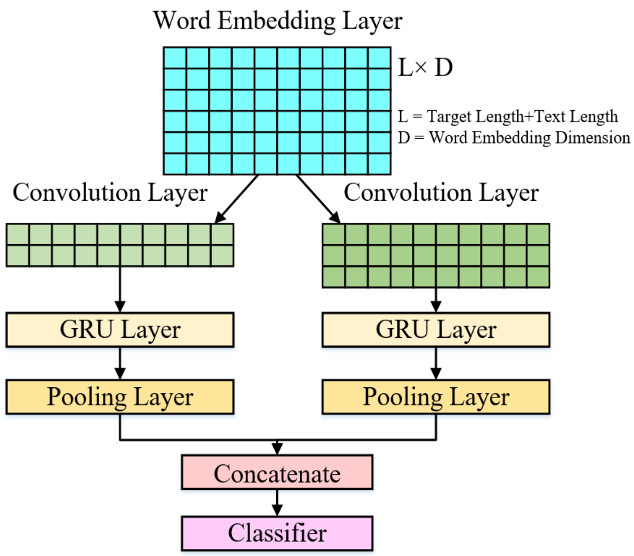

FIGURE 1. Structure diagram of the two-channel CNN-GRU fusion network model.

A. WORD EMBEDDING LAYER

The preprocessed text is expressed in the form of a vector in the embedding layer. Each word in the text is represented as a 1× n matrix and the whole text can be represented as an l ×n matrix, where l is the number of segmented words of text and n is the dimension of each word vector. The text is converted into digital form in this way, which is convenient for the algorithm to extract features.

B. CONVOLUTION LAYER

A convolutional neural network (CNN) [27] is a hierarchical feedforward neural network model, which is based on convolution and pooling operations. Convolution is used to extract local features for further processing by subsequent layers.

First, the output of the embedding layer is convoluted with multiple $n \times h$ filters (convolution kernels), where n is the word vector length of the embedding layer, and h is the filter size. After that, the output of the convolution layer is generated by connecting the results of the convolution operation, which varies with the filter size h.

$$m_i = f(w \cdot x_{i:i+h-1} + b) \quad (1)$$

$$M = [m_1, m_2, m_3, \ldots, m_{l-h+1}] \quad (2)$$

In Formula 1, m_i is the ith feature extracted by the convolution operation, f is a nonlinear function, w is the weight of the filter, h is the filter window size, and b is the bias.

The results of each convolution operation are combined as shown in Formula 2, where M is the output of the convolution layer and l is the number of segmented words in the text.

C. GRU LAYER

Both GRU [28] and LSTM [29] are gated recurrent neural networks (RNN) that can remember long sequences of information and so reduce information loss. Compared with LSTM, the GRU reduces the number of gated units, which decreases the processing time while preserving the accuracy.

A GRU contains two gated sub-units: a reset gate and an update gate. At each moment, the GRU receives the current state and the implicit state from the previous moment through its update gate, this determines the activation state of its own neurons. At the same time, the reset gate receives both of the above states and determines how much of the input information is to be forgotten. The input at the current moment is then combined with the weight and the output of the reset gate to get the memory content at the current moment through the activation function. Then the update gate receives the memory contents at the current moment and the implicit state from the previous moment to determine the output and implicit state at the current moment. The operation of the GRU can be summarized in the following equations.

$$\mathrm{z}_t = \sigma(W^{(\mathrm{z})} \cdot x_t + U^{(\mathrm{z})} \cdot h_{t-1}) \quad (3)$$

$$r_t = \sigma(W^{(\mathrm{z})} \cdot x_t + U^{(\mathrm{r})} \cdot h_{t-1}) \quad (4)$$

$$h_t' = \tanh(\mathrm{W} \cdot x_t + r_t \cdot \mathrm{U} \cdot h_{t-1}) \quad (5)$$

$$\mathrm{h}_t = \mathrm{z}_t \cdot h_{t-1} + (1 - \mathrm{z}_t) \cdot h_t' \quad (6)$$

In Formula 3, z_t is the update gate, $W^{(\mathrm{z})}$ and $U^{(\mathrm{z})}$ are the weights of z_t, σ is the activation function, x_t is the input at the current moment, and h_{t-1} is the implicit output from the previous moment.

