Wuhan University
Journal of Natural Sciences

2022, Vol.27 No.6, 465-475

Article ID 1007-1202(2022)06-0465-11
DOI https://doi.org/10.1051/wujns/2022276465

A Short Text Classification Model for Electrical Equipment Defects Based on Contextual Features

□ LI Peipei[1], ZENG Guohui[1†], HUANG Bo[1], YIN Ling[1], SHI Zhicai[1], HE Chuanpeng[1], LIU Wei[2], CHEN Yu[3]

1. School of Electrical and Electronic Engineering, Shanghai University of Engineering Science, Shanghai 201620, China;

2. CSG Smart Science & Technology Co., LTD, Shanghai 201203, China;

3. State Grid Shanghai Municipal Electric Power Company, Shanghai 200122, China

Abstract: The defective information of substation equipment is usually recorded in the form of text. Due to the irregular spoken expressions of equipment inspectors, the defect information lacks sufficient contextual information and becomes more ambiguous. To solve the problem of sparse data deficient of semantic features in classification process, a short text classification model for defects in electrical equipment that fuses contextual features is proposed. The model uses bi-directional long-short term memory in short text classification to obtain the contextual semantics of short text data. Also, the attention mechanism is introduced to assign weights to different information in the context. Meanwhile, this model optimizes the convolutional neural network parameters with the help of the genetic algorithm for extracting salient features. According to the experimental results, the model can effectively realize the classification of power equipment defect text. In addition, the model was tested on an automotive parts repair dataset provided by the project partners, thus enabling the effective application of the method in specific industrial scenarios.

Key words: short text classification; genetic algorithm; convolutional neural network; attention mechanism

CLC number: TN 918

Received date: 2022-09-11

Foundation item: Supported by the Scientific and Technological Innovation 2030—Major Project of "New Generation Artificial Intelligence" (2020AAA 0109300)

Biography: LI Peipei, female, Master candidate, research direction: natural language processing. E-mail: 2510243510@qq.com

† To whom correspondence should be addressed. E-mail: zenggh@sues.edu.cn

0 Introduction

Power equipment inspection is essential to maintain the system's regular operation. The equipment defects found in the inspection will be presented in the power defect system. Determining the type of equipment defects is a prerequisite for eliminating them. However, the present defect classification work is mainly completed by manual classification. As the scale of the power system continues to expand, the number of devices is increasing exponentially, which significantly increases the inspection workload[1-4]. Therefore, better utilization of short text classification to improve the efficiency of defect identification is an urgent problem in the power industry[5].

During the operation of electrical equipment, many defect data are generated[6,7], which are usually recorded manually by inspectors and classified by professionals according to their experience. In addition, these data are characterized by the lack of semantic information, sparse data, and high dependency. Therefore, improving the short text classification model is the key to classifying defects in power equipment[8].

With the development of deep learning, it has made exemplary achievements in the fields of computer vision[9], speech recognition[10,11], and text classification [12]. Convolutional neural network[13] (CNN) and recurrent neural network[14] (RNN) are the commonly used deep learning networks. However, CNN ignores the dependency features among local information and RNN is prone to the problems of gradient disappearance and gradient explosion. Liu *et al*[15] used CNN to classify short texts of electrical equipment defects, which reduced the

classification error rate comparing with the traditional machine learning classification methods. On this basis, scholars proposed the long short-term memory network[16] (LSTM), which effectively solved the problem of RNN. Further, bi-directional long-short term memory[17] (BiLSTM) was developed to obtain contextual features from text sequences forward and backward. Wei *et al*[18] proposed a fault detection classification method combining BiLSTM and CNN, which extracts local feature information through the maximum pool layer in CNN, but cannot extract global features. Therefore, the network needs to be further optimized to retain the features of global information.

Currently, graph neural networks (GNN) designed for short texts have achieved good results. Hao *et al*[19] first transformed the text into a text-graph structure, obtained word embeddings by graph convolution operations, and fed them to a classifier for text classification. Yao *et al*[20] transformed the text classification problem into a node classification problem and applied GNN to corpus graphs, which eventually achieved excellent text classification results. Hu *et al*[21] modeled corpus-level latent topic, entity, and document graphs, while Ye *et al*[22] operated on corpus-level latent topic, document, and word graphs. Both papers connect documents to different types of entities, such as latent topics and entities, but do not connect to other documents and cannot capture similarities between short texts.

For short text classification of electrical equipment defects, a large amount of irrelevant topic information involved in BiLSTM training will lead to degradation of classification performance. The attention mechanism assigns weights according to word importance and highlights contextually essential information. The combination of BiLSTM and the attention mechanism can further improve classification accuracy. But the ability of BiLSTM to capture contextual features is weak. Therefore, we introduce CNN to capture salient topic features and make full use of contextual features to improve the classification accuracy. Due to the random initialization of weight values, the gradient descent method used by CNN may fall into a local optimum solution, for which an optimization algorithm can be used to find the appropriate parameter values.

Above all, the main contributions of this paper are as follows:

1) In this paper, we propose a text classification model that combines BiLSTM with the Attention Mechanism and CNN optimized by the Genetic Algorithm. The feature vectors of the model inputs are constructed and selected by the bidirectional encoder representation from transformers (BERT) model, thus improving the accuracy of the short text classification model.

2) In order to obtain crucial semantic information in the sequence, this paper integrates BiLSTM with the Attention Mechanism to capture the important semantic information in the sentence by assigning different weights to the information extracted from the forward hidden layer and the backward hidden layer.

3) We introduce CNN to capture important local word order features from textual contextual features, and optimize CNN weight vectors with the help of the genetic algorithm to find the model with the best weight values.

