ATLANTIS PRESS

International Journal of Computational Intelligence Systems
Vol. 13(1), 2020, pp. 538–548
DOI: https://doi.org/10.2991/ijcis.d.200423.001; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Special Issue

Bi-GRU Sentiment Classification for Chinese Based on Grammar Rules and BERT

Qiang Lu, Zhenfang Zhu*, Fuyong Xu, Dianyuan Zhang, Wenqing Wu, Qiangqiang Guo

School of Information Science and Electrical Engineering, Shandong Jiao tong University, Jinan, Haitang Road, Changqing District, Shandong 250357, China

ARTICLE INFO

Article History

Received 28 Aug 2019
Accepted 16 Apr 2020

Keywords

Sentiment classification
Grammar rules
BERT
Bi-GRU

ABSTRACT

Sentiment classification is a fundamental task in NLP, and its aim to predict the sentiment polarities of the given texts. Recent researches show great interest in modeling Chinese sentiment classification. However, the complexity of Chinese grammar makes the performance of the existing Chinese sentiment classification model not perform well. In order to address the above problem, we propose a sentiment classification method based on grammar rules and bidirectional encoder representation from transformers (BERT). We first preprocess data through BERT model. Then we combine the Chinese grammar rules with Bi-gated recurrent neural network (GRU) in the form of constraints, and simulate the linguistic functions at the sentence by standardizing the output of adjacent positions. Extensive experiments on two public datasets demonstrate the effectiveness of our proposed method, and our findings in the experiment provide new insights for the future development of Chinese sentiment classification.

1. INTRODUCTION

Sentiment classification, also known as opinion mining, refers to the mining and analysis of viewpoint, sentiment and polarity of texts. Sentiment analysis is an important branch of natural language understanding, involving theories and methods in statistics, linguistics, psychology, artificial intelligence and other fields.

Sentiment classification is widely used in some network platforms of comment mechanism, such as hotel websites. We can analyze the comments of users through sentiment classification technology, and then display them in the comment area according to ranking rules and display proportion. It can provide reference for consumers to choose and provide basis for enterprises to improve product quality. This scene is also applicable to the merchandise evaluation of Amazon, Alibaba, JD and other e-commerce websites.

In recent years, with the rapid development of Internet technology, people have learned more about current affairs, finance, sports and entertainment through the network, and share their lives on the network. This makes the social platforms with convenience, originality and interaction, such as Weibo, popular with the majority of Internet users. At the same time, these social platforms spread quickly, release instantly, especially in emergencies, and became a social media platform for various information releases. Microblog has a large number of potential information data since the influence of the above reasons. Therefore, sentiment classification for social platforms such as microblog has great significance. In addition, sentiment classification technology plays an important role in the analysis of Internet public opinion, which discourse power and influx of Internet users make the environment of the Internet complicated. Using sentiment classification technology could obtain people's views and opinions on an event, and accurately grasp the development trend of public opinion. It will help the government guide the development of network public opinion in the right direction, and promote the healthy development of the network and social environment. At the same time, sentiment analysis in some areas such as election forecasting and stock forecasting has gradually played an important role.

In this paper, we propose a sentiment classification method based on Chinese grammar rules and bidirectional encoder representation from transformers (BERT) pretraining model. We first preprocess data through BERT model, and then combine Chinese grammar rules with Bi-gated recurrent neural network (GRU) model in the form of constraints. We simulate the linguistic functions of nonaffective words, affective words, negative words and degree words at sentence level by standardizing the output of adjacent positions of sentences. Comparing with other deep learning methods, our proposed model has better performance. The contributions of this research are as follows:

- The Chinese grammar rules are integrated into the deep learning model, which solves the problem of the complexity of Chinese grammar rules.
- Using the BERT pretraining model, the accuracy of the classification result is further improved.
- Our proposed method is applicable to sentiment classification in Hotel field, and provides a reference method for sentiment

*Corresponding author. Email: zhuzf@sdjtu.edu.cn

classification in e-commerce field, commentary field and public opinion field. It has good generalization ability.

The following sections are organized as follows: Section 2 presents related work, Section 3 introduces the model we proposed, Section 4 describes the experiments and evaluations and Section 5 presents the conclusion and introduces directions for future research.

2. RELATED WORK

The process of sentiment classification includes four parts: pretreatment, feature extraction, selection of sentiment classification algorithm and sentiment classification evaluation.

Pretreatment is the primary task of sentiment classification, and the main task is to clean data sources and generate datasets for sentiment classification model. Feature extraction is an important task of sentiment classification, and the main task is to convert datasets into feature vectors. Sentiment classification algorithm is the core of sentiment classification, and the main task is to select and apply the appropriate sentiment classification method to process feature vectors and score feature vectors. Sentiment classification evaluation is an important indicator of the accuracy for sentiment classification, and the main task is to evaluate the advantages and disadvantages of the model by comparing the experimental results with different models, which in the same experimental environment.

The first and second parts of this section will focus on the analysis of pretreatment technology and sentiment classification methods.

2.1. Pretreatment Technology

In the pretreatment technology, pretrained word vectors have greatly improved the field of NLP. Word2vec [1], a language model proposed by Google in 2013, has been widely used in the field of NLP due to its efficiency and usability. After that, Word embedding technology, which preprocesses a large number of unmasked data by word2vec and GloVe [2], has been gradually applied to the first layer of initialization neural network. In most tasks with limited training data, word embedding technology has achieved great results.

There are many methods to generate pretrained word vectors, which are mainly divided into statistical method and language model method. Among these methods, the language model method has been used by more researchers. The traditional word embedding techniques based on language model mainly include Word2Vec and GloVe. Word2Vec has two training methods. One is called CBOW [3], the core idea is to remove a word from a sentence, and use the above and below of this word to predict the word which will be removed. The second is Skip-gram, its opposite to CBOW, the core idea is to input a word and ask the network to predict its context words. Global Vectors for Word Representation (GloVe) is a word representation tool based on global count and overall statistics, it could express a word as a vector composed of real numbers, these vectors capture some semantic characteristics between words, such as similarity and analogy. By computing vectors, such as Euclidean distance or cosine similarity, we can calculate the semantic similarity between two words.

With the development of deep learning research, neural network has been gradually applied to word embedding technology, which Embeddings from Language Models (ELMo) [4] word embedding technology is a representative one. ELMo model is learned from the internal state of the deep bidirectional language model, and these word vectors can easily be added to automatic question and answer, text alignment, text categorization and other models. ELMo model has two characteristics, one is that the input of ELMo model is character rather than word, so they can use the advantage of subunit to calculate meaningful word representation, even though these words may be outside the vocabulary. Secondly, ELMo is a series of excitation functions on some layers for the bidirectional language model. Different layers of a language model encode different types of information. For example, part-of-speech tagging is well predicted by the lower level of Bi-LSTM, while word sense disambiguation is better coded by the higher level. By concatenating all the layers, it is possible to freely combine different word representations so as to achieve better model performance in downstream tasks.

