Computer Science and Information Systems 21(4):1607–1627 https://doi.org/10.2298/CSIS240314050W

A multi-feature Fusion Model Based on Long and Short Term Memory Network and Improved Artificial Bee Colony Algorithm for English Text Classification

Tianying Wen

Department of Education, Liaoning National Normal College
No. 45, Chongdong Road, Huanggu District, Shenyang, 110032, China
sarkozyteague@foxmail.com

Abstract. The traditional methods of English text classification have two disadvantages. One is that they cannot fully represent the semantic information of the text. The other is that they cannot fully extract and integrate the global and local information of the text. Therefore, we propose a multi-feature fusion model based on long and short term memory network and improved artificial bee colony algorithm for English text classification. In this method, the character-level vector and word-level vector representations of English text are calculated using a pre-training model to obtain a more comprehensive text feature vector representation. Then the multi-head attention mechanism is used to capture the dependencies in the text sequence to improve the semantic understanding of the text. Through feature fusion, the channel features are optimized and the spatial features and time series features are combined to improve the classification performance of the hybrid model. In the stage of network training, the weighted linear combination of maximum Shannon entropy and minimum cross entropy is used as the return degree evaluation function of the bee colony algorithm, and the scale factor is introduced to adjust the solution search strategy of leading bees and following bees, and the improved artificial bee colony algorithm is combined with the classification network to realize the automatic optimization and adjustment of network parameters. Experiments are carried out on public data set. Compared with traditional convolutional neural networks, the classification accuracy of the new model increases by 2% on average, and the accuracy of data set increases by 2.4% at the highest.

Keywords: English text classification, multi-feature fusion, artificial bee algorithm, long and short term memory network, multi-head attention mechanism.

1. Introduction

Text classification is a multi-domain technology that spans information retrieval, machine learning and natural language processing, and is an important research direction of information processing and data mining. The main goal is to automatically assign the classified text to the category according to the content features or attribute features of the text when the category is defined in advance [1]. According to the length of the text, text classification is divided into short text classification and long text classification, and the number of short text characters usually does not exceed 200 [2].

Text classification methods based on deep learning can be divided into word level model, character level model and word mixed level model. Word-level model focuses on using word-level features to accurately express sentence meaning [3,4]. Because English text has a space as a separator between words, the word-level model shows excellent results in English text classification. For Chinese text, because Chinese text does not contain similar separators, the word-level model must have excellent word segmentation ability, otherwise it will lead to wrong classification results. Based on the character-level model, characters are the main features to avoid the word segmentation problem [5,6], and at the same time, the adverse effects caused by excessive word segmentation results can be solved. However, because Chinese characters have many meanings, the character-level model may not be able to accurately express the meaning of sentences.

Recurrent neural network (RNN) and convolutional neural network (CNN) are the main neural network models commonly used in text classification [7,8]. Recurrent neural networks are a kind of network for modeling sequence data. Due to problems such as gradient disappearance and gradient explosion, a variant of long short-term memory (LSTM) is often used [9]. Since LSTM can only learn the global temporal features of the text and cannot learn the local spatial features of the text, CNN is generally used to learn the local features first, and then LSTM is combined to learn the temporal features. However, the spatial dimension of the single-channel word embeddings used by the existing hybrid models is low, and the feature representation is single. As a result, the one-dimensional convolutional neural network cannot give full play to the spatial feature learning ability, which affects the classification performance of the model.

In recent years, the mixed-level model based on words and words has been proved to be an effective way and has been widely used in natural language processing, such as Chinese question answering [10], text classification, etc. Tao et al. [11] proposed a Radicalaware Attention-based Four-Granularity (RAFG) model, which combined characters, words, character-level roots and word-level roots to implement text classification. In addition, RAFG employed serialized bidirectional LSTM and attention mechanism to capture and integrate features. Hao et al. [12] proposed a Mutual-Attention Convolutional Neural Network (MACNN), which generated two alignment information matrices with two-level features through word and character features, and used convolutional neural networks to generate integrated features to improve the classification performance of English texts. However, due to the existence of one-word polysemy in English characters, these mixed-level models did not solve the problem of inaccurate character-level feature representation. Therefore, this paper proposes a multi-feature fusion model based on LSTM and improved artificial bee colony (ABC) algorithm for English text classification. Our main contributions are as follows.

1. In this method, the character-level vector and word-level vector representations of English text are calculated using a pre-training model to obtain a more comprehensive text feature vector representation.
2. Then the multi-head attention mechanism is used to capture the dependencies in the text sequence to improve the semantic understanding of the text. Through feature fusion, the channel features are optimized and the spatial features and time series features are combined to improve the classification performance of the hybrid model.
3. In the stage of network training, the weighted linear combination of maximum Shannon entropy and minimum cross entropy is used as the return degree evaluation function of the bee colony algorithm, and the scale factor is introduced to adjust the solution search strategy of leading bees and following bees, and the improved artificial

bee colony algorithm is combined with the classification network to realize the automatic optimization and adjustment of network parameters.

4. A large number of experimental results on public data sets show that the proposed method achieves good classification results.

This paper is structured as follows. In section 2, we detailed introduce the related works. Multi-label educational emotion prediction model is proposed in section 3. Experiments are conducted in section 4. There is a conclusion in section 5.

2. Related Works

Traditional machine learning algorithms [13] often need feature selection in text classification, while deep learning algorithms are widely used because they can automatically carry out feature learning. Common structures include CNN and RNN, which are suitable for processing time series data and are widely used in text classification. Jin et al. [14] proposed three models based on LSTM to deal with text classification under multi-task learning. Senthil et al. [15] used bidirectional LSTM combined with feed-forward neural network for emotion analysis. Since LSTM can only output the features of the last moment and cannot make full use of the features of each moment, some scholars try to use attention mechanism to optimize the feature representation of LSTM. Lin et al. [16] used the attention mechanism to weight the features of each moment of LSTM, achieving good results in the emotion classification task. Hu et al. [17] introduced Multi-head Attention into bi-directional LSTM for emotion classification, and achieved better results than bi-directional LSTM. Because RNNs cannot learn spatial features and the training time is long, CNNs are used in the text field. Zhoum et al.[18] used CNN for text classification, adopted multi-path convolution to extract spatial features and global maximum pooling to retain the most important features, and verified the practicability of CNN in the field of text classification through experiments. Since global maximum pooling is prone to cause a large number of feature loss, Chen et al. [19] proposed a dynamic pooling idea, which adopted different K values in different pooling layers to preserve the first K maximum features, effectively solving the problem of serious feature loss in global maximum pooling. Jia et al. [20] applied capsule neural network to text classification, and achieved better results than classical CNN on some data sets. Bang et al. [21] tried to combine the attention mechanism in CNN, which effectively improved the ability of CNN to learn local features.