1 Model Architecture

The model architecture proposed in this paper is shown in Fig. 1, which includes five parts: encoding

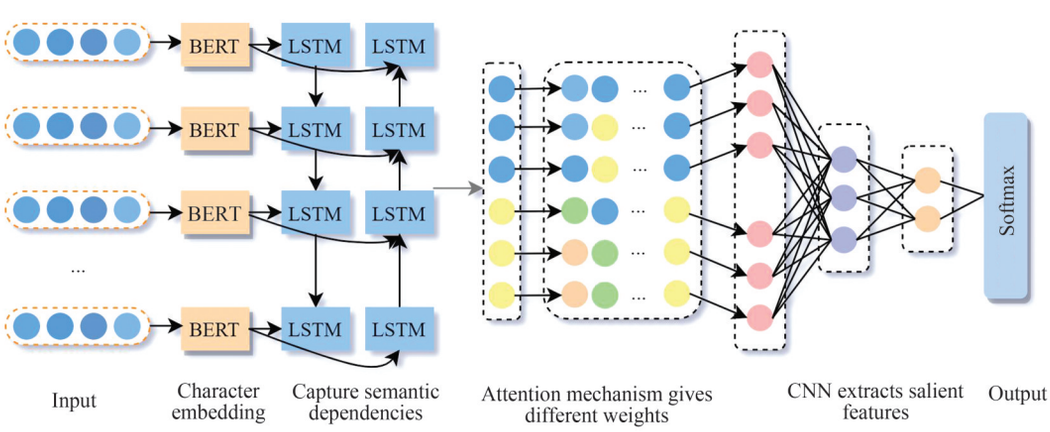

Fig. 1 BAGC model architecture

words with BERT, capturing semantic information with BiLSTM, giving different weights with the attention mechanism, capturing salient features with CNN and Softmax classification.

First, the character embedding part uses the BERT model as the initialization method for text representation, converting words into word feature vectors. Second, the input to the BiLSTM layer is the feature vector obtained from the word embedding layer. The contextual features are obtained from the forward and backward hidden layers while capturing the bidirectional semantic dependencies. Then, the attention mechanism assigns higher weights to the words that affect the semantic information. Meanwhile, CNN extracts locally significant features while ultimately maintaining long-term dependencies. Finally, the classification layer fuses the pooled features together to form a feature map, which is used for Softmax classification. Algorithm 1 shows the algorithmic representation of the BAGC model architecture.

Algorithm 1 BAGC model algorithm flow

Input: text $S=\{x_1,x_2,\cdots,x_T\}, x_i(i\in[1,T])$

Output: class label $\hat{C}$

1.Input to the word embedding layer and convert to word vector form:

$$\boldsymbol{S}=\{\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_T\}$$

2.Inputs to the BiLSTM layer are the word vectors:

$$\boldsymbol{h}_t=\mathrm{LSTM}(\boldsymbol{e}_t), t\in[1,T]$$

3.Input the $\boldsymbol{h}_t$ produced by the BiLSTM layer to the Attention layer, and obtain the implicit information through nonlinear transformation:

$$\boldsymbol{u}_t=\tanh(\boldsymbol{W}_h\boldsymbol{h}_t+\boldsymbol{b}_h)$$

4.Randomly initialize the attention matrix $\boldsymbol{v}$ and multiply $\boldsymbol{u}_t$ for normalization to form the attention matrix:

$$\boldsymbol{\alpha}_t=\frac{\boldsymbol{u}_t{}^{\mathrm{T}}\boldsymbol{v}}{\sum_t\exp(\boldsymbol{u}_t{}^{\mathrm{T}}\boldsymbol{v})}$$

5. Form the output:

$$\boldsymbol{s}_i=\sum_t\boldsymbol{\alpha}_t\boldsymbol{h}_t$$

6.Taking the text embedding vectors$\{\boldsymbol{s}_1,\boldsymbol{s}_2,\cdots,\boldsymbol{s}_n\}$ as the input of CNN, a window of h words are convolved through a filter to generate a new characteristic:

$$\boldsymbol{y}_i=f(\boldsymbol{W}_k*\boldsymbol{s}_{i:i+h-1}+\boldsymbol{b})$$

7. Using $\boldsymbol{y}_i$ as the input of the Softmax layer, the probability of the corresponding category of the text is generated:

$$P(S|C)=\mathrm{Softmax}(\boldsymbol{y}_i)$$

8. Classify text: $\hat{C}=\mathrm{argmax}P(S|C)$

9. Return $\hat{C}$

1.1 BERT Word Vector Encoding

In this paper, WordPiece is used to segment the input sentence. Each word is processed into a vector form of words, texts, and positions and simultaneously added to the BERT coding layer. As shown in Fig. 2, the sentence P, "Insulation aging of stator winding of Longhu line" is divided into several words, which become "Insulation," "aging," "of," "stator," "winding," " of," "Longhu," " line." They correspond to Token layer information, Segment layer information, and Position layer information, respectively. The token layer is the vector embedding of words, the Segment layer is used to distinguish which sentence belongs to which, and the Position layer is used to distinguish the position in the sequence.

Also, we set the sentence length of the BERT encoding layer input at 32 bits, where [CLS] and [SEP] take 2 characters each. When the length exceeds 30 bits, the subsequent sentences are cut off; when the length is less than 30 bits, supplementary <padding> is used.