In Formula 4, r_t is the reset gate, $W^{(\mathrm{r})}$ and $U^{(\mathrm{r})}$ are the weights of r_t.

In Formula 5, h_t' is the memory content at the current moment, tanh is the activation function, W and U are the weights at the current moment.

In Formula 6, h_t is the output at the current moment.

D. POOLING LAYER

Pooling is a process of extracting information, which aims to reduce the size of the output of the GRU layer. The maximum pooling method [30] is used to extract the maximum vector value of each input as the output of the pooling layer.

$$\hat{m} = \max\{M\} \tag{7}$$
$$z = [\hat{m}_1, \hat{m}_2, \hat{m}_3, \ldots\ldots, \hat{m}_k] \tag{8}$$

In Formulas 7 and 8, M is the characteristic vector from the GRU layer to the pooling layer, $\hat{m}$ is the maximum of M, z is the output result from the pooling layer, and k is the number of characteristics input to the pooling layer.

E. FUSION LAYER

The fusion layer is designed to merge two or more layers or tensors; in this case, it combines the multiple tensors from the polling layer into one tensor. This is implemented by the "Concatenate" method, which takes the last bit as the axis, and splices each output of the polling layer to create the output of this layer [31].

$$b = [a_1, a_2] \tag{9}$$

In Formula 9, b is the output of this layer, a_1, a_2 are outputs from the polling layer.

F. SOFTMAX CLASSIFIER

A softmax classifier [32] is commonly used for multiple classification problems. The fully connected layer is used to connect the input from the fusion layer and the value of each neuron is calculated using the softmax function. In practice, the maximum value of all neuron outputs is taken as the classification result.

$$\hat{p}_y = \mathrm{softmax}(\mathrm{W} \cdot \mathrm{x} + \mathrm{b}) \tag{10}$$
$$\hat{y} = argmax(\hat{p}_y) \tag{11}$$

In Formula 10, $\hat{p}_y$ is the probability that y is the predicted label. W and b are the weight and corresponding bias. In Formula 11, $\hat{y}$ is the finally predicted label.

IV. EXPERIMENT

A. EXPERIMENTAL DATA

The tagged dataset of the Chinese microblog stance detection task from the NLPCC 2016 conference was used to validate our method. There are 3,250 pieces of labeled data in this dataset (2,600 from the training set and 650 from the test set). Each case contains the subject, microblog text, and stance. There are five subject areas: "iPhone SE", "banning Spring Festival firecrackers", "Russia's military action in Syria", "opening the two-child policy", and "Shenzhen ban of motorcycles". The microblog contains text relating to these subjects, such as "considering the problem of air quality, we should use less firecrackers". The stance is divided into three categories: FAVOR, AGAINST, and NONE, which respectively indicate whether the author's stance on the subject is support, against, or other (neither support nor against). The distribution of stances for the different subjects is shown in Table 1.

TABLE 1. Distribution of stances for the different subjects.

SUBJECT	FAVOR	NONE	AGAINST	sum
iPhone SE	240	143	206	589
Spring Festival firecrackers	250	100	250	600
Russia's anti-terrorist action in Syria	250	99	248	597
open the second child	257	140	197	594
Shenzhen ban motorcycle and electrical	485	139	246	870

Overall, the ratios of the number of cases in the training set to the test set is 80% to 20%. The distribution of training and test sets for the different subjects is shown in Table 2.

TABLE 2. Distribution of training set and test set for the different subjects.

TARGET	Training	Test
iPhoneSE	471	118
Spring Festival firecrackers	480	120
Russia's anti-terrorist action in Syria	478	119
open the second child	475	119
Shenzhen ban motorcycle and electrical	696	174
sum	2600	650

B. EXPERIMENTAL SETTING

1) DATA PREPROCESSING

At the beginning of text preprocessing, the subject label and microblog text in the dataset are segmented using the Jieba Chinese word segmentation tool. Numbers and punctuation marks in the text are also removed. After that, the first 180,000 vectorization results in the open-source pre-training word vectors from the Institute of Chinese Information Processing of Beijing Normal University and the Database and Intelligent Information Retrieval (DBIIR) Laboratory of Renmin University of China are used for the vectorization of words in the corpus.