In the research of ELMo model, Che *et al.* [5] applied ELMo model to HIT-SCIR system which is a CoNLL-2018 shared task system for general dependency analysis. The system embedded deep contextualized word embeddings into both the part of speech tagger and parser, and used different set parsers with initialization training. Experimental results on the development data show the effectiveness of methods. In the final evaluation, HIT-SCIR system was ranked first according to LAS (75.84%) and outperformed the other systems by a large margin. Suzana *et al.* [6] proposed a model that uses character-level vector representations of words, which based on ELMo. This model can capture complex morpho-syntactic features and use them as indicators for irony or sarcasm across dynamic contexts. The experimental results show that model is tested on 7 different datasets derived from 3 different data sources, providing state-of-the-art performance in 6 of them, and otherwise offering competitive results.

2.2. Sentiment Classification Method

Traditional sentiment classification methods mainly include sentiment classification method based on dictionary and sentiment classification method based on machine learning.

The core of classification method based on sentiment dictionary is the combination of dictionary and rules. This method obtains a large number of sentiment words, it gives weight by weighting algorithm and constructs a sentiment dictionary, then it calculates the sentiment score of sentences by using specific formula with syntactic rules, and finally predict the sentiment polarity. Therefore, the construction of sentiment dictionary is the core of this method. In recent years, many representative achievements have been made in the field of English sentiment dictionary construction and Chinese sentiment dictionary construction. In the construction of English sentiment dictionary, Hu *et al.* [7] argued that the main factor affecting the sentiment polarity of the text in comments is the polarity of adjectives. He created a sentiment dictionary with adjectives as its core, evaluating the emotional polarity of the review text based on the polarity in the sentiment dictionary, and proposing to judge the polarity of adjectives in comments by the synonyms and antonyms of adjectives in WordNet [8].

In the construction of Chinese sentiment dictionary, Zhang *et al.* [9] proposed a sentiment analysis method of Chinese Weibo text based on sentiment dictionary. First, the sentiment dictionary was expanded by extracting and constructing degree adverb dictionary, network dictionary, negative dictionary and other related dictionaries, and then the sentiment value of Weibo text was obtained by calculating the weight. Finally, Weibo text is divided into positive, negative and neutral. The experimental results show the effectiveness of this method.

The core of machine learning classification method is feature engineering. This method extracts the lexical, grammatical and semantic features of text, then trains the classifier model with corpus text data. Finally, text that needs to be analyzed is input into the trained model to obtain the sentiment classification results. The most commonly used feature methods include N-Gram feature [10], POS feature [11], TF-IDF feature [12], and most commonly used classification methods include KNN [13], Naive Bayesian [14], SVM [15]. Pang *et al.* [16] pioneered the application of machine learning in sentiment classification. They classified the sentiment data of movie reviews by Naive Bayes, Maximum Entropy Classification [17] and support vector machine (SVM) models respectively. Zhang *et al.* [18] proposed a selective ensemble algorithm based on machine learning BPSO and stochastic subspace method, which solved the problem of high dimensionality of eigenvector space and improved the accuracy and generalization ability of sentiment classification.

With the deepening of the research on deep learning, sentiment classification methods begin to gradually transit to deep learning methods such as convolutional neural network (CNN) method and recurrent neural network (RNN) method.

CNN [19] is a kind of feed forward neural network (FNN) [20], which contains convolution computation and has deep structure. It consists of input layer, convolution layer, pooling layer and full connection layer. CNN firstly obtains data through the input layer, then extracts data features through the convolution layer and reduces the dimension of extracted features through the pooling layer to obtains local optimal solution, finally the results are output through the full connection layer. For example, Li *et al.* [21] proposed a fine-grained sentiment analysis model based on multi-granularity convolution kernel CNN, which incorporates attribute features into Word2Vec model and constructs text feature vectors. Experiments show that fine-grained sentiment analysis method based on multi-granularity convolution kernel CNN can further improve the accuracy of sentiment classification.

RNN [22] is a kind of recursive neural network, which takes sequence data as input and recursively connects all nodes (cyclic units) in the direction of sequence evolution. It consists of input layer, hidden layer and output layer. RNN has memory function, and the input of each sequence is related to its previous output. It can remember the output of the hidden layer at one moment and take the output of the previous moment and the input of the current unit as the input of the current hidden layer. For example, Abdi *et al.* [23] proposed a new sentiment classification method based on deep learning, RNSA, which used an RNN composed of long- and short-term memory (LSTM) to overcome some shortcomings of traditional methods.

This paper focuses on the influence of pretreatment technology and sentiment classification methods for Chinese sentiment classification. Therefore, we will describe the proposed model in the latter part. The first part of the third section focuses on the analysis of BERT model and Transformer model. The second part focuses on the analysis of GRU model and the third part focuses on the analysis of Bi-GRU Chinese sentiment classification model based on grammar rules and BERT.

3. BI-GRU CHINESE SENTIMENT CLASSIFICATION BASED ON GRAMMAR RULES AND BERT

Grammar rules are very important to the task of Chinese sentiment classification. Chinese grammar rules mainly include nonemotional words, sentiment words, negative words and degree words. In Chinese sentences, the modified words of negative words and degree words are usually at the left side of the modified, but sometimes the modified words are at the right side of negative words and degree words. In order to better address this issue, we first employ Bi-GRU model to let itself decide which direction should start from. Then, in order to improve the accuracy of the model, we apply the BERT Chinese pre-training language model into our model. The word vector could better represent the polysemy of words with BERT model, and enhance the semantic representation of sentences.

3.1. BERT Chinese Pretraining Model

In recent years, researchers have been used pretrained neural networks as language models, and on this basis, fine-tuned downstream tasks have achieved good results. For example, Devlin *et al.* [24] proposed a BERT pretraining language model. The model achieved good results which compared with the traditional N-gram language model [9], and made better use of the context information of words and got better expression of word vectors.

3.1.1. BERT model

BERT model uses Bidirectional Transformer [25] as encoder, which can have deeper layers and better parallelism, The linear Transformer is not susceptible to mask marker, and simply reduce the weight of mask tags with self-attention. The model innovatively puts forward two tasks: Masked Language Model and Next Sentence Prediction, which capture the representation of word level and sentence level respectively, and conduct joint training. The structure of the model is shown in Figure 1.

- Masked language model
  The role of the Masked Language Model is to train deep bidirectional language representation vectors, which randomly obscure certain words in the sentence and let the encoder predict the original vocabulary of those words. In the model, 15% of the words were randomly occluded as training samples, 80% of them were replaced by masked token, 10% were replaced by a random word, and 10% remained unchanged.
- Next sentence prediction
  Next sentence prediction refers to pretraining a binary classification model to learn the relationship between sentences.