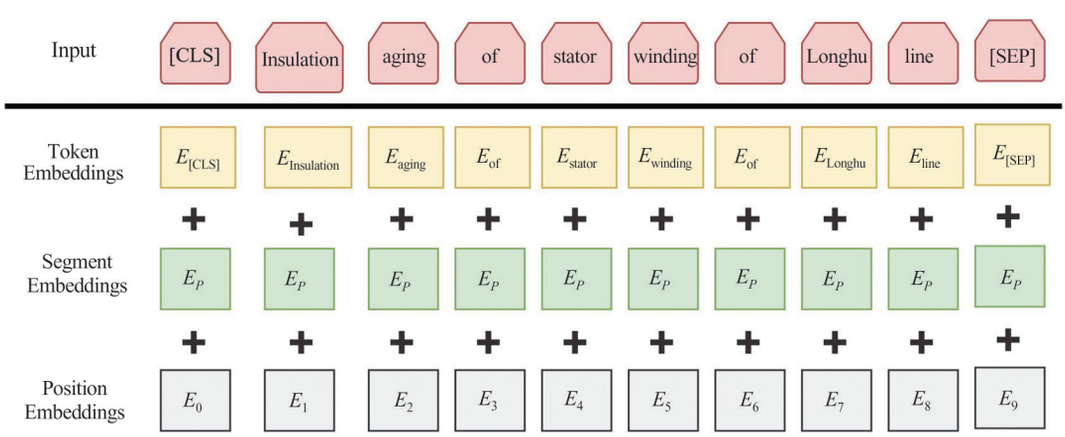

Fig. 2 Representation of BERT input layer

Among them, [CLS] represents the special symbol of classification output, and [SEP] represents the end of a sentence, occupying one character, respectively. The calculation formula for the input layer is as follows:

$$\boldsymbol{h}_1 = \boldsymbol{E}_{\text{Token}} + \boldsymbol{E}_{\text{Segement}} + \boldsymbol{E}_{\text{Position}} \tag{1}$$

For a text S consisting of T words, text is converted into word vector form by using BERT pre-training language model:

$$\boldsymbol{S} = \left\{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_T\right\} \tag{2}$$

1.2 BiLSTM Captures Bidirectional Semantic Dependencies

LSTM solves the problem of gradient disappearance and explosion to a certain degree. However, LSTM cannot extract contextual information, and BiLSTM can better capture bidirectional semantic dependencies. Therefore, this paper captures the context features of the text through BiLSTM, which is composed of forwarding and backwarding LSTM units and gains information from two reverse orientations separately. The information output of sentence P through the BERT encoding layer is $\boldsymbol{P} = \left\{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_T\right\}$, which is input to the BiLSTM layer, where the dimension corresponding to the matrix $\boldsymbol{P}$ is the number of batch training samples × the maximum sentence length × the hidden layers of BiLSTM. BiLSTM encodes the sentence P, and each word vector output contains contextual features. The formula is as follows:

$$\overrightarrow{\boldsymbol{h}_i^p} = \overrightarrow{\mathbf{LSTM}}\left(\overrightarrow{\boldsymbol{h}_{i-1}^p}, \boldsymbol{p}_i\right) \tag{3}$$

$$\overleftarrow{\boldsymbol{h}_i^p} = \overleftarrow{\mathbf{LSTM}}\left(\overleftarrow{\boldsymbol{h}_{i+1}^p}, \boldsymbol{p}_i\right) \tag{4}$$

$$\boldsymbol{h}_i^p = \overrightarrow{\boldsymbol{h}_i^p} \oplus \overleftarrow{\boldsymbol{h}_i^p} \tag{5}$$

where $\overrightarrow{\boldsymbol{h}_i^p}$ and $\overleftarrow{\boldsymbol{h}_i^p}$ are the output results of BiLSTM forward LSTM and reverse LSTM at time i, respectively. $\boldsymbol{h}_i^p$ is the output layer information of BiLSTM at time i.

1.3 The Attention Mechanism Assigns Different Weights

The contribution of each word in the text sequence to the classification is different. For example, in sentence P, "Insulation aging of stator winding of Longhu line", after the BiLSTM layer operation, the result obtained is that each word plays an equally important role, but "Insulation aging" obviously plays a more critical role actually. Accordingly, this paper introduced the Attention Mechanism, which calculates the correlation coefficient between words in the text sequence, and the weights is assigned to the word vector according to the correlation between words.

The input of the attention layer is $\boldsymbol{h}_t$ generated by the BiLSTM layer, and the hidden information is obtained by nonlinear transformation. After the Attention Mechanism runs, the output of sentence P is a word vector. The word vector of "Insulation aging" has a greater impact on text classification than other word vectors, so "Insulation aging" should be given a higher weight in the output word vector. In this paper, we convert $\boldsymbol{h}_t$ to $\boldsymbol{u}_t$ by fully connected layer operation, and the formula is as follows:

$$\boldsymbol{u}_t = \tanh\left(\boldsymbol{W}_h \boldsymbol{h}_t + \boldsymbol{b}_h\right) \tag{6}$$

The word-to-word correlation coefficients are calculated with the help of scoring functions. It measures the correlation between words and words, which is converted into a probability distribution by Softmax.

$$\boldsymbol{a}_t = \frac{\boldsymbol{u}_t{}^{\mathrm{T}} \boldsymbol{v}}{\sum_t \exp\left(\boldsymbol{u}_t{}^{\mathrm{T}} \boldsymbol{v}\right)} \tag{7}$$

Finally, the feature vector is formed after the weighting operation:

$$\boldsymbol{s}_i = \sum_t \alpha_t \boldsymbol{h}_t \tag{8}$$

where $\boldsymbol{h}_t$ is the characteristic vector output when the model runs the BiLSTM at time t, $\boldsymbol{b}_h$ is the corresponding offset at time t, $\boldsymbol{W}_h$ is the weight coefficient matrix at time t, $\boldsymbol{v}$ is the attention matrix initialized as circumstances warrant, $\boldsymbol{\alpha}_t$ is the weight of each word in the sentence, $\boldsymbol{s}_i$ is the weighted output vector.