2) PARAMETER SETTING

All parameter matrices and bias values in this experiment were randomly initialized. The remaining parameters are shown in Table 3.

TABLE 3. Main parameter setting in our experiment.

Type	Parameters
Word embedding	300
filters	50
Window size	1,3
GRU layer size	64
Iterations	20
Activation function	Relu
Dropout	0.5

3) EVALUATION INDICATOR

The supported/opposed/average F1 scores provided by the NLPCC were used to evaluate the results, which are calculated as follows.

$$F_{FAVOR} = \frac{2 \times P_{FAVOR} \times R_{FAVOR}}{P_{FAVOR} + R_{FAVOR}} \tag{12}$$

$$F_{AGAINST} = \frac{2 \times P_{AGAINST} \times R_{AGAINST}}{P_{AGAINST} + R_{AGAINST}} \tag{13}$$

$$F_{avg} = \frac{F_{FAVOR} + F_{AGAINST}}{2} \tag{14}$$

In Formula 12, F_{FAVOR}, P_{FAVOR} and R_{FAVOR} are the F1 score, precision, and recall for the supported corpus. In Formula 13, $F_{AGAINST}$, $P_{AGAINST}$ and $R_{AGAINST}$ are the F1-score, precision, and recall of the opposed corpus. In Formula 14, F_{AVG} is the average of F_{FAVOR} and $F_{AGAINST}$. In addition, the commonly used Accuracy (ACC) [13], [22] is selected to evaluate the effect of our method.

4) BASELINE

The proposed method is compared with the following baselines.

A) Zhou [20] proposed a Semantic Attention-based Model (AS-biGRU-CNN), which combines the Bi-GRU network, specific attention layer, and CNN network to complete stance detection. Its architecture is shown in Figure 2. This model also was used as the state-of-the-art comparison in this paper.

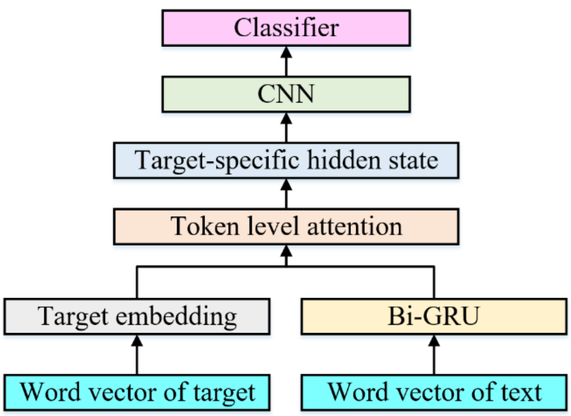

FIGURE 2. Structure diagram of semantic attention-based Model.

By comparing Figure 1 with Figure 2, the differences between the two models can be seen as follows:

A. In the input phase, our method imports the target and text together into the algorithm. However, Zhou's method extracts features from the target and text separately, and then uses an attention mechanism to connect them.

B. Our method uses CNN to extract n-gram features. However, Zhou's method uses Bi-GRU to extract n-gram features. CNN is used to extract features in the hidden state to complete the final prediction.

B) The hybrid model [21] based on the bi-directional long short-term memory (Bi-LSTM) network and CNN as proposed by Nanyu achieved good experimental results on stance detection in NLPCC 2016. Its architecture is shown in Figure 3. This model was used as the state-of-the-art comparison in this paper.

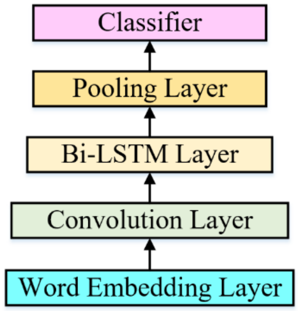

FIGURE 3. Structure diagram of Nanyu hybrid neural network model.

By comparing Figure 1 with Figure 3, the differences between two models can be seen as follows:

A. Our model uses two channels whose convolution layer window size on each channel is different, so different n-gram features from the text can be extracted. Conversely, Nanyu's model only uses one channel, which means it can only extract a single size of n-gram feature.