Because CNN and RNN have different emphasis, many scholars put forward a hybrid model combining the advantages of both. Huang et al. [22] proposed a recurrent convolutional neural network (RCNN), which used a bidirectional cycle structure to model the context of features and realized the core idea of convolution. Moradzadeh et al. [23] put forward the hybrid model C-LSTM, which provided the mode of combining CNN and RNN. Reference [24] used multiple convolutions to learn spatial features, and learned temporal features through LSTM after fusion. Reference [25] extracted more abstract spatial features through stacked convolution pooling layers for each route, and then combined LSTM for time series feature learning after fusion. The LSTM-CNN [26] explored the feature learning mode of time sequence before space. On this basis, the BRCAN model proposed by Zhou et al. [27] used bidirectional LSTM to learn context information, and then combined CNN and attention mechanism to weight key features, achieving good classification effect on multiple data sets. Hasib et al. [28] explored a variety of attention mechanisms and conducted a comprehensive comparative evaluation. The C-HAN model proposed by Long et al. [29] divided text representation into two stages: word-sentence and sentence-document, and compared the effects of word vector and word vector on model performance. The TSOHHAN model proposed by Li et al. [30] combined the role of title in topic classification and achieved a classification accuracy rate superior to that of traditional hierarchical attention networks. Different from the small-scale shallow neural network model proposed by the above scholars, the Google team proposed a pre-trained language model BERT [31], which had achieved excellent results in a number of NLP tasks.

In terms of feature extraction, deep learning methods employ nonlinear activated neural networks to process large amounts of input data, avoiding heavy and time-consuming feature engineering. Recurrent neural networks and convolutional neural networks are widely used in the field of text classification. Kim et al. [32] applied CNN to text classification tasks for the first time and proposed a Text Convolutional Neural Network (TextCNN). TextCNN used convolution kernel of different sizes to extract local static features of text, which could effectively extract local text information. Based on TextCNN architecture, Xu et al. [33] proposed a new CNN model. First, dynamic maximum pooling was used to capture richer information in text. Secondly, in order to further improve the classification accuracy, a hidden bottleneck layer was added after the pooling layer. Ferjani et al. [34] proposed a word-level Deep Pyramid Convolutional Neural network, which enabled the model to capture long-distance dependencies in text by stacking convolutional layers and maximum pooling layers. However, the convolution operation could not take into account the text position information of the sequence. Although RNN can extract text context information and consider text position information, there is gradient explosion problem in the training process, which will seriously affect the experimental results.

In recent years, attention mechanism has been widely used in the field of text classification. Ding et al. [35] proposed a directed self-attention network, which achieved good experimental results by using only the attention mechanism without using CNN and RNN structures. Meng et al. [36] combined CNN and the attention mechanism, used the attention mechanism to calculate the weight of words with higher relevance to text semantics, and further extracted semantic features through CNN. Zhang et al. [37] proposed a Coordinated CNN-LSTM-Attention (CCLA) model. The model first learned the text sequence representation using CCLA, and then obtained the emotional tendencies in the text through the classifier.

In terms of text representation, Gong et al. [38] proposed a static word vector representation model, word2vec. Different from the previous high-dimensional sparse one-hot vector, this model could map text sequences to low-dimensional dense vectors, while also taking into account the contextual semantic information of the text in which the sequences were located.

Pre-trained models have also achieved good results in text representation. Lin et al. [39] proposed a pre-training model Bert based on Transformer. The model training task was divided into two steps. The first step was unsupervised pre-training using large-scale corpus. The second step was to fine-tune the model to accommodate different NLP tasks. Li et al [40] proposed Enhanced Language Representation with Informative Entities (ERNIE) on the basis of Bert. ERNIE improved the mask semantic model by using global information to predict parts of the mask, and by using the information entities in the knowledge graph, so that the model could better learn the complete semantic representation. Peng et al. [41] proposed ERNIE2.0 for sustainable learning. The model used a large number of corpus models for unsupervised training tasks, and then used pre-training tasks to continuously update the model to further improve the language representation ability of the model.

Although scholars have proposed a variety of hybrid models, the existing hybrid models still have the following problems.

1. Single-channel word embedding is commonly used, the spatial dimension is low, the feature representation of text is single, and only one-dimensional convolution algorithm can be used on a single channel, which cannot give full play to the spatial feature learning ability of convolution;
2. When the existing CNN-RNN hybrid model fuses multi-channel convolution features, the feature timing after fusion is often damaged, which affects the subsequent LSTM layer's learning process of timing features.

Therefore, in order to make full use of dual-channel features, the model proposed in this paper first learns spatial features independently in the two channels, and then uses point-by-point convolution fusion channel features to enhance the spatial feature learning ability of the convolution layer. When fusing multi-path convolutional features, bidirectional LSTM combined with attention mechanism is used in each path for timing feature learning, and the features of each path are concatenated to represent text, effectively avoiding the problem that the multi-path convolutional feature fusion process will cause damage to the timing feature after fusion before entering LSTM. At the same time, we use an improved ABC to optimize the network parameters and further optimize the effect. Experiments show that the proposed text classification model achieves good classification performance on multiple data sets.

3. Proposed English Text Classification Model

Although the traditional word mixing model has a good ability to express features, the character-level features fail to take into account the polysemy of English characters. Therefore, this paper proposes a English text classification method with multi-feature fusion model based on long and short term memory network and improved artificial bee colony algorithm to achieve the task of English text classification, as shown in Figure 1.

3.1. Input Layer

The input layer mainly solves the problem of text sequence partitioning. Consider a English sentence S, which is divided into two distinct sequences: word-level $S_{word} = w_1, w_2, \cdots, w_n$ and character-level $S_{char} = c_1, c_2, \cdots, c_t$. Where n and t are the sequence lengths obtained by dividing sentences according to word and word respectively. In order to better illustrate the problem of sequence partitioning, taking ”Today we see the film and eat watermelon” as an example, Figure 2 shows the results of word-level sequence and character-level sequence partitioning in English text.