1.4 CNN Extracts Salient Features

Based on the context features of the text, this paper uses CNN to gain further the topic's salient characteristics of text alignments from contextual attributes. A convolution operation obtains the featured graph. The convolution result is sampled by pooling layer to decrease the magnitude of the convolution vector and avoid overfitting.

In this paper, the convolution operation is performed by using a plurality of convolution kernels, and the $\left\{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_n\right\}$ obtained by the Attention layer is expressed as:

$$\boldsymbol{s}_{1:n} = \boldsymbol{s}_1 \oplus \boldsymbol{s}_2 \oplus \cdots \oplus \boldsymbol{s}_n \tag{9}$$

Passing a convolutional filter with an h-width window, the new features are obtained:

$$\boldsymbol{y}_i = f\left(\boldsymbol{W}_k * \boldsymbol{s}_{i:i+h-1} + \boldsymbol{b}\right) \tag{10}$$

where $\oplus$ is the concatenation operator, $\boldsymbol{b}$ stands for the bias, $\boldsymbol{W}_k$ denotes the corresponding weight matrix corresponding of diverse convolution kernels, i acts for the i-

th eigenvalue, h indicates the scale of the convolution kernel, and f indicates the Relu nonlinear activation function, $\boldsymbol{y}_i$ represents the consequence of convolution calculation.

Then, the extracted critical information is pooled. The largest eigenvalue is extracted in the sampling window, and all sampled eigenvalues are combined into $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, \cdots, \boldsymbol{y}_n\}$, the export of CNN, such as formula (11):

$$\boldsymbol{y} = \sum_{i=1}^{n-h+1} \max(\boldsymbol{y}_i) \tag{11}$$

1.5 Genetic Algorithm to Optimize CNN Parameters

The core of the genetic algorithm [23-25] (GA) is parameter encoding, population initialization, and determination of fitness function, and then the optimal solution is obtained by the search. The classical CNN learning method uses the fastest descent algorithm for learning, and the learning performance is strongly affected by the initial weight settings of the convolutional and fully connected layers. The optimal weights are obtained after selection, crossover, and variation operations as the initial weights of CNN. These weights are used as the initial weights of the CNN, and their learning performance is better than the initial weights randomly selected by the fastest descent algorithm.

The main problem of training CNN with the fastest descent algorithm is falling into the local optimal solution. To solve this problem, we introduced the GA. The fundamental idea is using the GA to determine the initial weights of the CNN classifier. Firstly, multiple sets of initial weights are selected, and the combination methods of each set of weights are encoded as chromosomes. Different weight combination methods are generated by chromosomes' selection, crossover, and mutation operations. Then, the fitness value of chromosomes is used to select the optimal weight combination, where the fitness value is the CNN classification accuracy using different weight combination techniques.

The optimization improvement model process is shown in Fig. 3.

The steps of chromosome encoding and fitness solution are as follows:

1) decoding the chromosomes to get the CNN's convolutional and fully connected layers' initial weights;

2) using the fastest descent algorithm to train the CNN network d times;

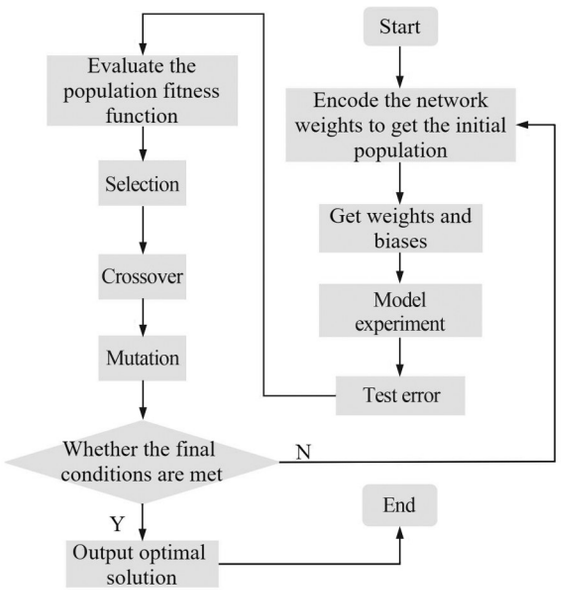

Fig. 3 Genetic algorithm to optimize CNN network

3) calculating the CNN's accuracy as the fitness value of the corresponding chromosome. The optimization algorithm is shown in Algorithm 2,where the parameter d is the number of iterations in training, M is the population size, PC is the crossover probability, PM is the mutation probability, weight is the CNN weight, learning_rate is the learning rate, fitness(x_i) represents fitness function, sorted( ) is the sorting function.

Algorithm 2 Genetic Algorithm to optimize CNN network

Input: Genetic Algorithm Population, M=20, PC=0.65, PM=0.05

Output: Final population G

1. Chromosomes encoded by initialized weight combinations

2.Do {

3. Initializing the CNN classifier and train classifier with the steepest descent algorithm, updating network weights:
$\text{weight} = \text{weight} - \text{learning}_{\text{rate}} * \text{gradient}$

4. Using the accuracy of each network as the fitness of chromosomes, calculate and rank:
best_fitness = sorted(cur_evaluation, key = lambdax:x['train_acc'])

5. Performing genetic operations, the higher the fitness, the greater the probability of being selected:
$p_i = \dfrac{\text{fitness}(x_i)}{\sum_{i=1}^{n} \text{fitness}(x_i)}$

6. } while (The fitness value meets the termination condition or the reproduction algebra reaches the upper limit)

7.Return G

2 Experiments and Analysis

2.1 Dataset Introduction

In order to study the classification effect of the model constructed in this paper on the defective texts of electric equipment, 6 750 defective text data from the electric equipment records of the State Grid Corporation of China are randomly selected as the research object, and there are six fault categories: blockage, leakage, misalignment, failure, invalidation and others. Among them, 5 400 texts are used in the training set, and the texts used in the test set and validation set are 675, respectively. A sample of dataset is shown in Table 1.