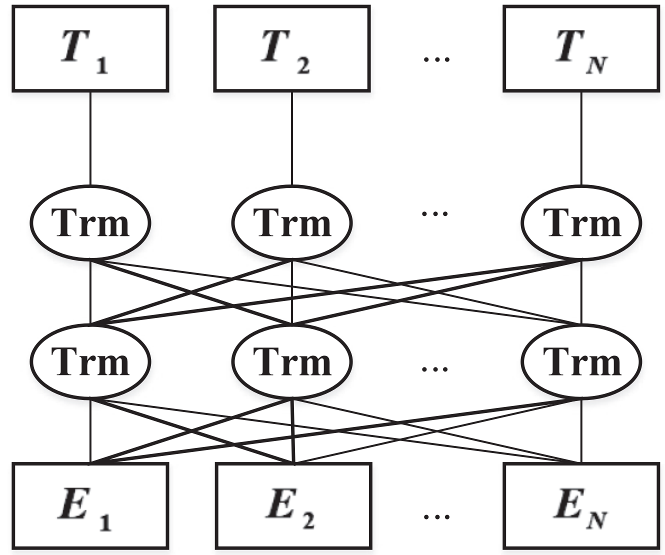

Figure 1 | Bidirectional Encoder Representation from Transformers (BERT) model.

In the many NLP tasks, such as QA and NLI, the language model needed to understand the relationship between two sentences, but it could not directly produce this understanding. In order to understand the relationship between sentences, this method pretrains a task of next sentence prediction. The specific method is to replace some sentences randomly, and then use the previous sentence to predict Is Next/Not Next.

3.1.2. Transformer model

The core part of BERT is the bidirectional Transformer model, which uses Encoder–Decoder [26] framework and relies entirely on attention mechanism to construct the global dependence between input and output. Transformer model can be divided into two parts: Encoder and Decoder. Encoder is stacked by multiple Encoders with the same structure, and decoder is stacked by multiple Decoders with the same structure. Each Encoder can be decomposed into two sub-layers: self-attention layer and feed-forward neural network layer. The sentence input from Encoder first passes the self-attention layer, which helps the Encoder to focus on the other words of the input sentence when encoding each word, and then the output from the self-attention layer is transmitted to the FNN layer. The FNN corresponding to each word in each position is exactly the same. In addition to self-attention layer and FNN layer of Encoder, there is an Encoder–Decoder attention layer between the two layers to focus on the relevant parts of the input sentence. The structure of the layer is shown in Figure 2.

The core module of Encoder unit is Self-Attention module, the Self-Attention module takes into account all other words in a sentence when encoding a word, and determines how to code the current word. The calculation method is as shown in function (1):

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

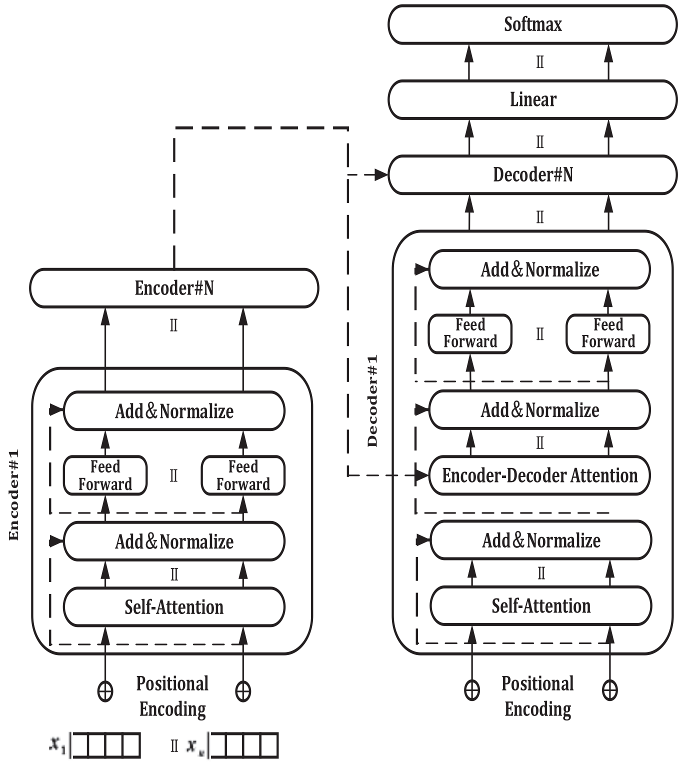

Figure 2 | Transformer model.

The vectors Q, K and V represent query vectors, Key vectors and Value vectors respectively. Q vector means that we need to pay attention to other words (including ourselves) when coding the current word, so we need to have a query vector. Key vector can be considered as the key of the word for the information to be retrieved, and Value vector is the real content. The core idea of attention module is to calculate the relationship between each word in a sentence and all the words in the sentence, which reflects the relevance and importance of different words in the sentence. Then, we can get a new representation of words with adjusting the weight of each word. This new representation not only implies the word itself, but also the relationship between other words and the word, so it is a more global expression than a simple word vector.

Transformer model further improves the self-attention layer by adding a mechanism called "multi-head" attention, and improves the performance of the attention layer in two aspects. On one hand, it expands the ability of the model to concentrate on different positions, on the other hand, it gives multiple "representation subspaces" of the attention layer, and the calculation methods are as shown in functions (2) and (3):

$$MultiHead(Q,K,V) = Concat(head_1, ..., head_h)W^O \tag{2}$$

$$where\ head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{3}$$

where W_i^Q, W_i^K, W_i^V and W^O are weight matrices and $W_i^Q \in \mathbb{R}^{d_{\text{model}}\times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}}\times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}}\times d_v}$, $W^O \in \mathbb{R}^{hd_v\times d_{\text{model}}}$. In order to alleviate the problem of gradient disappearance caused by increasing depth in neural networks, the Transformer model adds residual connections [27] and layer normalization [28] in each Self-Attention layer, as shown in functions (4) and (5):

$$LN(x_i) = \alpha \times \frac{x_i - \mu_L}{\sqrt{\sigma_L^2 + \varepsilon}} + \beta \tag{4}$$

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{5}$$

Since the model contains no recurrence and no convolution, in order to make use of the order of the sequence, the model adds positional encodings to the input embeddings at the bottoms of the encoder and decoder stacks. The positional encodings have the same dimension as the embeddings, so that the two can be summed. There are many choices of positional encodings, learned and fixed. As shown in functions (6) and (7), we use sine and cosine functions of different frequencies to represent position information.