B. Our model uses GRU to handle the convolution layer output, while Nanyu used a Bi-LSTM.

C. The pooling results from two channels used in our model have to be merged so our model has a fusion layer that is not needed in Nanyu's model.

C. EXPERIMENTAL RESULTS

In order to verify the effectiveness of this method, the following four experiments were carried out: 1) comparison of effectiveness between CNN and single-channel CNN-GRU model, 2) comparison of effectiveness between a single-channel and two-channel CNN-GRU model, 3) comparison of effectiveness between our two-channel CNN-GRU model and several other learning models, and 4) comparison of effectiveness between our method against a CNN-GRU model with different numbers of channels.

1) COMPARISON BETWEEN CNN AND SINGLE-CHANNEL CNN-GRU MODELS

In order to verify the effectiveness of adding a GRU layer to a CNN model, a CNN and single-channel CNN-GRU model with different filter window sizes are generated and compared. The detailed experiments are as follows.

A, CNN-1: A CNN network with a filter window size of 1.

B, CNN-3: A CNN network with a filter window size of 3.

C, CNN-5: A CNN network with a filter window size of 5.

D, CNN-7: A CNN network with a filter window size of 7.

E, CNN-GRU-1: A single-channel CNN-GRU network with a filter window size of 1.

F, CNN-GRU-3: A single-channel CNN-GRU network with a filter window size of 3.

G, CNN-GRU-5: A single-channel CNN-GRU network with a filter window size of 5.

H, CNN-GRU-7: A single-channel CNN-GRU network with a filter window size of 7.

The above experimental results are shown in Table 4. It can be seen that the single-channel CNN-GRU method is better than the CNN on the problem of stance detection, which means that adding a GRU layer to a CNN model may improve precision and solve the problem of information loss when processing time-series data.

TABLE 4. Effect comparison of cnn and single-channel CNN-GRU model with different filter window sizes.

Method	ACC	F_{avg}	F_{FAVOR}	$F_{AGAINST}$
CNN-1	0.544	0.506	0.460	0.552
CNN-3	0.536	0.514	0.465	0.563
CNN-5	0.556	0.513	0.452	0.574
CNN-7	0.525	0.470	0.415	0.526
CNN-GRU-1	0.592	0.593	0.540	0.645
CNN-GRU-3	0.595	0.600	0.553	0.649
CNN-GRU-5	0.595	0.599	0.554	0.644
CNN-GRU-7	0.589	0.585	0.544	0.627

2) COMPARISON BETWEEN SINGLE AND TWO CHANNEL CNN-GRU MODELS

In order to find the optimal CNN-GRU model for this algorithm, a single-channel and a two-channel CNN-GRU model with different filter window sizes were generated. The vectorized text was input to the model and used to extract the characteristics and classify the text. The detailed experiments are as follows.

A, CNN-GRU-1: A single-channel CNN-GRU network with a filter window size of 1.

B, CNN-GRU-3: A single-channel CNN-GRU network with a filter window size of 3.

C, CNN-GRU-5: A single-channel CNN-GRU network with a filter window size of 5.

D, CNN-GRU-7: A single-channel CNN-GRU network with a filter window size of 7.

E, Dual-CNN-GRU-13: A two-channel CNN-GRU network with filter window sizes of 1 and 3.

F, Dual-CNN-GRU-35: A two-channel CNN-GRU network with filter window sizes of 3 and 5.

G, Dual-CNN-GRU-57: A two-channel CNN-GRU network with filter window sizes of 5 and 7.

The above experimental results are shown in Table 5. It can be seen that the two-channel method provides better results than the single channel method on the problem of stance detection. The classification effectiveness of two-channel CNN-GRU fusion networks with different filter window sizes are different. When the window sizes are 1 and 3, the corresponding ACC and F1 score are both the highest, which indicates that the best effectiveness may be achieved.

3) COMPARISON WITH RELATED METHODS

In order to demonstrate the effectiveness of our method, a 5-fold cross-validation was applied to compare it with an SVM, CNN, GRU and the hybrid model developed by Nanyu. In each case, the vectorized text was input to the model and used to extract the characteristics and classify the text.