Table 1 Dataset data sample

Defect description	Category
There is a foreign body inside the bearing	blockage
The main transmission oil pressure relay penetratess oil, no more than 6 drops/min	leakage
Wenxiang Transformer No. 1 Main Variant Transformer Winding	misalignment
Zhushan transformer bus tie switch mechanism heater failure	failure
Insulation aging of stator winding of Longhu line	invalidation
The axis of the hole system and the cross-sections at both ends are not parallel	others

Short texts on electrical defects are different from other Chinese texts, which have the following four characteristics on the whole:

1) Most defects have an inseparable relationship with the exclusive domain of power equipment, and there are many electrical specialized words in the text. In addition, due to the unique expression habits of the inspectors, there may be different descriptions for the same component, such as "shake meter" and "megger".

2) Due to the complexity of defects and the different recording habits of the inspectors, the length of each defective short text varies from each other, the shortest can be as few as four words, and the longest can be as long as 40 words.

3) The exact fault location can lead to different classifications due to different types of faults. For example, faults caused by display panels are classified into two types: display panel black screen and display panel unclear.

4) Faulty texts include a vast quantity of data, and the similarity of different forms of defective data may be high and lack sufficient semantic information. However, traditional text classification models have unavoidable limitations in classifying texts with high similarity. Meanwhile, classifying defective texts requires high storage space and computational power for classification models.

2.2 Hyperparameters Settings

Based on the experimental process, some hyperparameters of this model are set as shown in Table 2.

Table 2 Experimental hyperparameter settings

Hyperparameter name	Value setting
dimensions of embeddings	200
max length of sentences	32
number of epoches	30
size of BiLSTM	256
size of batch	64
dropout	0.1
learning rate	5E−5
size of convolution kernel	[2,3,4]
optimizer	Adam

1) Word vector dimension: The setting of the word vector dimension affects the word representation's accuracy. As shown in Fig. 4, with the increase of word vector dimension, the classification accuracy shows increasing and decreasing trends. The accuracy reaches the highest value when the word vector dimension is 200, indicating that the word meaning cannot be represented accurately when the word vector dimension is low. Meanwhile, higher dimensionality can also make the vector representation too sparse and cause redundancy.

2) Neural network hidden layer: The operational capability of CNN is determined on the quantity of hidden layers. The following conclusion can be drawn from Fig. 5. Along with the increasing of hidden layers, the

classification accuracy shows an increasing and decreasing trend. When the number of hidden layers is set to 256, the model reaches the best accuracy.

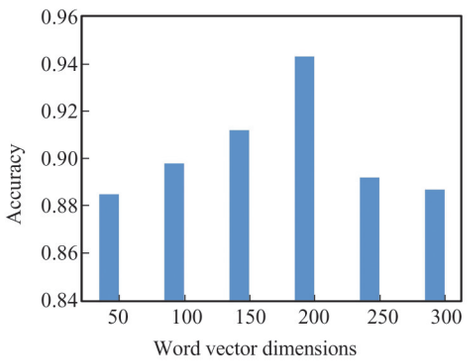

Fig. 4 Word vector dimensions setting

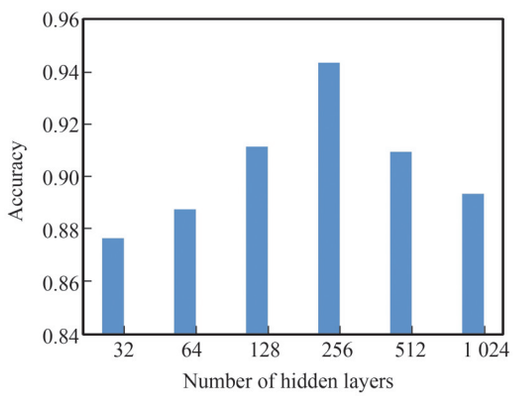

Fig. 5 Number of hidden layers setting

3) CNN convolution kernel size: The sizes of convolution kernels affect CNN's ability to obtain local features. As shown in Fig. 6, choosing convolution kernel size as [2, 3, 4], the model accuracy can be higher than other convolution kernels.

4) Setting of epoch number: The M_{F1} score and train loss F_{Loss} of different epoch is shown in Fig. 7 and Fig. 8, respectively. With the increase of iterations, the training process tends to be stable, and the model's M_{F1} score and F_{Loss} tends to converge. When the iterations reach 30, the training set M_{F1} = 93.65% and the verification set M_{F1} = 91.58%.