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \tag{6}$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{\text{model}}}\right) \tag{7}$$

3.2. Gated Recurrent Neural Networks

GRUs [29] is a variant of RNN. It combines short-term memory with long-term memory through delicate gate control, and solves the problem of gradient disappearance to a certain extent. Compared with LSTM [30], GRU can achieve considerable results, and it is easier to train than LSTM, which can greatly improve the training efficiency.

The input and output structure of GRU is the same as that of ordinary RNN. There is a current input x_t and a hidden state h_{t-1} passed down by the previous node, which contains information about the previous node. Combined x_t with h_{t-1}, GRU will get the output h_t of the current hidden node. Compared with the gate of LSTM, GRU has only two gated units, which are reset gate and update gate, as shown in Figure 3.

In the first step, GRU obtains two gated states r_t and z_t through last transmitted state h_{t-1} and input x_t of the current node.

$$r_t = \sigma\left(w_r \cdot [h_{t-1}, x_t]\right) \tag{8}$$

$$z_t = \sigma\left(w_z \cdot [h_{t-1}, x_t]\right) \tag{9}$$

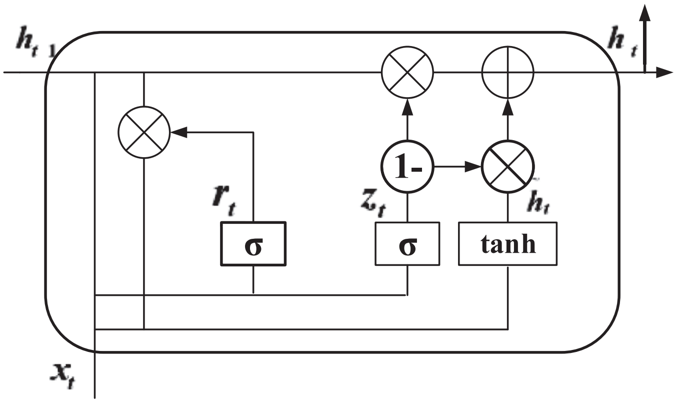

Figure 3 Gated Recurrent Neural Network (GRU) model.

Among them, r_t is the gated control of reset. z_t is the gated control of update, w_r is the weight of reset gate, w_z is the weight of update gate, σ is the sigmoid function. From this function, the data can be converted into values in the range of 0–1, which would act as the gated signal.

In the second step, the reset gate r_t is used to get the reset data $h_{t-1}^{'}$, then, $h_{t-1}^{'}$ and input x_t are spliced together. The data is scaled to $-1 \sim 1$ by a *tanh* activation function, which is defined as follows:

$$\tilde{h}_t = tanh\left(w_{\tilde{h}} \cdot \left[h'_{t-1}, x_t\right]\right) \tag{10}$$

$$h'_{t-1} = h_{t-1} \cdot r_t \tag{11}$$

$\tilde{h}_t$ mainly contains the data of the current input x_t. Targeted addition of $\tilde{h}_t$ to the current hidden state is equivalent to remembering the state of the current moment, similar to the selective memory stage of LSTM.

The third step is the most critical step in GRU, which named memory updating stage. At this stage, we have two steps of forgetting and remembering at the same time. We use the previously obtained update gated state function and the update function are as follows:

$$h_t = z_t \cdot \tilde{h}_t + (1 - z_t) \cdot h_{t-1} \tag{12}$$

The range of updated gating signal is 0~1, the gated signal is much closer to 1, the more data is remembered. The gated signal is much closer to 0, the more forgotten it is. We use the same gate to make forgetting and selecting memories at the same time. $(1 - z_t) \cdot h_{t-1}$ means selective forgetting of the original hidden state, z_t could be regarded as forget gate and forget some unimportant information in h_{t-1}, $z_t \cdot \tilde{h}_t$ means that the update gate selectively memorizes $\tilde{h}_t$ which contains the information of the current node, z_t should be regarded as selective memory of some information in $\tilde{h}_t$.

Traditional GRU model could only process data from forward sequence, and reverse sequence is also very important in text sentiment classification. Bi-GRU model first combines forward GRU with reverse GRU, and then uses two independent hidden layers to process data from forward and reverse simultaneously. This structure make the model learn more context information and improve the classification accuracy.

3.3. Bi-GRU Chinese Sentiment Classification Model Based on Grammar Rules and BERT Pretraining Model

We propose a Bi-GRU Chinese sentiment classification model based on grammar rules and BERT in this paper. We first preprocess data through BERT model, then we construct four kinds of grammar rule regulators in the form of regularization, including nonsentiment word regulators, sentiment word regulators, negative word regulators and degree word regulators. The core idea of the model is simulating the linguistic functions at sentence by standardizing the output of adjacent positions of sentences. Figure 4 shows the construction process of the Bi-GRU model based on grammar rules and BERT.

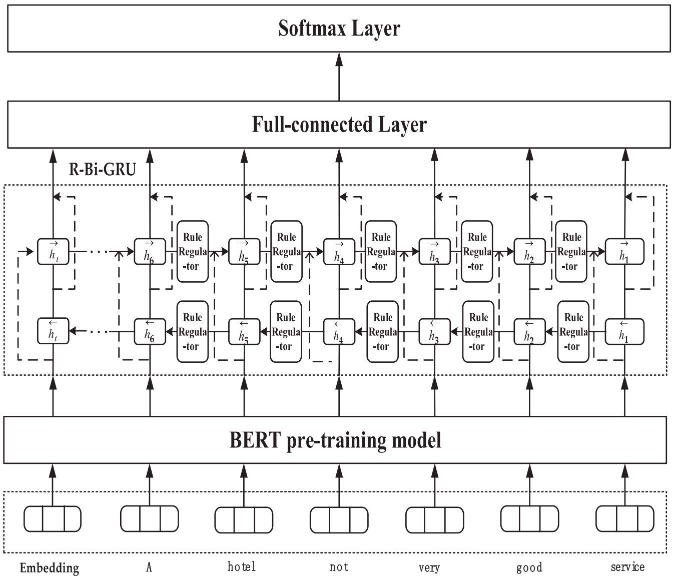

Figure 4 Bi-gated recurrent neural networks (GRU) model combining grammar rules and Bidirectional Encoder Representation from Transformers (BERT).

The construction process of Bi-GRU model based on Chinese grammar rules and BERT is divided into three steps. The first step is to transform the input data into word vectors with BERT, and then input the word vectors into model. The second step is to add Chinese grammar rules into the model in the form of regularization constraints. The third step is to judge the part of speech for the input word vectors with the grammar rule regulator. Then, the linguistic functions at sentence are simulated by standardizing the output of adjacent positions.

This paper gives an example of Bi-GRU model based on Chinese grammar rules and BERT. As shown in Figure 4, in the sentence "A hotel with not very good service," the predicted sentiment distribution of "A hotel with not very good service" and "hotel with not very good service" should be close to each other, and the predicted sentiment distribution of "good service" and "not good service" should be completely different. Therefore, four kinds of grammar rule regulators are proposed in this paper.