TABLE 5. Effect comparison of single-channel and two-channel CNN-GRU model with different filter window sizes.

Method	ACC	F_{avg}	F_{FAVOR}	$F_{AGAINST}$
CNN-GRU-1	0.592	0.593	0.540	0.645
CNN-GRU-3	0.595	0.600	0.553	0.649
CNN-GRU-5	0.595	0.599	0.554	0.644
CNN-GRU-7	0.589	0.585	0.544	0.627
Dual-CNN-GRU-13	0.606	0.622	0.589	0.655
Dual-CNN-GRU-35	0.605	0.619	0.583	0.655
Dual-CNN-GRU-57	0.602	0.594	0.550	0.638

A, SVM: The vectorized text is processed with an SVM.

B, CNN: The vectorized text is processed with a CNN.

C, GRU: The vectorized text is processed with a GRU.

D, Zhou-NN: The vectorized text is processed with the AS-biGRU-CNN model developed by Zhan. Since its database is different from ours, some parameters are adjusted as follows.

Hyperparameter: $W_{WindowSize} = 7$, $h_{HiddenSize} = 128$, $e_{word} = 300$, $p_{drop} = 0.5$

AdaGrad parameter: reg $= 10^{-8}$, $\alpha = 0.01$

In the above parameters, the $W_{WindowSize}$ is the filter window size of the CNN layer, which decides how many word embeddings are processed by convolution layer at one time. The $h_{HiddenSize}$ is the hidden layer output size. The e_{word} is the dimensionality of word embedding. The p_{drop} is the probability of dropping out word embedding to speed up processing. The parameters of reg and Î" are used for AdaGrad.

E, Nanyu-NN: The vectorized text is processed with the hybrid model developed by Nanyu. The parameters of this model were as follows.

Hyperparameter: $W_{WindowSize} = 5$, $h_{HiddenSize} = 50$, $e_{word} = 50$, $p_{drop} = 0.5$

AdaGrad parameter: reg $= 10^{-8}$, $\alpha = 0.01$

In the above parameters, the definitions of $W_{WindowSize}$, $h_{HiddenSize}$, e_{word} and p_{drop} are the same as those in the Zhou-NN model.

F, Dual-CNN-GRU: The text vectorization results are processed with the selected model (Dual-CNN-GRU-13).

The comparison results of different methods are shown in Table 6. It can be seen that:

TABLE 6. Comparison results of different methods.

Method	ACC	F_{avg}	F_{FAVOR}	$F_{AGAINST}$
SVM	0.475	0.466	0.449	0.483
CNN	0.544	0.506	0.460	0.552
GRU	0.550	0.589	0.568	0.611
Zhou-NN	0.605	0.619	0.581	0.658
Nanyu-NN	0.595	0.600	0.585	0.615
Dual-CNN-GRU	0.606	0.622	0.589	0.655

1. Compared with traditional machine learning methods, our method has significant advantages because the ACC and average F1 score are increased by nearly 13.1% and 15.6%, respectively. In general, traditional machine learning methods involve feature engineering, which has poor generalization capability. The existing generic features do not perform well in stance detection. Our method can automatically extract features hidden in the data through deep learning, so as to utilize the actual data distribution.

2. Compared with network models with a single structure such as CNN and GRU, the ACCs are improved by 6.2% and 5.6%, and the average F1 values are increased by 11.6% and 3.3%, respectively, which indicate that our method can produce better classifications. Therefore, for the stance detection problem, a better effect may be achieved using a combination of the CNN's ability to extract spatial features and the GRU's ability to extract temporal features.

3. Compared with the neural network of Zhou, we do not use the attention mechanism to fuse the target and text information but use GRU to complete the process. However, both methods have achieved good results, and the ACC and average F1 value of our method are 0.1% and 0.3% higher than those of Zhou-NN, respectively, which may indicate that using attention mechanisms to combine goals and topics is a good approach.

4. Compared with the neural network of Nanyu, the classification is carried out by fusing two CNN-GRU network models with different sizes of filter window. The experimental results show that the ACC and average F1 value of this method are 1.1% and 2.2% higher than that of the neural network developed by Nanyu,respectively, which indicates that our method can improve the effectiveness of stance detection.