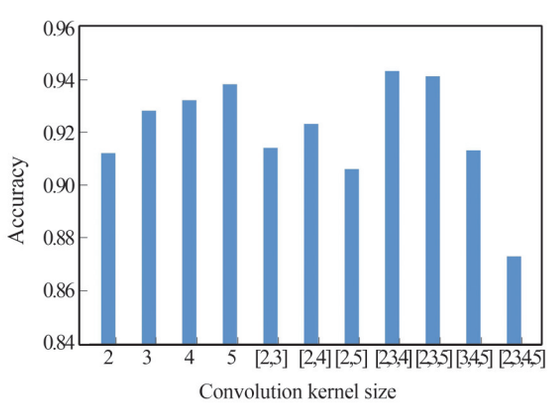

Fig. 6 Convolution kernel sizes setting

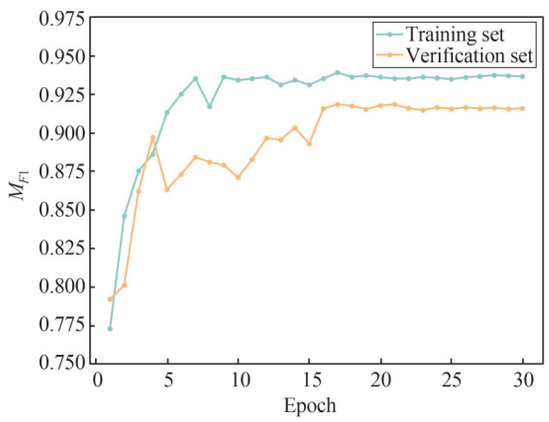

Fig. 7 M_{F1} evaluation index curve

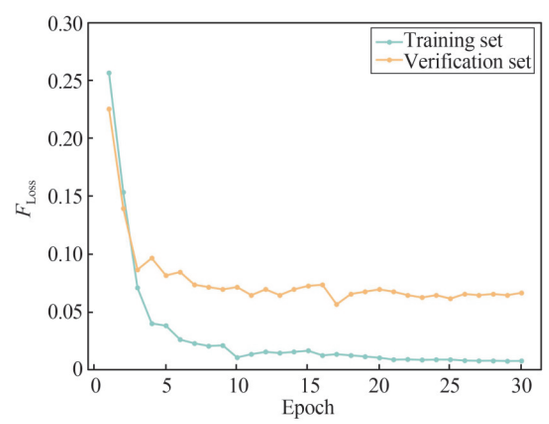

Fig. 8 F_{Loss} evaluation index curve

2.3 Description of Evaluation Indicators

In binary classification problems, accuracy, precision, recall and $F1$ value are the usually used methods to measure model performance, and the corresponding formula is as follows. Accuracy refers to the proportion of correctly classified samples to the total number of samples. Precision refers to the proportion of correctly predicted positive classes to all predicted positive classes. Recall refers to the proportion of correctly predicted positive classes to all actual positive classes. $F1$ integrally evaluates the results of accuracy and recall. TP (true positive) indicates that positive samples are correctly identified as positive samples. FP (false positive) indicates that negative samples are incorrectly identified as positive samples. TN (true negative) indicates that

negative samples are correctly identified as negative samples. FN (false negative) indicates that positive samples are incorrectly identified as negative samples.

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}} \tag{12}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{13}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{14}$$

$$F1 = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{15}$$

Since the power defect text contains six categories, it is a multi-classification problem. In order to comprehensively evaluate the classification effect of the model, we use the macro-average composite index. Among them, the macro precision rate M_{P} and the macro recall rate M_{R} are defined as follows, n representing the number of experimental sample data categories.

$$M_{F1} = \frac{2\times M_{\text{P}}\times M_{\text{R}}}{M_{\text{P}}+M_{\text{R}}}\times 100\% \tag{16}$$

$$M_{\text{P}} = \frac{1}{n}\sum_{i=1}^{n}\text{Precision}_i\times 100\% \tag{17}$$

$$M_{\text{R}} = \frac{1}{n}\sum_{i=1}^{n}\text{Recall}_i\times 100\% \tag{18}$$

2.4 Comparative Experiment and Analysis

To further validate the effectiveness of the BAGC model in text classification, we conducted comparison experiments with the baseline model on the dataset. Table 3 shows the results of comparative experiments.

1) TextCNN[26]: In this model, the CNN is applied to the text classification task, by using multiple sizes of convolution kernels, critical information in the sentence can be extracted, thus enabling better capture of local relevance. Then the model is connected with Softmax for classification.

2) TextRNN[27]: This model uses a two-layer RNN, which is good at capturing more comprehensive sequence information. Then the most critical features are automatically screened out by maximum pooling, and then a fully connected layer is used for classification.

3) FastText[28]: The input of the model is a word sequence and output is the belongingness probability of this sequence to different categories. The words and phrases in the sequence are formed into feature vectors, which are mapped to the middle layer by a linear transformation, and then mapped to labels. Meanwhile, a nonlinear activation function is used to predict the categorical labels.

4) BERT[29]: The model takes the output CLS-marked vector from the last encoder of BERT to generate the probability values belonging to each label through a fully connected layer, and then the largest one will be chosen as the prediction result.

5) BiLSTM_Attention [30]: The model is a dynamic pre-trained word vector model that captures textual, contextual feature information using a multi-layer Bidirectional Transformer architecture. The hidden layer acquires the global classification feature information, and generates the probability value of each label through the fully connected layer to get the final prediction result.

Table 3 shows that the model in this paper outperforms the baseline method. The M_{F1} of the BAGC model improves by 3.4%, 4.17%, 3.54%, 1.2%, and 2.94% when compared with the TextCNN model, TextRNN model, FastText model, BERT model, and BiLSTM_Attention model, respectively, which means a better learning performance.

Table 3 Overall comparison of experimental results %

Model	M_{P}	M_{R}	M_{F1}
TextCNN	90.92	90.77	90.84
TextRNN	90.20	89.95	90.07
FastText	91.32	90.08	90.70
BERT	93.21	92.87	93.04
BiLSTM_Attention	91.52	91.08	91.30
BAGC (ours)	94.31	94.18	94.24

Figure 9 shows the single-label classification accuracy of the BAGC model and the baseline model on data samples. It is obtained from Fig. 9 that our model shows a trend of being more accurate than these baseline mod-

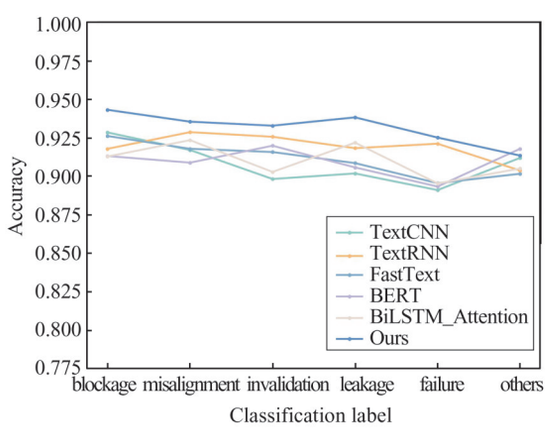

Fig. 9 Single-label classification accuracy

els, and only the single-label accuracy of our model is lower than the other models.