Nonsentiment word regulator: If adjacent words are nonsentiment words, the sentiment distribution of adjacent positions should be similar. This rule is applicable in most cases.

Sentiment word regulator: If the current word is a sentiment word, the sentiment distribution of the current position should be significantly different from the next or previous position. We use sentiment transfer to deal with this phenomenon.

Negative word regulator: If the current word is a negative word, the sentiment polarity of the current position will be reversed than adjacent words, such as polarity from positive to negative and negative to positive, but sometimes it depends on the words they modify. The negative word regulator uses a specific transformation matrix to simulate the language performance.

Degree word regulator: If the current word is a degree word, the sentiment polarity of the current position will be enhanced or weakened than adjacent words. For fine-grained sentiment classification, it is very important to model with degree words. The degree word regulator uses a specific transformation matrix to simulate the language performance.

i. **Nonsentiment word regulator**
The rule of non-sentiment word regulator: If the words of adjacent position are nonsentiment words, the sentiment distribution of the adjacent positions should be similar. The nonsentiment word regulator is combined into the model in the form of constraints. It has the following definitions:

$$\overrightarrow{R}_t^{(ur)} = max\left(0, JS\left(\overrightarrow{p}_t \,||\, \overrightarrow{p}_{t-1}\right) + M\right) \tag{13}$$

$$\overleftarrow{R}_t^{(ur)} = max\left(0, JS\left(\overleftarrow{p}_t \,||\, \overleftarrow{p}_{t+1}\right) + M\right) \tag{14}$$

where M is the boundary parameter, p_t is the predicted sentiment distribution at position t (predicted distribution based on h_t in function (12)), $JS\left(p_t \,||\, p_{t-1}\right)$ is Jensen–Shannon (JS) divergence [31], $JS\left(p_t \,||\, p_{t-1}\right)$ measures the similarity of two probability distributions. The features of JS divergence are as follows:

1. The range of JS divergence is [0, 1], and the predicted distribution is the same as 0, but the opposite is 1.
2. JS divergence possesses the character of symmetry.
When the words of adjacent position are nonsentiment words, the sentiment distribution of the two words will be closer, and according to functions (13) and (14), the value of nonsentiment word regulator is 0. If the sentiment distribution of two adjacent words is much different, the value of nonsentiment word regulator will become larger.

ii. **Sentiment word regulator**
The rule of sentiment word regulator: If the word of the current position is a sentiment word, the sentiment distribution of the current position should be significantly different from that of the adjacent position.

As shown in Figure 4, in the sentence "A hotel with not very good service," when position $t = 2$ (reverse), the input word is "good" and the predicted sentiment distribution is more positive than the first word "service." In order to simulate this phenomenon, we deal with the sentiment transfer for each category of words in the sentiment dictionary. Each category of sentiment words has its corresponding sentiment transfer distribution V_s, which is acquired through model learning. The sentiment word regulator is combined into the model in the form of constraints, which is defined as follows:

$$\overrightarrow{p}_{t-1}^{(sr)} = p_{t-1} + V_{s(t)} \tag{15}$$

$$\overleftarrow{p}_{t+1}^{(sr)} = p_{t+1} + V_{s(t)} \tag{16}$$

$$\overrightarrow{R}_t^{(sr)} = max\left(0, JS\left(\overrightarrow{p}_t \,||\, \overrightarrow{p}_{t-1}^{(sr)}\right) + M\right) \tag{17}$$

$$\overleftarrow{R}_t^{(sr)} = max\left(0, JS\left(\overleftarrow{p}_t \,||\, \overleftarrow{p}_{t+1}^{(sr)}\right) + M\right) \tag{18}$$

where $\overrightarrow{p}_{t-1}^{(sr)}$ and $\overleftarrow{p}_{t+1}^{(sr)}$ are the sentiment distribution of location t with transformation, $s(t)$ is the former sentiment category of location t.

If the current position is a sentiment word, the sentiment distribution of the adjacent position plusses sentiment transfer distribution will be close to the current location. And according to functions (17) and (18), the value of the sentiment word regulator is 0. If the sentiment distribution is far away, the value of the sentiment word regulator increases.

iii. **Negative word regulator**
The influence of a negative word on a text depends on the sentiment polarity of the word it modifies. In general, negative words will reverse the sentiment polarity of words which they modify. For example, in the sentence "not good," the positive polarity of the modifier "good" is transformed into negative polarity, while in the sentence "not bad," the negative polarity of the modifier "bad" is transformed into neutral polarity.

The rule of negative word regulator: If the word of current position is a negative word, the sentiment distribution of the adjacent position with the treatment of negative matrix T_m should be close to that of the current position. The negative matrix T_m is learned by model and optimized by model training. At most times, the modified words of negative words are usually at the right side of the modified text. But sometimes, the modified words are at the left side of negative words. To better solve this issue, we apply Bi- GRU and let the model determine which side should be chosen. In Bi-GRU, we compute a transformed sentiment distribution on $\vec{p}_{t-1}$ of the forward GRU and also that on $\overleftarrow{p}_{t+1}$ of the backward GRU, and compute the minimum distance of the distribution of the current position to the two distributions, The process can be formulated as follows:

$$\vec{p}_{t-1}^{(nr)} = softmax\left(T_{m_i} \times \vec{p}_{t-1}\right) \tag{19}$$

$$\overleftarrow{p}_{t+1}^{(nr)} = softmax\left(T_{m_i} \times \overleftarrow{p}_{t+1}\right) \tag{20}$$

$$R_t^{(nr)} = min \begin{cases} max\left(0, JS\left(\vec{p}_t \,||\, \vec{p}_{t-1}^{(nr)}\right) + M\right) \\ max\left(0, JS\left(\overleftarrow{p}_t \,||\, \overleftarrow{p}_{t+1}^{(nr)}\right) + M\right) \end{cases} \tag{21}$$

where $\vec{p}_{t-1}^{(nr)}$ and $\overleftarrow{p}_{t+1}^{(nr)}$ are the sentiment distribution by the processing of negative matrix. T_m is the negative matrix for each negative word m. *softmax* represents the normalized exponential function. To train the model, one first needs to calculate the predicted sentiment distribution for each node, and then calculate the posterior probability of each node.

iv. **Degree words regulator**
Degree words change the sentiment intensity of text. It could change weak positive to strong positive and weak negative to strong negative, which is very important for fine-grained sentiment classification.