In principle, the two channels of our model can extract 1-gram and 3-gram features of microblogs by using filters of size 1 and 3. Nanyu's model has one filter with a size of 5, which means it can only extract 5-gram features. As a result, our model can better express the diversity of word length from Chinese microblogs than Nanyu's model, so it is superior to the Nanyu model in the accuracy of stance detection.

4) COMPARISON WITH MULTI-CHANNEL CNN-GRU MODELS

In order to compare the effect of a different number of channels, the CNN-GRU model with a single-channel, two-channels, three-channels, five-channels, and eight-channels are compared with the Nanyu-NN baseline. In the field of deep learning, with the increase of complexity of the model structure, the running speed will slow down. However, the complexity of the model structure is difficult to quantify numerically, so the time for processing the same data is used to express the speed of this algorithm. The detailed experiments are as follows.

A, Zhou-NN: The trained AS-biGRU-CNN model of Zhou is used to process the test data set.

B, Nanyu-NN: The trained hybrid model of Nanyu is used to process the test data set.

C, CNN-GRU: The vectorized text is processed with a single-channel CNN-GRU network with a filter window size of 5.

D, Dual-CNN-GRU: The vectorized text is input to the two-channel CNN-GRU network with filter window sizes of 1 and 3.

E, Three-CNN-GRU: The vectorized text is input to the three-channel CNN-GRU network with filter window sizes of 1, 2, and 3.

F, Five-CNN-GRU: The vectorized text is input to the five-channel CNN-GRU network with filter window sizes of 1, 2, 3, 4, and 5.

G, Eight-CNN-GRU: The vectorized text is input to the eight-channel CNN-GRU network with filter window sizes of 1, 3, 5, 7, 9, 11, 13, and 15.

The experimental results are shown in Table 7. It can be seen that:

TABLE 7. F_{avg} **and running time of different methods.**

Method	ACC	F_{avg}	Time
Zhou-NN	0.605	0.619	2.00 s
Nanyu-NN	0.595	0.600	0.86 s
CNN-GRU	0.595	0.600	0.86 s
Dual-CNN-GRU	0.606	0.622	1.32 s
Three-CNN-GRU	0.604	0.616	1.76 s
Five-CNN-GRU	0.598	0.612	2.42 s
Eight-CNN-GRU	0.592	0.593	3.65 s

A. The structural difference between the Zhou-NN and the CNN-GRU is that the former uses the attention mechanism. Zhou-NN uses Bi-GRU to extract the target and text feature separately, and then fuses them with the attention mechanism. However, CNN-GRU uses the single GRU to complete the above process directly without an attention mechanism. Therefore, the running-time of CNN-GRU is significantly reduced.

B. The structural difference between the Nanyu-NN and the CNN-GRU is that the former is based on a Bi-LSTM while the latter is based on a GRU. The Bi-LSTM and GRU have similar accuracy but the GRU can reduce the computation time by reducing the number of gated units. Thus, their Favg is almost the same but Nanyu's model has a significantly longer run-time than the CNN-GRU.

C. The run-time of our model is significantly increased compared with the CNN-GRU because of the additional channel but it is still slightly faster than that of Nanyu-NN. Also, since two channels can extract n-gram features of different lengths, its accuracy is better than that of the CNN-GRU and Nanyu-NN.

D. As the number of channels increases, the run time also increases but the accuracy of stance detection does not improve. The reason for this may be that the 1-gram and 3-gram features in the microblog are more suitable for stance detection than other n-gram features. Splicing other n-gram features onto "1-gram" and "3-gram" features will reduce the accuracy of stance detection.

To sum up the above statements, the two-channel model works best for CNN-GRU models with different numbers of channels.