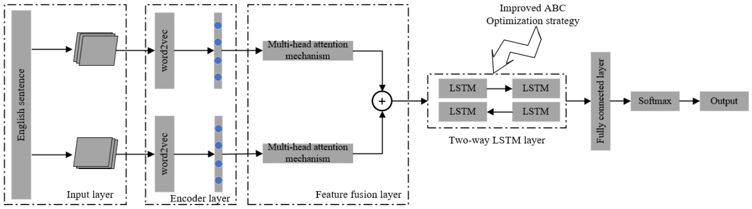

Fig. 1. Proposed English text classification model

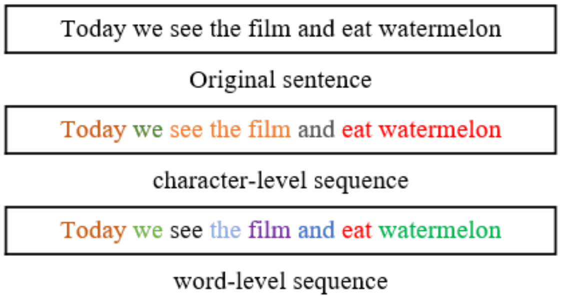
Today we see the film and eat watermelon
Original sentence
Today we see the film and eat watermelon
character-level sequence
Today we see the film and eat watermelon
word-level sequence

Fig. 2. Text sequence division

3.2. Encoder Layer

The main function of the encoding layer is to represent the sequence of text from the input layer in a continuous space. It receives two levels of features, the word vector representation and the word vector representation, (i.e. S_{char} and S_{word}) and outputs two embedded matrices through a pre-trained model.

Traditional language representation methods are static and can not solve the problem of polysemy in English text. Therefore, this paper uses ERNIE to complete the character-level vector representation of text, and optimizes the traditional static representation method to dynamic representation method. ERNIE improves the mask strategy on the basis of BERT, and introduces the enhanced language representation model of knowledge graph, which can complete the vectoring representation of Chinese characters more accurately. The ERNIE model adopts multi-layer Transformer module and uses bidirectional Transformer structure for text vectorization. The text sequence S_{char} is first entered into multiple self-attention layers to obtain contextual information. Then, the context information is input into the Normalize layer for residual connection and normalization operations, and then the linear change is processed through the feed-forward neural network. After repeating the previous step, the dynamic word vector representation is obtained by combining the knowledge external entity information and the prior semantic information of the masking strategy. Through the vector representation obtained by training, the coding layer converts the character-level text sequence S_{char} into a character-level vector $e_{char} = [e^1_{char}, e^2_{char}, \cdots, e^t_{char}]$.

In this paper, the common word2vec model is used to calculate the word-level vector of text. word2vec is a model for training static word vectors that can map high-dimensional, sparse one-hot vectors to low-dimensional, dense word vectors. word2vec has two training modes [42], namely CBOW and Skip-gram. Because Skip-gram is better than CBOW in learning rare words, Skip-gram is more accurate. Therefore, this paper uses Skip-gram mode to train word vectors. By training the word vector, the input text sequence S_{word} is transformed into the word level vector $e_{word} = [e^1_{word}, e^2_{word}, \cdots, e^n_{word}]$.

3.3. Feature Fusion Layer

The feature layer aims to generate a comprehensive feature representation of the input text S by combining contextual word-level features with character-level features. Since the word-level and character-level embedding matrices are located in two independent Spaces, direct fusion of features along time series will damage text representation information, so this paper adopts two independent feature layers to extract character-level and character-level text features respectively. The feature layer consists of the attention layer and the convolution layer, and the details are as follows.

The attention layer. In order to integrate features, highlight the features with high relevance to the text semantics, and enhance their weight. This paper uses Transformer [43] multi-head attention encoder to capture full text context information. Multiple attention is calculated using multiple attention, and then the results of each attention are spliced to obtain different levels of semantic information. The calculation is shown in formula (1):

$$h_i = softmax(\frac{QK^T}{\sqrt{d_k}})V. \tag{1}$$

Where, Q, K, V are the weighted matrix, d is the dimension of the vector, softmax function normalizes the output result row.

Next, combine each self-attention output to obtain multi-head attention output, which is calculated as shown in formula (2):

$$Multihead = Concat(h_1, h_2, \cdots, h_y)W^T. \tag{2}$$

Where y is the total amount of self-attention and W^T is the matrix for transforming the dimensions of the concatenation results.

The convolution layer. In order to prevent semantic information loss in the attention layer, the feature extracted from the character level vector and word level vector and the text embedding matrix in the coding layer are folded into two two-dimensional tensors respectively. Similar to a picture with three channels, these stacked feature tensors can be viewed as an image with two channels. After stacking features into two-dimensional tensors, the convolution layer can extract feature information of different dimensions from multi-channel feature vectors.

The convolutional layer is composed of CNN, Relu function and one-dimensional maximum pooling, and its structure is shown in Figure 3. It has a strong local feature extraction ability, which can capture important phrase features. Moreover, this convolution layer uses multiple convolution kernels for feature mapping, which can extract more feature information than a single convolution kernels. In a convolutional neural network, a sub-feature c_i is calculated by a filter w and a feature window $z_{i:i+k-1}$, as shown in the formula (3):

$$c_i = f(w * z_{i:i+k-1} + b). \tag{3}$$

Where b is the biased term and f is the Relu function. The filter is applied to the feature representation $z = z_1, z_2, \cdots, z_{l-k+1}$, resulting in the feature vector:

$$C = [c_1, c_2, \cdots, c_{l-k+1}]. \tag{4}$$

Where, k is the step dimension length and l is the text sequence length. In a two-channel architecture, the filter acts on each channel. And after calculating c_i, it is entered into the pooling layer. The role of the pooling layer is to reduce the model parameters and reduce the risk of over-fitting the model while preserving the main features.

In addition, j types of convolution kernel are used in this paper, and the number of each convolution kernel is m. The output result of each convolution kernel is $H_i (1 \leq i \leq j)$, and its calculation formula is shown in equation (5):

$$H_i = concat(max(c_1), max(c_2), \cdots, max(c_m)). \tag{5}$$

max is a one-dimensional pooling operation and $concat$ is a concatenation operation.