The rule of degree word regulator: If the word of current position is degree word, the sentiment distribution of the adjacent position with the treatment of degree matrix A_k should be close to that of the current position. The degree matrix A_k is learned by model and optimized by model training. At most times, the modified words of degree words are usually at the right side of the modified text. But sometimes, the modified words are at the left side of degree words. To better solve this issue, we apply Bi- GRU and let the model determine which side should be chosen. In Bi-GRU, we compute a transformed sentiment distribution on $\vec{p}_{t-1}$ of the forward GRU and also that on $\overleftarrow{p}_{t+1}$ of the backward GRU, and compute the minimum distance of the distribution of the current position to the two distributions, The process can be formulated as follows:

$$\vec{p}_{t-1}^{(dr)} = softmax\left(A_{k_j} \times \vec{p}_{t-1}\right) \tag{22}$$

$$\overleftarrow{p}_{t+1}^{(dr)} = softmax\left(A_{k_j} \times \overleftarrow{p}_{t+1}\right) \tag{23}$$

$$R_t^{(dr)} = min \begin{cases} max\left(0, JS\left(\vec{p}_t \,||\, \vec{p}_{t-1}^{(dr)}\right) + M\right) \\ max\left(0, JS\left(\overleftarrow{p}_t \,||\, \overleftarrow{p}_{t+1}^{(dr)}\right) + M\right) \end{cases} \tag{24}$$

where $\vec{p}_{t-1}^{(dr)}$ and $\overleftarrow{p}_{t+1}^{(dr)}$ are the sentiment distribution by the processing of degree matrix. A_k is the degree matrix for each degree word k. *softmax* represents the normalized exponential function. To train the model, one first needs to calculate the predicted sentiment distribution for each node, and then calculate the posterior probability of each node.Finally, in order to maintain the continuity of model prediction, we fuse four rule regulators into the loss function of Cross Entropy:

$$L = -\sum_i p_i \log q_i + \sum_i \sum_t R_{t,i} + \lambda ||w||^2 \tag{25}$$

where p_i is the actual distribution of samples, q_i is the predictive distribution of samples, $R_{t,i}$ is one or more combinations of four rule regulators at sentence i, t is the position of words in sentences, $\lambda ||w||^2$ is the regularization term.

4. EXPERIMENT AND EVALUATION

In this section, we first describe the experimental dataset. Then, we describe the experimental details. Afterward, we introduce the several frontier models as baseline models. Finally, we analyze the experimental results to prove the performance of our proposed model.

4.1. Dataset

We use two datasets to evaluate the performance of our model, which were collected by Tsinghua University Chinese Hotel Comment Corpus and Professor Tan Songbo. The Sentiment Dictionary uses the Chinese Sentiment Vocabulary Ontology Library of DUTIR. There are 22012 sentiment vocabularies, which are divided into 7 categories and 21 subcategories. Each category of vocabulary has the sentiment polarity score, including 11229 positive sentiment words and 10783 negative sentiment words. Negative words and degree words are collected by HowNet Chinese Negative Words Dictionary and Degree Words Dictionary. There are 59 negative words and 219 degree words. The distribution is shown in Table 1.

4.2. Experimental Details

All the experiments in this paper were performed in the experimental environment shown in Table 2.

The experiment uses BERT model to train word vectors and the BERT pretraining model provided by Google is divided into two kinds: BERT-Base and BERT-Large. The two models have the same network structure which only some parameters are different. We used BERT-Base in this experiment. BERT-Base consists of 12 layers, the hidden layer is 768 dimensions, using 12 head modes and a total of 110M parameters. The dimension of Bi-GRU hidden layer is 300, model learning rate is 0.01, Batch Size is 64 and L2 Regular coefficient is 0.001. The function of Margin boundary parameter is to make the trained classifier have a better grasp of classification. By referring to the research of Mats *et al.* [32], we set the Margin parameter as $\frac{1}{5}\ln 2$. We used AdaGrad [33] optimization algorithm to train model. The specific parameters are set as follows (Table 3):

4.3. Experimental Results

In the experiment, we use standard CNN model, RNN model, LSTM model, Bi-LSTM model and Bi-GRU model as baseline model and compare them in the same datasets. We extract 5000 words from each corpus, 80% as training dataset, 10% as test dataset and 10% as validation dataset for cross-validation. In the initial experiment, there was a phenomenon of overfitting. In order to deal with the phenomenon of overfitting, we added the dropout [34] parameter to the model. We introduced the dropout strategy in the training process of the fully connected layer for the model. The dropout mechanism is to randomly discard some training parameters in each iteration. The workflow of dropout is to randomly select and temporarily hide some units in the nerve layer of one cycle, then train and optimize the neural network. In the next cycle, some other neurons will be hidden until the end of training. During the experiment, we compared the Accuracy of dropout (keep_prob parameter) between 0.1 and 1, and finally chose keep_prob of 0.5 as the optimal value. The experimental results are shown in Figure 5.

We found that the number of iterations is also very important to the training results of the model. The model has better fitting ability with increasing the number of iterations. According to the experimental results shown in Figure 5, we set dropout to 0.5 and on this basis, we compare the number of iterations with the change of accuracy, and finally determine that the best accuracy is when the number of iterations is 700. The experimental results are shown in Figure 6.

We use standard CNN, RNN, LSTM, Bi-LSTM and Bi-GRU as benchmark models, and compare them with the Bi-GRU model (RB-Bi-GRU) which combines grammar rules and BERT pretraining. Our experimental datasets use the same datasets. The experimental results are as follows:

As shown in Table 4, Bi-LSTM model based on grammar rules and BERT is superior to the traditional CNN, RNN, LSTM and Bi-LSTM models in accuracy, which reflecting the importance of grammatical rules for Chinese sentiment classification. To further verify the performance of the proposed model, we compared it with the frontier baseline models. Four frontier Chinese sentiment classification models with good results were selected as the baseline model. The baseline model is as follows:

Table 1 Sentiment dictionary.

Dictionary Name	Dictionary Value
Positive sentiment words	11229
Negative sentiment words	10783
Negative words	59
Degree words	219

Table 2 Experimental configuration and environment.

Experimental Environment	Environmental Configuration
Operating system	Ubuntu 16.04
CPU	Intel E5-2640v4 2.40 GHz
Memory	4*16G
Programing language	Python 3.6
Tensorflow	1.13.1
Word embedding tool	BERT

Table 3 Model parameter setting.

Parameter Name	Parameter Value
BERT layer	12
BERT hidden layer	768
BERT head mode	12
BERT parameter	110
Bi-GRU hidden layer	300
learning rate	0.01
Batch_Size	64
L2 Regular coefficient	0.001

BERT, Bidirectional Encoder Representation from Transformers; GRU, Bidirectional Encoder Representation from Transformers.