V. CONCLUSION

In the field of stance detection, deep learning methods have become a research hotspot due to their ability to extract features automatically. However, the CNN method has the problem of information loss when dealing with time-series data, and a single size of filter cannot accurately extract features that have different lengths, which limits the application scope of deep learning in stance detection. To solve the above problems, we propose a stance detection method based on a two-channel CNN-GRU fusion network model and validate it with data from NLPCC 2016. The experimental results show that the ACC and average F1 score achieved using this method represents an improvement of more than 13.1% and 15.6%, respectively, compared with a traditional SVM method. The improvements are 6.2% and 11.6% when compared with the CNN method, 5.6% and 3.3% better than the GRU method, and 1.1% and 2.2% better when compared with the baseline model of Nanyu, respectively, while the run-time remains the same, and achieves the same accuracy but less run-time than another baseline model of Zhou's. Therefore, our method has better classification effectiveness than a single structure model. Finally, we found that the run-time increases with increasing filter size but the accuracy is not improved. Therefore, the two-channel CNN-GRU model may be the optimal choice. In the future, an attention mechanism and a wider range of data will be applied to continuously improve and optimize this method. In addition, the development of public opinion analysis based on this method will be another future research direction.

ACKNOWLEDGMENT

The authors would like to thank LetPub (http://www.letpub.com) for its linguistic assistance during the preparation of this manuscript.

REFERENCES

[1] M. Wang, Z. H. Ning, C. Xiao, and T. Li, "Sentiment classification based on information geometry and deep belief networks," *IEEE Access*, vol. 6, pp. 35206–35213, 2018.

[2] T. Hercig, P. Krejzl, and P. Král, "Stance and sentiment in Czech," *Computación y Sistemas*, vol. 22, no. 3, pp. 787–794, 2018.

[3] S. M. Mohammad, P. Sobhani, and S. Kiritchenko, "Stance and sentiment in tweets," *Trans. Internet Technol.*, vol. 17, no. 3, p. 26, 2017.

[4] K. Dey, R. Shrivastava, and S. Kaushik, "Topical stance detection for Twitter: A two-phase LSTM model using attention," in *Proc. Eur. Conf. Inf. Retr.* Cham, Switzerland: Springer, 2018, pp. 529–536.

[5] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "Semeval-2016 task 6: Detecting stance in tweets," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 31–41.

[6] J. Du, R. Xu, Y. He, and L. Gui, "Stance classification with target-specific neural attention networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 3988–3994.

[7] B. Jing and J. Donghong, "Based BILSTM-CNN Chinese microblogging position detection model," *Comput. Appl. Softw.*, vol. 35, no. 3, pp. 266–274, 2018.

[8] S. Somasundaran and J. Wiebe, "Recognizing stances in ideological online debates," in *Proc. NAACL HLT Workshop Comput. Approaches Anal. Gener. Emotion Text*, 2010, pp. 116–124.

[9] P. Anand, M. Walker, R. Abbott, J. E. F. Tree, R. Bowmani, and M. Minor, "Cats rule and dogs drool!: Classifying stance in online debate," in *Proc. 2nd Workshop Comput. Approaches Subjectivity Sentiment Anal.*, 2011, pp. 1–9.

[10] M. Wojatzki and T. Zesch, "Ltl. uni-due at semeval-2016 task 6: Stance detection in social media using stacked classifiers," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 428–433.

[11] M. Tutek, I. Sekulic, P. Gombar, I. Paljak, F. Culinovic, F. Boltuzic, M. Karan, D. Alagić, and J. Šnajder, "Takelab at semeval-2016 task 6: Stance classification in tweets using a genetic algorithm based ensemble," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 464–468.

[12] Z. Zhang and M. Lan, "ECNU at SemEval 2016 task 6: Relevant or not? Supportive or not? A two-step learning system for automatic detecting stance in tweets," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 451–457.

[13] J. Xu, S. Zheng, J. Shi, Y. Yao, and B. Xu, "Ensemble of feature sets and classification methods for stance detection," in *Natural Language Understanding and Intelligent Applications*. Cham, Switzerland: Springer, 2016, pp. 679–688.

[14] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, "Stance detection with bidirectional conditional encoding," 2016, *arXiv:1606.05464*. [Online]. Available: https://arxiv.org/abs/1606.05464

[15] G. Zarrella and A. Marsh, "MITRE at SemEval-2016 task 6: Transfer learning for stance detection," 2016, *arXiv:1606.03784*. [Online]. Available: https://arxiv.org/abs/1606.03784

[16] P. Vijayaraghavan, S. Vosoughi, and D. Roy, "Automatic detection and categorization of election-related tweets," in *Proc. 10th Int. AAAI Conf. Web Social Media*, 2016, pp. 703–706.