To verify the model's generalization ability, the classification accuracy of each defect category is tested. As shown in Table 4, the M_{F1} values for the "blockage" and "leakage" categories are 95.27% and 96.59%, respectively. The reason may be that in the text preprocessing stage, the data of these two categories are more different from other categories. At the same time, the weaker correlation with other categories allows the data in this category to be better distinguished. In addition, we have applied the model to the customer service datasets provided by the partners of the project on which this paper relies (two automobile components manufacturing companies). For the classification of defects in the example sentence "The fuel pump suddenly stops working and the throttle suddenly sticks", the model gives the correct classification result, and this category belongs to "engine part failure". The experimental results prove that the model starts from the actual needs of automotive enterprises, helps them to quickly obtain product defect information and further improves defect management, and realizes the effective validation of this method in specific industrial application scenarios.

Table 4 Classification results of BAGC model %

Categories	M_P	M_R	M_{F1}
blockage	95.38	95.16	95.27
misalignment	92.61	92.72	92.66
invalidation	93.93	93.71	93.82
leakage	96.95	96.24	96.59
failure	94.68	94.72	94.70
others	92.31	92.53	92.42

2.5 Analysis of Ablation Experiments

To further verify the validity of the components of the BAGC model, we performed ablation experiments. BAGC/B is the BAGC model removing the BiLSTM module, which extracts contextual semantics. BAGC/A is the BAGC model removing the Attention Mechanism. BAGC/C is the BAGC model removing the process of CNN module, which captures local word order features of the text. BAGC/G is the BAGC model removing the GA optimization in CNN weight vector operation. Table 5 shows the experimental results.

As can be seen from Table 5, the Attention Mechanism and the BiLSTM layer have a large impact on the classification ability of the BAGC model. Compared with BAGC/B, the M_{F1} of the BAGC model is improved by 2.63%. Once the BiLSTM layer is removed, the effectiveness of the BAGC model will be significantly reduced, proving that BiLSTM can make up for the shortcomings of a single deep learning model and better extract contextual semantic information from text data.

Table 5 Results of ablation experiments %

Model	M_P	M_R	M_{F1}
BAGC/B	91.79	91.43	91.61
BAGC/A	93.12	92.58	92.85
BAGC/C	93.26	93.12	93.19
BAGC/G	93.53	93.08	93.45
BAGC	94.31	94.18	94.24

The M_{F1} of the BAGC model is enhanced by 1.39% compared with BAGC/A. It is demonstrated that the Attention Mechanism can assign more weights to the words that affect the semantics, which improves the performance of the model. The M_{F1} of the BAGC model is improved by 1.05% compared with BAGC/C, which shows that the convolutional layer has less impact on the model than other components, but it still helps to improve the classification accuracy. Compared with BAGC/G, the M_{F1} of the BAGC model is improved by 0.79%, which indicates the validity of the Genetic Algorithm in CNN layer on improving the classification performance.

To sum up, each component of the BAGC model is necessary, and the results obtained after shows that the BAGC model removing any of them are suboptimal.

3 Conclusion

Based on the shortcomings of sparse data and insufficient semantic features in the short text classification process, a deep short text classification model incorporating contextual features is proposed. The input to the model is a text vector generated using the BERT word vector model. In order to better extract contextual semantic information from the samples, words will be given unequal weight values according to their different importance. Based on this, a CNN is optimized by the Genetic Algorithm to capture important local word order features. According to the experimental results, the model can effectively identify defect classes and improve short text classification ability by fusing contex-

tual features. Our model can better identify the data in the text of power equipment defects, effectively complete the data structuring process of the text, and assist to solve the emergency defect processing scheme.

As the level of intelligent grid detection increases, there will be more unstructured data related to the grid, such as images and audio. These diverse fault expressions can present defects from multiple dimensions. In the future, multi-source heterogeneity and data fusion are the development trends. We can integrate unstructured and structured data, build a Knowledge Graph in the field of electric power, and realize the query of electric power knowledge base, so as to further improve the fault diagnosis accuracy.

References

[1] Jin H W, Liu X J, Liu W W, *et al*. Analysis on ubiquitous power Internet of Things based on environmental protection [J]. *IOP Conference Series*: *Earth and Environmental Science*, 2019, **300**(4): 042077.

[2] Chen K, Mahfoud R J, Sun Y H, *et al*. Defect texts mining of secondary device in smart substation with GloVe and attention-based bidirectional LSTM[J]. *Energies*, 2020, **13** (17): 4522.

[3] Bakr H M, Shaaban M F, Osman A H, *et al*. Optimal allocation of distributed generation considering protection[J]. *Energies*, 2020, **13**(9): 2402.

[4] Liu G C, Zhao P, Qin Y, *et al*. Electromagnetic immunity performance of intelligent electronic equipment in smart substation's electromagnetic environment[J]. *Energies*, 2020, **13** (5): 1130.

[5] Sun H F, Wang Z Y, Wang J H, *et al*. Data-driven power outage detection by social sensors[J]. *IEEE Transactions on Smart Grid*, 2016, **7**(5): 2516-2524.