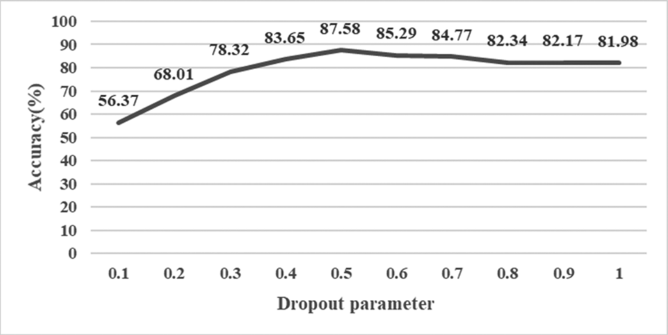

Figure 5 Effect of dropout on model performance.

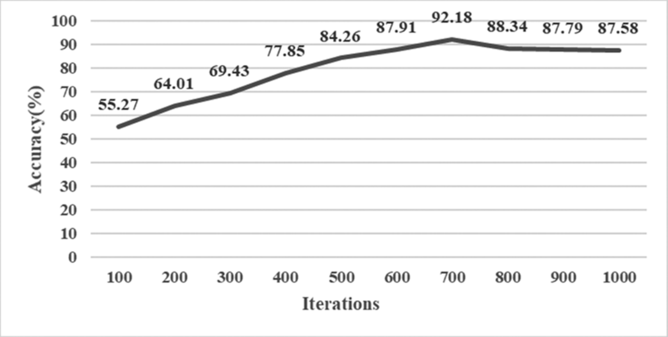

Figure 6 Effect of iterations on model performance.

- Bi-LSTM model [35]: This method integrates the contribution of sentiment information into the traditional TF-IDF algorithm and generates weighted word vectors. Weighted word vectors are input into Bi-LSTM to capture contextual information effectively and better represent comment vectors.
- Stacked Bi-LSTM model [36]: This method combines Word2vec model with Stacked Bi-LSTM model. Firstly, the word 2vec model is used to capture the semantic features of words and convert them into high-dimensional word vectors; then the Stacked Bi-LSTM model is used to extract the features of continuous word vectors; finally, the binary softmax classifier is used to predict the sentiment direction through the semantic and context features.
- CNN-Bi-LSTM model [37]: This method uses CNN based on Bi-LSTM, which can extract sequence features from global information and consider the relationship between these features. Bi-LSTM not only solves the problem of long-term dependence, but also considers the context of text.
- BI-LSTM-CRF model [38]: In this method, aggregation A21 is expressed as a sequence marker by fine-grained aspect-level affective analysis problem. Aspect words and sentiment words are extracted by aspect-level sentiment word pairing and aspect-level sentiment classification. Firstly, bidirectional LSTM-CRF model is used to extract aspect terms and sentiment words. Then, some grammar rules based on dependency analysis are used to match aspect and sentiment words. Finally, the dominant classifier is used to determine the sentiment polarity.

We use the same datasets to conduct comparative experiments. The experimental results are shown in Table 5.

As shown in Table 5, the Bi-LSTM model based on grammar rules and BERT is superior to the frontier Chinese sentiment classification model in accuracy. The main reason is that due to the ambiguity of Chinese vocabulary and the complexity of grammar rules, existing neural network models do not take into account these factors. The Bi-GRU model based on Chinese grammar rules and BERT takes grammar rules into account and simulates the function of grammar rules by judging the sentiment distribution of adjacent positions.

Table 4 Comparison with standard classification model.

Model	Accuracy (%)
RNN	72.63
CNN	81.65
LSTM	82.17
Bi-LSTM	84.51
Bi-GRU	85.46
RB-Bi-GRU	92.18

RNN, Recurrent Neural Network; CNN, Recurrent Neural Network; LSTM, long-term and short-term memory; BERT, Bidirectional Encoder Representation from Transformers; GRU, Bidirectional Encoder Representation from Transformers.

Table 5 Comparison with frontier classification model.

Model	Accuracy (%)
Bi-LSTM model	88.32
Stacked Bi-LSTM model	89.54
CNN-Bi-LSTM model	90.17
BI-LSTM-CRF model	87.51
RB-Bi-GRU	92.18

LSTM, long-term and short-term memory; GRU, Bidirectional Encoder Representation from Transformers.

In order to further verify the effect of grammar rules on accuracy, we add five comparative experimental models based on Bi-GRU, which only contains nonsentiment word regulator, sentiment word regulator, negative word regulator, degree word regulator, or BERT model. And compared with the Bi-GRU model based on grammar rules and BERT (RB-Bi-GRU). The experimental results are as follows:

As shown in Table 6, among the four grammar rule regulators, the sentiment word regulator has the greatest impact on the accuracy of model classification. Because the existence of negative words and degree words is far less than that of nonsentiment words and sentiment words in datasets, the effect of the regulator of negative words and degree words is relatively weak. Similarly, BERT pretraining model also has a great improvement in classification accuracy, so experiments show that grammar rules and BERT pretraining model have a good improvement in Chinese sentiment classification of accuracy.

5. CONCLUSIONS

Aiming at the ambiguity of Chinese vocabulary and the complexity of grammar rules, we propose a Bi-GRU sentiment classification method for Chinese based on grammar rules and BERT. It first employs BERT to process word vectors, then integrates Chinese grammar rules such as nonsentiment words, sentiment words, negative words and degree words into Bi-GRU model in the form of regularization to solve the sentiment transfer of words in Chinese sentence level, which further improves the accuracy of sentiment classification.

Since we only consider finding the optimal solution of parameters through a single experiment, and does not consider finding the optimal solution among the correlated parameters. Finding the optimal solution among the correlated parameters is the part that we need to improve and perfect in the next step. In future research, we can try to experiment in more fields to improve the generalization ability of the model, and integrate Chinese grammar rules into other sentiment classification models to further improve the accuracy of model classification.

Table 6 The comparison of experiments involving only grammar rules and BERT pre-training model.

Model	Accuracy (%)
R(ur)-Bi-GRU	86.12
R(sr)-Bi-GRU	90.55
R(nr)-Bi-GRU	87.06
R(dr)-Bi-GRU	87.49
BERT-Bi-GRU	89.64
RB-Bi-GRU	92.18

BERT, Bidirectional Encoder Representation from Transformers; GRU, Bidirectional Encoder Representation from Transformers.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHORS' CONTRIBUTIONS

Writing—original draft, Q.L.; Writing—review & editing, Z.Z.; Data curation, F.X.; Methodology, Q.L.; Software, Q.G.; Funding acquisition, Z.Z.; Project administration, Q.L.; Investigation, D.Z.; Validation, W.W.

ACKNOWLEDGMENTS

This work was supported National Social Science Foundation (19BYY076); Science Foundation of Ministry of Education of China (14YJC860042) and Shandong Provincial Social Science Planning Project (19BJCJ51, 18CXWJ01, 18BJYJ04, 17CHLJ07).

REFERENCES

[1] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv: 1301.3781, 2013. https://arxiv.org/abs/1301.3781.

[2] J. Pennington, R. Socher, C. Manning, Glove: global vectors for word representation, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 1532–1543.