[17] W. Wei, X. Zhang, X. Liu, W. Chen, and T. Wang, "Pkudblab at semeval-2016 task 6: A specific convolutional neural network system for effective stance detection," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 384–388.

[18] M.-L. Shyu, Y. Yan, and J. Chen, "Efficient large-scale stance detection in tweets," *Int. J. Multimedia Data Eng. Manage.*, vol. 9, no. 3, pp. 1–16, 2018.

[19] P. Wei, J. Lin, and W. Mao, "Multi-target stance detection via a dynamic memory-augmented network," in *Proc. 41st Int. SIGIR Conf. Res. Develop. Inf. Retr.*, 2018, pp. 1229–1232.

[20] Y. Zhou, A. I. Cristea, and L. Shi, "Connecting targets to tweets: Semantic attention-based model for target-specific stance detection," in *Proc. Int. Conf. Web Inf. Syst. Eng.* Cham, Switzerland: Springer, 2017, pp. 18–32.

[21] N. Yu, D. Pan, M. Zhang, and G. Fu, "Stance detection in Chinese MicroBlogs with neural networks," in *Natural Language Understanding and Intelligent Applications*. Cham, Switzerland: Springer, 2016, pp. 893–900.

[22] J. Lin, Q. Kong, W. Mao, and L. Wang, "A topic enhanced approach to detecting multiple standpoints in Web texts," *Inf. Sci.*, vol. 501, pp. 483–494, Oct. 2019.

[23] Y. Igarashi, H. Komatsu, S. Kobayashi, N. Okazaki, and K. Inui, "Tohoku at SemEval-2016 task 6: Feature-based model versus convolutional neural network for stance detection," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 401–407.

[24] G. Rajendran, B. Chitturi, and P. Poornachandran, "Stance-in-depth deep neural approach to stance classification," *Proc. Comput. Sci.*, vol. 132, pp. 1646–1653, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918308640

[25] Q. Sun, Z. Wang, S. Li, Q. Zhu, and G. Zhou, "Stance detection via sentiment information and neural network model," *Frontiers Comput. Sci.*, vol. 13, no. 1, pp. 127–138, 2019.

[26] P. Sobhani, D. Inkpen, and X. Zhu, "Exploring deep neural networks for multitarget stance detection," *Comput. Intell.*, vol. 35, no. 1, pp. 82–97, 2019.

[27] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 1995. [Online]. Available: https://nyuscholars.nyu.edu/en/publications/convolutional-networks-for-images-speech-and-time-series and https://philpapers.org/rec/LECCNF

[28] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," 2017, *arXiv:1702.01923*. [Online]. Available: https://arxiv.org/abs/1702.01923
[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[30] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Aug. 2011.
[31] L. Chao, J. Tao, M. Yang, Y. Li, and Z. Wen, "Long short term memory recurrent neural network based multimodal dimensional emotion recognition," in *Proc. 5th Int. Workshop Audio/Vis. Emotion Challenge*, 2015, pp. 65–72.
[32] G. E. Hinton and R. R. Salakhutdinov, "Replicated softmax: An undirected topic model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1607–1614.

WENFA LI was born in Henan, China, in 1974. He received the Ph.D. degree in computer software and theory from the Institute of Computing Technology, Chinese Academy of Sciences, China, in 2009. He is currently a Professor with the Software Engineering Department, College of Robotics, Beijing Union University, China. His research interests include the areas of big data, information security, and machine learning.

YILONG XU was born in Beijing, China, in 1993. He received the B.S. degree in computer science and technology from the Beijing Union University, where he is currently pursuing the master's degree. His research interests include the areas of natural language processing and machine learning.

GONGMING WANG was born in Henan, China, in 1981. He received the Ph.D. degree in computer system architecture from the Chinese Academy of Sciences. He held a postdoctoral position at Tsinghua University. He is currently a Researcher with Inspur Software Group Company Ltd. His research interests include the areas of big data and machine learning.