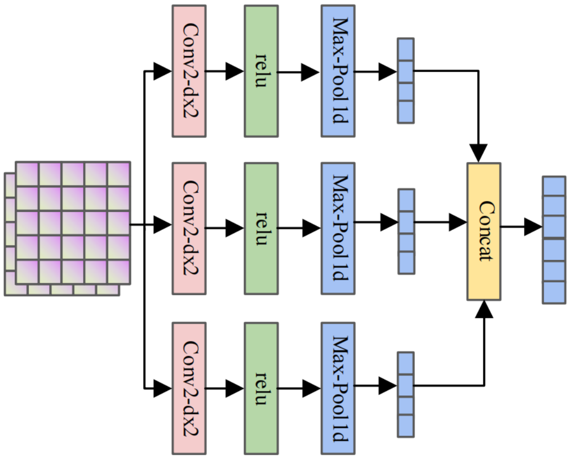

Fig. 3. Convolution layer structure

3.4. Two-way LSTM Layer

For long texts, the context information of words is sufficient and there are often long distance semantic associations. Compared with short texts with few features and insufficient time sequence information, long texts have higher requirements for the time sequence of features. Before the feature is input to LSTM, the multi-path convolution is fused first, which cannot guarantee the timing of the features after fusion, and greatly affects the

LSTM learning process of long text timing features. Let M_1 and M_2 respectively represent the feature graph matrix formed by the convolution of different paths. If M_1 and M_2 are joined horizontally, due to the difference in the convolution kernel size, M_1 and M_2 are different in row dimension, and can only be filled with 0, so that the convolution feature map size remains unchanged, which will cause the timing features of M_1 and M_2 cannot be completely aligned, resulting in the overall quality of the timing feature decline. If M_1 and M_2 are spliced vertically, the global order of the overall features cannot be guaranteed after splicing.

In order to avoid the shortcomings of the above fusion methods, the model in this paper uses bidirectional LSTM to learn the timing features of each route, and combines the bidirectional timing features of each route to represent the final text, avoiding the problem of declining the quality of timing features caused by the fusion of various features before entering the LSTM. Since the traditional forward LSTM can only learn the above information of features, ignoring the below information of features, this paper uses two-way LSTM to learn the context information of features at the same time, which greatly improves the learning ability of temporal features of the model. In order to make full use of the output features of all LSTM moments, the model uses the attention mechanism to carry out weighted summation of the features of each LSTM moment to improve the output quality of LSTM. The bidirectional LSTM layer in this paper is shown in Figure 4.

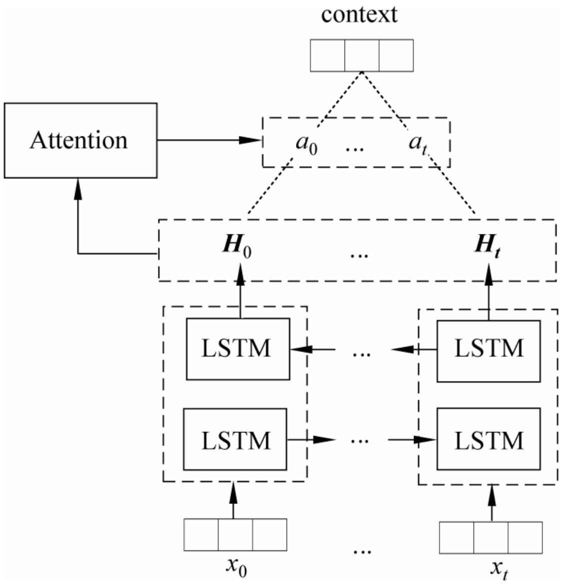

Fig. 4. Bidirectional LSTM layer

Let i represent the $i - th$ time and $i \in [0, t]$. x_i represents the input vector at time i. M_k represents the feature graph matrix formed by one convolution, then M_k can be expressed as the concatenation of multiple row vectors, as shown in equation (6).

$$M_k = x_0 \oplus x_1 \oplus \cdots \oplus x_t. \tag{6}$$

LSTM receives x_i as an input vector in chronological order. c_t indicates the LSTM unit status. h_t indicates the final output of the LSTM unit. f_t, i_t, o_t indicate the memory gate, input gate, and output gate respectively. σ indicates the Sigmoid activation function. W_f, W_i, W_o, W_c, b_f, b_i, b_o, b_c indicate the parameters that the network needs to learn. The final output calculation of LSTM is shown in equations (7-12). Since the model uses the bidirectional LSTM learning timing feature, the final output of bidirectional LSTM is obtained by combining the forward LSTM output with the reverse LSTM output.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \tag{7}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i). \tag{8}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o). \tag{9}$$

$$q_t = \sigma(W_c \cdot [h_{t-1}, x_t] + b_c). \tag{10}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot q_t. \tag{11}$$

$$h_t = o_t \tanh(c_t). \tag{12}$$

Since LSTM can only learn the output vector of the last moment and cannot make full use of the output of each moment, this paper uses the attention mechanism to complete the weighted fusion of the output features of each moment. Let H_i represent the output vector of the bidirectional LSTM layer at time i. e_i indicates how important H_i is to the overall semantic representation of the text. a_i represents the weight of H_i's contribution to the overall semantic representation of the text. According to the above definition, the attention weight of the bidirectional LSTM layer is calculated as shown in equations (13) and (14).

$$e_i = u^T \cdot \tanh j(W_a \cdot H_i + b_a). \tag{13}$$

$$a_i = \frac{exp(e_i)}{\sum_{j=0}^{t} exp(e_i)}. \tag{14}$$

Among them, u^T, W_a and b_a are the parameters that the network needs to learn. $\tanh$ is a nonlinear activation function. After the attention weights of each moment of the bidirectional LSTM layer are obtained, the output vectors of all moments of the bidirectional LSTM layer are weighted and summed by equation (15). The resulting vector v is the characteristic vector of the final output of the entire bidirectional LSTM layer.

$$v = \sum_{i=0}^{t} a_i H_i. \tag{15}$$

Let v_i represent the document representation vector learned after the $i-th$ way convolutional features pass through the bidirectional LSTM layer, then the document representation vector v_d finally formed by the model can be represented as the concatenation of n-way convolutional document representation vectors, as shown in equation (16).

$$v_d = v_1 \oplus v_2 \oplus \cdots v_n. \tag{16}$$

After the final representation vector v_d of the text is obtained, the final category output of v_d is carried out through the full connection layer and the Softmax layer. Let c represent a class. n indicates the number of categories. d represents the output vector of the document vector v_d after passing through the fully connected layer. d_c represents the value of the component of the vector d belonging to class c. p_c represents the probability that the text is classified as c. W_c and b_c are parameters that need to be learned in the fully connected layer network. f is a nonlinear activation function, then p_c is calculated as shown in equations (17) and (18).

$$d = f(W_c \cdot v_d + b_c). \tag{17}$$

$$p_c = \frac{exp(d_c)}{\sum_{k=1}^{n} exp(d_k)}. \tag{18}$$

3.5. Network Parameter Optimization Based on Improved ABC

Artificial bee colony algorithm (ABC) is an optimization method to imitate the honey collecting behavior of bees [44]. The minimum search model for generating swarm intelligence consists of three basic components: honey source, hired bees (also known as lead bees), and unhired bees (including scout bees and follower bees). Two basic behavioral models include recruiting bees for a honey source and giving up a honey source. The honey source is used to represent the solution of the problem, and the value of the honey source is expressed by the income degree function, which is the optimization object function. The guide bees correspond to the honey source one by one, which is used to store the relevant information of the corresponding honey source and share it with other bees with a certain probability. The main task of scout bees and follower bees is to find and extract honey. The process of bees gathering honey is the process of finding the optimal solution of the objective function. The basic swarm algorithm flow is shown in Figure 5.