[3] Q. Luo, W. Xu, J. Guo, A study on the CBOW model's overfitting and stability, in Proceedings of Proceedings of the 5th International Workshop on Web-scale Knowledge Representation Retrieval & Reasoning, Shanghai, China, 2014, pp. 9–12.

[4] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, New Orleans, LA, USA, 2018, pp. 2227–2237.

[5] W. Che, Y. Liu, Y. Wang, B. Zheng, T.J. Liu, Towards better UD parsing: deep contextualized word embeddings, ensemble, and treebank concatenation, in Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Brussels, Belgium, 2018.

[6] S. Ilić, E. Marrese-Taylor, J.A. Balazs, Y.J. Matsuo, Deep contextualized word representations for detecting sarcasm and irony, in Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Brussels, Belgium, 2018.

[7] M. Hu, B. Liu, Mining and summarizing customer reviews, in Proceedings of Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, Seattle, WA, USA, 2004, pp. 168–177.

[8] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K.J. Miller, Introduction to WordNet: an on-line lexical database, Int. J. Lexicograph. 3 (1990), 235–244.

[9] S. Zhang, Z. Wei, Y. Wang, T. Liao, Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary, Future Gener. Comput. Syst. 81 (2018), 395–403.

[10] J. Graovac, M. Mladenović, I. Tanasijević, NgramSPD: exploring optimal n-gram model for sentiment polarity detection in different languages, Intell. Data Anal. 23 (2019), 279–296.

[11] N. Zalmout, N. Habash, Adversarial multitask learning for joint multi-feature and multi-dialect morphological modeling, in Proceedings of Proceedings of the 57th Conference of the Association for Computational Linguistics, Florence, Italy, 2019, pp. 1775–1786.

[12] T. Zhang, S.S. Ge, An improved TF-IDF algorithm based on class discriminative strength for text categorization on desensitized data, in Proceedings of Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence, Suzhou, China, 2019, pp. 39–44.

[13] K. Li, Y. Gu, P. Zhang, W. An, W. Li, Research on KNN algorithm in malicious PDF files classification under adversarial environment, in Proceedings of Proceedings of the 2019 4th International Conference on Big Data and Computing, Guangzhou, China, 2019, pp. 156–159.

[14] L. Yang, Y. Xiang, Naive bayes and BiLSTM ensemble for discriminating between mainland and Taiwan variation of mandarin Chinese, in Proceedings of Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects, Ann Arbor, MI, USA, 2019, pp. 120–127.

[15] K. Lu, J. Wu, Sentiment analysis of film review texts based on sentiment dictionary and SVM, in Proceedings of Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence, Suzhou, China, 2019, pp. 73–77.

[16] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, in Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA, 2002, vol. 10, pp. 79–86.

[17] X. Xie, S. Ge, F. Hu, M. Xie, N. Jiang, An improved algorithm for sentiment analysis based on maximum entropy, Soft Comput. 23 (2019), 599–611.

[18] Z. Qingqing, L. Xilin, Classifying sentiments based on BPSO random subspace, Data Anal. Knowl. Discov. 1 (2017), 71–81.

[19] Y. Kim, Convolutional neural networks for sentence classification, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014.

[20] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in Proceedings of Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 2010, pp. 249–256. http://proceedings.mlr.press/v9/glorot10a.html.

[21] L. Hui, C. Yaqing, Fine-grained sentiment analysis based on convolutional neural network, Data Anal. Knowl. Discov. 3 (2019), 95–103.

[22] A. Ezen-Can, E.F. Can, Hybrid RNN at SemEval-2019 task 9: blending information sources for domain-independent suggestion mining, in Proceedings of Proceedings of the 13th International Workshop on Semantic Evaluation, Minneapolis, MN, USA, 2019, pp. 1199–1203.

[23] A. Abdi, S.M. Shamsuddin, S. Hasan, J. Piran, Deep learning-based sentiment classification of evaluative text based on Multi-feature fusion, Inf. Process. Manage. 56 (2019), 1245–1259.

[24] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, arXiv: 1810.04805, 2018.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, arXiv: 1706.03762, 2017. https://arxiv.org/abs/1706.03762.

[26] S. Vosoughi, P. Vijayaraghavan, D. Roy, Tweet2vec: learning tweet embeddings using character-level CNN-LSTM encoder-decoder, in Proceedings of Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy, 2016, pp. 1041–1044.

[27] Y. Wang, F. Tian, Recurrent residual learning for sequence classification, in Proceedings of Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, 2016, pp. 938–943.

[28] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, arXiv: 1607.06450, 2016. https://arxiv.org/abs/1607.06450.

[29] R. Dey, F.M. Salemt, Gate-variants of Gated Recurrent Unit (GRU) neural networks, in Proceedings of 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 2017, pp. 1597–1600.

[30] M. Sundermeyer, R. Schlüter, H. Ney, LSTM neural networks for language modeling, in Proceedings of Thirteenth Annual Conference of the International Speech Communication Association, Portland, OR, USA, 2012, pp. 194–197. https://www.isca-speech.org/archive/interspeech_2012/i12_0194.html.

[31] J. Lin, Divergence measures based on the Shannon entropy, IEEE Trans. Inf. Theor. 37 (1991), 145–151.

[32] M. Byrkjeland, F.G. de Lichtenberg, B. Gambäck, Ternary Twitter sentiment classification with distant supervision and sentiment-specific word embeddings, in Proceedings of Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Brussels, Belgium, 2018, pp. 97–106.

[33] M.C. Mukkamala, M. Hein, Variants of rmsprop and adagrad with logarithmic regret bounds, in Proceedings of Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 2017, vol. 70, pp. 2545–2553. https://arxiv.org/abs/1706.05507.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014), 1929–1958. https://dl.acm.org/doi/10.5555/2627435.2670313.

[35] G. Xu, Y. Meng, X. Qiu, Z. Yu, X. Wu, Sentiment analysis of comment texts based on BiLSTM, IEEE Access. 7 (2019), 51522–51532.

[36] J. Zhou, Y. Lu, H.-N. Dai, H. Wang, H. Xiao, Sentiment analysis of Chinese microblog based on stacked bidirectional LSTM, IEEE Access. 7 (2019), 38856–38866.

[37] K. Zhou, F. Long, Sentiment analysis of text based on CNN and bi-directional LSTM model, in Proceedings of 2018 24th International Conference on Automation and Computing (ICAC), Newcastle upon Tyne, UK, 2018, pp. 1–5.

[38] H. Xiong, H. Yan, Z. Zeng, B. Wang, Dependency parsing and bidirectional LSTM-CRF for aspect-level sentiment analysis of Chinese, in Proceedings of JIST (Workshops & Posters), Awaji, Japan, 2018, pp. 90–93. https://ceur-ws.org/Vol-2293/jist2018pd_paper4.pdf.