[6] Li Y C, Zhang P, Huang R. Lightweight quantum encryption for secure transmission of power data in smart grid[J]. *IEEE Access*, 2019, **7**: 36285-36293.

[7] Wang H F, Liu Z Q. An error recognition method for power equipment defect records based on knowledge graph technology[J]. *Frontiers of Information Technology & Electronic Engineering*, 2019, **20**(11): 1564-1577.

[8] Yu X H, Xue Y S. Smart grids: A cyber-physical systems perspective[J]. *Proceedings of the IEEE*, 2016, **104**(5): 1058-1070.

[9] Niall O M, Sean C, Anderson C, *et al*. Deep learning vs. traditional computer vision[C]// *Computer Vision Conference*. Las Vegas: CVC, 2020, **943**:128-144.

[10] Sun Y, Hu Y X, Zhang X Y, *et al*. Emotional dimension PAD prediction for emotional speech recognition [J]. *Journal of Zhejiang University* (*Engineering Science*), 2019, **53**(10): 2041-2048(Ch).

[11] Duan R, Wang Y L, Qin H X. Artificial intelligence speech recognition model for correcting spoken English teaching[J]. *Journal of Intelligent & Fuzzy Systems*, 2021, **40**(2): 3513-3524.

[12] Wu H P, Liu Y L, Wang J W. Review of text classification methods on deep learning[J]. *Computers*, *Materials and Continua*, 2020, **63**(3):1309-1321.

[13] Manickavasagam R, Selvan S, Selvan M. CAD system for lung nodule detection using deep learning with CNN[J]. *Medical & Biological Engineering & Computing*, 2022, **60** (1): 221-228.

[14] Mikolov T, Karafiat M, Burget L, *et al*. Recurrent neural network based language model [C]//11*th Annual Conference of the International Speech Communication Association*. Florence: ISCA, 2011: 2877-2880.

[15] Liu Z Q, Wang H F, Cao J, *et al*. Research on text classification model of power equipment defects based on convolutional neural networks[J]. *Power Grid Technology*, 2018, **42** (2): 644-651.

[16] Athiwaratkun B, Stokes J W. Malware classification with LSTM and GRU language models and a character-level CNN [C]//2017 *IEEE International Conference on Acoustics*, *Speech and Signal Processing*. New York: IEEE, 2017: 2482- 2486.

[17] Zennaki O, Semmar N, Besacier L. Inducing multilingual text analysis tools using bidirectional recurrent neural networks[C]//26*th International Conference on Computational Linguistics*. Osaka: COLING, 2016: 450-460.

[18] Wei D Q, Wang B, Lin G, *et al*. Research on unstructured text data mining and fault classification based on RNN-LSTM with malfunction inspection report[J]. *Energies*, 2017, **10**(3): 406.

[19] Peng H P, Li J X, He Y, *et al*. Large-scale hierarchical text classification with recursively regularized deep graph-CNN [C]// *Proceedings of the* 2018 *World Wide Web Conference-WWW*'18. New York: ACM Press, 2018: 1063-1072.

[20] Yao L, Mao C S, Luo Y. Graph convolutional networks for text classification[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, **33**(1):7370-7377.

[21] Hu L M, Yang T C, Shi C, *et al*. Heterogeneous graph attention networks for semi-supervised short text classification [C]// *Proceedings of the* 2019 *Conference on Empirical Methods in Natural Language Processing*. Stroudsburg: As-

sociation for Computational Linguistics, 2021, **39**(3): 4821-4830.

[22] Ye Z H, Jiang Y L, Li Z Y, *et al*. Document and word representations generated by graph convolutional network and bert for short text classification[C]// 24*th European Conference on Artificial Intelligence*. Spain: ECAI, 2020: 2275-2281.

[23] Li J C, Li L. A hybrid genetic algorithm based on information entropy and game theory[J]. *IEEE ACCESS*, 2020, **8**: 36602-36611.

[24] Jiao L L, Luo S L, Liu W T, *et al*. A genetic algorithm-based fuzzing method for binary programs [J]. *Journal of Zhejiang University* (*Engineering Science*), 2018, **52**(5): 1014-1019 (Ch).

[25] Fadel I A, Alsanabani H, Öz C, *et al*. Hybrid fuzzy-genetic algorithm to automated discovery of prediction rules[J]. *Journal of Intelligent & Fuzzy Systems*, 2021, **40**(1):43-52.

[26] Kim Y. Convolutional neural networks for sentence classification[C]//*Proceedings of the* 2014 *Conference on Empirical Methods in Natural Language Processing* (*EMNLP*). Stroudsburg: Association for Computational Linguistics, 2014: 1746-1751.

[27] Liu P F, Qiu X P, Huang X J, *et al*. Recurrent neural network for text classification with multi-task learning [C] // *IJCAI*'16: *Proceedings of the* 25*th International Joint Conference on Artificial Intelligence*. New York: AAAI Press, 2016: 2873-2879.

[28] Joulin A, Grave E, Bojanowski P, *et al*. Bag of tricks for efficient text classification[C]// *Proceedings of the* 15*th Conference of the European Chapter of the Association for Computational Linguistics*. Stroudsburg: Association for Computational Linguistics, 2017, **2**: 427-431.

[29] Devlin J, Chang M W, Lee K, *et al*. BERT: Pre-training of deep bidirectional transformers for language understanding [C]//*Proceedings of the* 2019 *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Stroudsburg: Association for Computational Linguistics, 2019: 4171-4186.

[30] Zhou P, Shi W, Tian J, *et al*. Attention-based bidirectional long short-term memory networks for relation classification [C]//*Proceedings of the* 54*th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg: Association for Computational Linguistics, 2016: 207-221.

□