However, like all other evolutionary algorithms, ABC also has the inherent problem of too slow or too fast convergence. Therefore, this paper adopts anti-learning algorithm, and proposes an improved artificial bee colony algorithm to improve the strategies of leading and following bees to search for honey sources by introducing scale factors, so as to realize the adaptive adjustment of the artificial bee colony algorithm. The specific policy modification expression is as follows.

$$v_{ij} = x_{ij} + \phi(x_{ij} - x_{kj}). \tag{19}$$

Where, $i = 1, 2, \cdots, N$, $j = 1, 2, \cdots, D$. Where v is used to indicate the location of pre-search for new nectar sources. x is used to indicate the specific location of the current nectar source. $\phi \in (-1, 1)$ is a scale regulating factor, which determines the scale

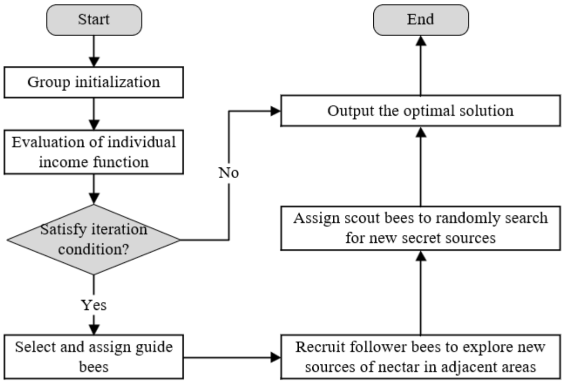
Start
Group initialization
Evaluation of individual income function
Satisfy iteration condition?
Yes
No
Select and assign guide bees
Recruit follower bees to explore new sources of nectar in adjacent areas
Assign scout bees to randomly search for new secret sources
Output the optimal solution
End

Fig. 5. ABC flow chart

of movement from the current solution to the next feasible solution. N is the number of nectar sources. D is the dimension of the solution. The expression for ϕ is as follows.

$$\phi(G+1) = \left\{ \phi_l + rand_1 \sqrt{rand_2^2 + rand_3^2} \quad if P_i < rand_4 \phi_u \quad otherwise \right. \tag{20}$$

Where, $rand_i \in [0, 1]$, $i = 1, 2, 3, 4$. P_i is the selection probability value of the $i-th$ leader bee, that is, $P_i = \frac{Fitval_i}{\sum_{k=1}^{N} Fitval_k}$. According to the above adjustment principle, the calculation expression of the $G+1-th$ generation scaling factor ϕ_{G+1} is as follows.

$$\phi(G+1) = \{ 2\phi_l - \phi(G+1) \quad if \phi(G+1) < \phi_l 2\phi_u - \phi(G+1) \quad if \phi(G+1) > \phi_u \tag{21}$$

Among them, $Fitval_i$ is the return degree evaluation function value of the $i-th$ honey source. $rand_i$ is a random number between [0,1]. ϕ_u and ϕ_l are used to set the boundary of the range value of $\phi(G+1)$ under manual intervention.

The modified artificial bee colony algorithm is more close to the natural process of bees to find honey sources, and can ensure the diversity and balance of solutions in the algorithm.

In this paper, the linear weighting function of the maximum Shannon entropy and the minimum cross entropy is used as the objective evaluation function of the network classification result (i.e. the return degree function of the bee colony algorithm). By adjusting the scale factor ϕ of the solution adjustment strategy of the leading bees and the following bees, an improved artificial bee colony algorithm is proposed to optimize the parameters of the classification network. The step of the algorithm is as follows.

1. Group initialization. At the initial time, N feasible solutions are randomly generated $X = (\beta, V_\theta, a_\theta)$. The resulting formula is as follows.

$$X_i^j = X_{min}^j + rand(0,1)(X_{max}^j - X_{min}^j). \tag{22}$$

Where, $i = 1, 2, 3$. $\beta \in [0.001, 200]$, $V_\theta \in [0.001, 200]$, $Va_\theta \in [0.001, 100]$.

2. Automatically adjust the solution search strategy of leading and following bees.
3. Set the income degree function. In this paper, the linear weighted sum of maximum Shannon entropy and minimum cross entropy is used as the evaluation standard for parameter optimization adjustment of each generation.

$$fit = \rho H_1(p) - (1-\rho)H_2(p). \tag{23}$$

$$H_1(p) = p_1 \log_2 p_1 + p_0 \log_2 p_0. \tag{24}$$

$$H_2(p) = p_1 \log_2 \frac{p_0}{p_1} + p_0 \log_2 \frac{p_1}{p_0}. \tag{25}$$

Where $H_1(p)$ and $H_2(p)$ represent the maximum Shannon entropy and the minimum cross entropy in the fitness function in sequence. p_1 and p_0 represent the probability that the output value of the network is 1 and 0. ρ is the weighting coefficient of the fitness function, and $\rho \in [0,1]$. Where, if ρ takes 1 or 0 as the evaluation function, the two special cases are the cases where the maximum aroma entropy and the lowest cross entropy are adopted. Then the return degree evaluation function of the improved artificial bee colony algorithm is:

$$Fitval(\beta, V_\theta, a_\theta) = \begin{cases} \dfrac{1}{1+fit} & if fit \geq 01 + |fit| \quad if fit < 0 \end{cases} \tag{26}$$

4. Algorithm termination condition. To improve the termination conditions of the artificial bee colony algorithm, the local maximum iteration times (threshold $limit$ and global maximum iteration times) are set to ensure the termination of the algorithm.

3.6. Output Layer

Through the feature layer, the feature vector V_w based on the word level and the feature vector V_c based on the character level are obtained respectively, and the two feature vectors are spliced to obtain the fused feature vector V, which is the final representation of the input text. Then, input V into the fully connected neural network to obtain the output vector $O \in R^K$(K is the number of categories), as shown in formula (27):

$$O = sigmoid(V * W_f). \tag{27}$$

Where, W_f is the weight matrix of a fully connected network. $sigmoid$ is a nonlinear activation function. Finally, the values in O are mapped to conditional probabilities S through softmax, and the category with the highest probability of conditional probabilities is the prediction category. The calculation of S is shown in formula (28):

$$S = argmax(softmax(O)). \tag{28}$$

4. Experimental Results and Analysis

This section will first introduce the data set used in the experiment, the experimental environment and parameter settings, and the baseline method. Then, the experimental evaluation of the proposed method and baseline method on three data sets is given. Finally, the ablation experiment and comparison with the advanced method is given.

4.1. Dataset

The classification task is performed on three English text data sets: Reuters, the English Journal Articles Dataset (EJA), and the Agency French Press (AFP). Summary statistics for the data set are shown in Table 1.

Table 1. Dataset

Dataset	Training	Verification	Testing	Category number	Maximum length
Reuters	40000	5000	5000	14	32
EJA	10000	1000	1000	10	32
AFP	200000	20000	20000	15	36

EJA. This data set is an English journal paper dataset containing article titles. 10000 samples are randomly selected for training, 1000 samples for validation and 1000 samples for testing. The first 32 characters (including punctuation) are taken from each sample to form a new dataset.

Reuters. This dataset is a social news dataset with headlines and contains 40000 samples for training, 5000 samples for validation, and 5000 samples for testing. The first 32 characters (including punctuation) are taken from each sample to form a new dataset.

AFP. This dataset is divided into 15 categories, with 200000 samples reserved for training, 20000 for validation, and 20000 for testing. The first 36 characters (including punctuation characters) are taken from each sample as a new dataset.

4.2. Experimental Environment and Parameter Setting

This experiment is conducted in Python 3.7 and Pytorch 1.8.0, using an Intel(R) Core i5-10600KF CPU and NvidiaRTX 3050Ti [45].

The main parameter Settings of the experimental model are shown in Table 2. On EJA and Reuters data sets, the padding size is set to 32; On the AFP dataset, the padding size is set to 36. At the same time, in order to avoid model over-fitting and ineffective consumption of computing resources, the training process of the model is terminated in advance when the training effect of the model is not improved after 1000 batches. In addition, when using the word2vec model to train the word vector in this paper, the dimension of the word vector is 300, the size of the training window is 5, the threshold of word frequency is 5, and the number of training iterations is 5.

4.3. Baseline Method

In order to verify the validity of the proposed method in English text classification, this paper compares the proposed method with the following methods.

TextRNN. It respectively inputs words, characters and words mixed into word2vec to get text features, and apply TextRNN as a classifier. TextRNN uses the forward LSTM module and the backward LSTM module respectively to obtain the sum of the forward and backward hidden vectors, and finally concatenates the two into the softmax layer to obtain the classification result.

Table 2. Parameter configuration

Parameter	value
batch-size	64
learning rate	$5e^{-5}$
CNN filter size	2,3,4
CNN filter num	256
epoch	30
dropout	0.4
Multihead attention heads	5

RCNN. It inputs words, characters and words into word2vec respectively to get text features, and uses RCNN as a classifier. RCNN is a hybrid network model composed of two layers. In the first layer, RNN is applied to learn the text representation with word embeddings, and in the second layer, CNN and maximum pooling are applied to extract the most obvious features.

ERNIE. ERNIE designed a new continuous multi-paradigm unified pre-training framework and utilized specific self-attention codes to control the content of predicted conditions.

Attention+CNN. It takes ERNIE and word2vec output as text features, and applies the attention mechanism and CNN as classifiers.

When the accuracy is unbalanced or irregular, the model cannot be evaluated accurately [46]. The F1 value is used as the performance evaluation index, and the accuracy rate and recall rate are used as the reference. F1 is shown in formula (29).

$$F1 = \frac{2PR}{P+R}. \tag{29}$$

Precision (P) indicates the precision rate, and Recall (R) indicates the recall rate.

4.4. Results Analysis

To evaluate the effectiveness of proposed method, experiments are conducted on three datasets. Taking accuracy rate, recall rate and F1 as evaluation indicators, the experimental results are shown in Table 3, Table 4 and Table 5. It can be noted that the model proposed in this paper achieves higher performance than other baseline methods. In order to find the internal reasons, the experimental results are analyzed in detail as follows.

The experimental results are shown in Table 3, Table 4 and Table 5. By comparing the experimental results of TextRNN-w, TextRNN-c and TextRNN-wc on the three datasets, it can be found that TextRNN-wc has the best classification performance, followed by TextRNN-w. The results show that TextRNN-wc with two-level features has a higher F1 value. This not only proves that character-level features and word-level features can be used well together, but also shows that character-level features and word-level features promote each other. At the same time, similar conclusions can be drawn from the comparison results of RCNN-w, RCNN-c and RCNN-wc.

Then, comparing the proposed method with the experimental results of ERNIE, the performance of the proposed method is obviously better. The results show that combining

Table 3. Results on EJA/%

Model	P	R	F1
TextRNN-w	79.88	78.84	79.36
TextRNN-c	79.15	78.31	78.73
TextRNN-wc	80.19	79.67	79.93
RCNN-w	78.26	77.54	77.90
RCNN-c	79.22	78.62	78.92
RCNN-wc	80.87	80.41	80.64
ERNIE	83.66	83.10	83.38
Attention+CNN	85.65	85.43	85.54
Proposed	86.48	86.24	86.36

Table 4. Results on Reuters/%

Model	P	R	F1
TextRNN-w	92.07	91.39	91.73
TextRNN-c	91.86	91.46	91.66
TextRNN-wc	92.55	91.81	92.18
RCNN-w	93.72	93.38	93.55
RCNN-c	91.66	91.30	91.48
RCNN-wc	94.88	94.40	94.64
ERNIE	93.67	93.61	93.64
Attention+CNN	94.85	94.97	94.91
Proposed	95.81	95.71	95.76

the dynamic feature representation model ERNIE with the traditional static feature representation model word2vec can improve the performance of text classification. In addition, the experimental results of TextRNN-wc, RCNN-wc and proposed method show that the proposed method uses dynamic word vectors to represent character-level features, which can represent text information more accurately and contribute to the representation of text features.

Finally, on the three data sets, the proposed method has better classification performance than Attention+CNN. Although Attention+CNN introduces CNN and attention mechanism, it does not consider that there may be feature information loss in the process

Table 5. Results on AFP/%

Model	P	R	F1
TextRNN-w	91.96	92.56	92.26
TextRNN-c	91.83	91.53	91.78
TextRNN-wc	92.87	92.51	92.69
RCNN-w	93.12	92.42	92.77
RCNN-c	92.63	92.37	92.50
RCNN-wc	93.74	93.56	93.65
ERNIE	94.29	94.15	94.22
Attention+CNN	95.03	94.89	94.96
Proposed	96.14	96.10	96.12

of feature extraction. By stacking text embedding matrix before extracting local key features from CNN, the proposed method retains more text information, which can make the model obtain higher performance.

4.5. Ablation Experiment

Although the above experiments prove that the proposed method has good text classification performance, in order to display the functions of each part of the model. In this section, we will conduct ablation experiments on the proposed method in order to better demonstrate the contributions of each part of the model. The results of ablation experiments are shown in Table 6, Table 7, and Table 8.

Table 6. Ablation results on EJA/%

Model	P	R	F1
LSTM	83.76	83.38	83.57
LSTM+ABC	85.44	85.20	85.32
LSTM+attention	84.50	84.66	84.58
Proposed	86.48	86.24	86.36

Table 7. Ablation results on Reuters/%

Model	P	R	F1
LSTM	94.03	93.77	93.90
LSTM+ABC	94.86	94.74	94.80
LSTM+attention	95.34	95.20	95.27
Proposed	95.81	95.71	95.76

Table 8. Ablation results on AFP/%

Model	P	R	F1
LSTM	93.26	92.98	93.12
LSTM+ABC	95.22	95.12	95.17
LSTM+attention	94.83	95.01	94.92
Proposed	96.14	96.10	96.12

The results of the ablation experiment are shown in Table 6, Table 7 and Table 8. Comparing LSTM with LSTM+attention, it can be seen that LSTM+attention has better classification performance than LSTM. This experimental result proves that the attention layer can highlight the features that are highly relevant to text semantics and enhance their weight. From the experiments of LSTM and LSTM+ABC, it can be found that

LSTM+ABC has higher classification performance. This shows that the network parameter optimization based on ABC is helpful for text classification. Finally, the four cases are compared, and the proposed method achieves higher performance on the three data sets. The results show that the convolution layer can make up for the lack of local feature extraction ability in the attention layer, and the model is more efficient when the attention layer constructs richer semantic information.

Finally, we choose three advanced methods to compare with the method in this paper including GMT [47], SSF [48], SCLTW [49]. From the table 9, the proposed method in this paper obtains the optimal F1 value.

Table 9. Comparison results with different method (F1)/%

Model	GMT	SSF	SCLTW	Proposed
EJA	79.88	82.63	85.25	86.36
Reuters	91.08	92.71	94.65	95.76
AFP	91.92	93.25	95.33	96.12

5. Conclusion

Aiming at the difficulty of text representation and feature extraction in English text classification tasks, this paper proposes a method of English short text classification with mixed features and multiple attention. Firstly, combining word-level features and character-level features, multilevel feature representation of text data is carried out. Then the attention layer and convolution layer are used to extract the global and local features of the text respectively. Finally, the two features are fused and text categories are predicted by the full connection layer and softmax layer. At the same time, the improved ABC is used to optimize the network parameters, and better classification effect is obtained. Experiments are carried out on three public data sets and compared with the current mainstream classification models. The experimental results show that the proposed method has good classification performance compared with the baseline model. Although the method proposed in this paper has certain performance, it also has some obvious shortcomings. The number of parameters in the model is large and the calculation cost is high. The next step will be to verify the effectiveness of the algorithm on more data sets. At the same time, considering the deficiency of the model, the model is studied in a finer granularity.

References

1. Jha V, Savitha R, Shenoy P D, et al. A novel sentiment aware dictionary for multi-domain sentiment classification[J]. Computers & Electrical Engineering, 2018, 69: 585-597.
2. Yuan Z, Wu S, Wu F, et al. Domain attention model for multi-domain sentiment classification[J]. Knowledge-Based Systems, 2018, 155: 1-10.
3. Minaee S, Kalchbrenner N, Cambria E, et al. Deep learning–based text classification: a comprehensive review[J]. ACM computing surveys (CSUR), 2021, 54(3): 1-40.

4. X. Meng, X. Wang, S. Yin, et al. Few-shot image classification algorithm based on attention mechanism and weight fusion[J]. Journal of Engineering and Applied Science. 70, 14 (2023). https://doi.org/10.1186/s44147-023-00186-9.
5. Kowsari K, Brown D E, Heidarysafa M, et al. Hdltex: Hierarchical deep learning for text classification[C]//2017 16th IEEE international conference on machine learning and applications (ICMLA). IEEE, 2017: 364-371.
6. Sun N, Du C. News text classification method and simulation based on the hybrid deep learning model[J]. Complexity, 2021, 2021: 1-11.
7. Lyu S, Liu J. Convolutional recurrent neural networks for text classification[J]. Journal of Database Management (JDM), 2021, 32(4): 65-82.
8. Du C, Huang L. Text classification research with attention-based recurrent neural networks[J]. International Journal of Computers Communications & Control, 2018, 13(1): 50-61.
9. Sari W K, Rini D P, Malik R F. Text Classification Using Long Short-Term Memory with GloVe[J]. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI), 2019, 5(2): 85-100.
10. Liu J, Yang Y, Lv S, et al. Attention-based BiGRU-CNN for Chinese question classification[J]. Journal of Ambient Intelligence and Humanized Computing, 2019: 1-12.
11. Tao H, Tong S, Zhao H, et al. A radical-aware attention-based model for chinese text classification[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01): 5125-5132.
12. Hao M, Xu B, Liang J Y, et al. Chinese short text classification with mutual-attention convolutional neural networks[J]. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 2020, 19(5): 1-13.
13. Wang P, Fan E, Wang P. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning[J]. Pattern Recognition Letters, 2021, 141: 61-67.
14. Jin N, Wu J, Ma X, et al. Multi-task learning model based on multi-scale CNN and LSTM for sentiment classification[J]. IEEE Access, 2020, 8: 77060-77072.
15. Senthil Kumar N K, Malarvizhi N. Bi-directional LSTM-CNN combined method for sentiment analysis in part of speech tagging (PoS)[J]. International Journal of Speech Technology, 2020, 23: 373-380.
16. Lin J, Ma J, Zhu J, et al. Short-term load forecasting based on LSTM networks considering attention mechanism[J]. International Journal of Electrical Power & Energy Systems, 2022, 137: 107818.
17. Hu Z, Chen L, Luo Y, et al. Eeg-based emotion recognition using convolutional recurrent neural network with multi-head self-attention[J]. Applied Sciences, 2022, 12(21): 11255.
18. Zhou H. Research of text classification based on TF-IDF and CNN-LSTM[C]//Journal of Physics: Conference Series. IOP Publishing, 2022, 2171(1): 012021.
19. Chen J, Kakillioglu B, Velipasalar S. Background-aware 3-D point cloud segmentation with dynamic point feature aggregation[J]. IEEE Transactions on Geoscience and Remote Sensing, 2022, 60: 1-12.
20. Jia X, Wang L. Attention enhanced capsule network for text classification by encoding syntactic dependency trees with graph convolutional neural network[J]. PeerJ Computer Science, 2022, 8: e831.
21. Bang J, Park J, Park J. GACaps-HTC: graph attention capsule network for hierarchical text classification[J]. Applied Intelligence, 2023: 1-18.
22. Huang W, Lin M, Wang Y. Sentiment analysis of Chinese e-commerce product reviews using ERNIE word embedding and attention mechanism[J]. Applied Sciences, 2022, 12(14): 7182.
23. Moradzadeh A, Teimourzadeh H, Mohammadi-Ivatloo B, et al. Hybrid CNN-LSTM approaches for identification of type and locations of transmission line faults[J]. International Journal of Electrical Power & Energy Systems, 2022, 135: 107563.

24. Singh R, Saurav S, Kumar T, et al. Facial expression recognition in videos using hybrid CNN & ConvLSTM[J]. International Journal of Information Technology, 2023, 15(4): 1819-1830.
25. Chung J, Jang B. Accurate prediction of electricity consumption using a hybrid CNN-LSTM model based on multivariable data[J]. PloS one, 2022, 17(11): e0278071.
26. ETNER H. Multi-Label Text Analysis With A CNN And LSTM Based Hybrid Deep Learning Model[J]. Adyaman Üniversitesi Mühendislik Bilimleri Dergisi, 2022, 9(17): 447-457.
27. Zhou S, Guo S, Du B, et al. A Hybrid Framework for Multivariate Time Series Forecasting of Daily Urban Water Demand Using Attention-Based Convolutional Neural Network and Long Short-Term Memory Network[J]. Sustainability, 2022, 14(17): 11086.
28. Hasib K M, Azam S, Karim A, et al. Mcnn-lstm: Combining cnn and lstm to classify multi-class text in imbalanced news data[J]. IEEE Access, 2023.
29. Long S, Han S C, Wan X, et al. Gradual: Graph-based dual-modal representation for image-text matching[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2022: 3459-3468.
30. Li Y A, Han C, Jiang X, et al. Phoneme-Level Bert for Enhanced Prosody of Text-To-Speech with Grapheme Predictions[C]//ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023: 1-5.
31. Kim Y, Kim J H, Lee J M, et al. A pre-trained BERT for Korean medical natural language processing[J]. Scientific Reports, 2022, 12(1): 13847.
32. Soni S, Chouhan S S, Rathore S S. TextConvoNet: A convolutional neural network based architecture for text classification[J]. Applied Intelligence, 2023, 53(11): 14249-14268.
33. Xu X, Li D, Zhou Y, et al. Multi-type features separating fusion learning for Speech Emotion Recognition[J]. Applied Soft Computing, 2022, 130: 109648.
34. Ferjani I, Hidri M S, Frihida A. SiNoptiC: Swarm intelligence optimisation of convolutional neural network architectures for text classification[J]. International Journal of Computer Applications in Technology, 2022, 68(1): 82-100.
35. Ding Y, Jia M, Miao Q, et al. A novel time-frequency Transformer based on self-attention mechanism and its application in fault diagnosis of rolling bearings[J]. Mechanical Systems and Signal Processing, 2022, 168: 108616.
36. Meng S, Zhou Y, Gao Z. Refined self-attention mechanism based real-time structural response prediction method under seismic action[J]. Engineering Applications of Artificial Intelligence, 2024, 129: 107380.
37. Zhang Y, Zheng J, Jiang Y, et al. A text sentiment classification modeling method based on coordinated CNN-LSTM-attention model[J]. Chinese Journal of Electronics, 2019, 28(1): 120-126.
38. Gong J, Zhang J, Guo W, et al. Short Text Classification Based on Explicit and Implicit Multiscale Weighted Semantic Information[J]. Symmetry, 2023, 15(11): 2008.
39. Lin X, Xiong G, Gou G, et al. Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification[C]//Proceedings of the ACM Web Conference 2022. 2022: 633-642.
40. Li J, Katsis Y, Baldwin T, et al. SPOT: Knowledge-Enhanced Language Representations for Information Extraction[C]//Proceedings of the 31st ACM International Conference on Information & Knowledge Management. 2022: 1124-1134.
41. Peng B, Zhang T, Han K, et al. BVMHA: Text classification model with variable multihead hybrid attention based on BERT[J]. Journal of Intelligent & Fuzzy Systems (Preprint): 1-12.
42. F. Xiao, S. Yu and Y. Li, ”Efficient Large-Capacity Caching in Cloud Storage Using Skip-Gram-Based File Correlation Analysis,” in IEEE Access, vol. 11, pp. 111265-111273, 2023, doi: 10.1109/ACCESS.2023.3322725.
43. L. Teng et al., ”FLPK-BiSeNet: Federated Learning Based on Priori Knowledge and Bilateral Segmentation Network for Image Edge Extraction,” in IEEE Transactions on Network and Service Management, vol. 20, no. 2, pp. 1529-1542, June 2023, doi: 10.1109/TNSM.2023.3273991.

44. S. Yin, J. Liu, L. Teng. An Improved Artificial Bee Colony Algorithm for Staged Search[J]. TELKOMNIKA Telecommunication, Computing, Electronics and Control. 14(3):1099-1104, 2016.
45. Y. Jiang, S. Yin. Heterogenous-view Occluded Expression Data Recognition Based on Cycle-Consistent Adversarial Network and K-SVD Dictionary Learning Under Intelligent Cooperative Robot Environment. Computer Science and Information Systems, vol. 20, no. 4, 2023. https://doi.org/10.2298/CSIS221228034J.
46. Yin S, Li H, Sun Y, et al. Data Visualization Analysis Based on Explainable Artificial Intelligence: A Survey[J]. IJLAI Transactions on Science and Engineering, 2024, 2(2): 13-20.
47. Abdulla H H H A, Awad W S. Text Classification of English News Articles using Graph Mining Techniques[C]//ICAART (3). 2022: 926-937.
48. Fkih F, Alsuhaibani M, Rhouma D, et al. Novel Machine Learning-Based Approach for Arabic Text Classification Using Stylistic and Semantic Features[J]. Computers, Materials & Continua, 2023, 75(3).
49. Guo J, Zhao B, Liu H, et al. Supervised contrastive learning with term weighting for improving Chinese text classification[J]. Tsinghua Science and Technology, 2022, 28(1): 59-68.

Tianying Wen is with the Department of Education, Liaoning National Normal College. She had published several papers related to her major. Research direction: education analysis, data analysis, artificial intelligence.

Received: March 14, 2024; Accepted: June 21, 2024.