Research Article

Fusing Logical Relationship Information of Text in Neural Network for Text Classification

Heyong Wang and **Dehang Zeng**

Department of E-Business, South China University of Technology, Guangzhou, China

Correspondence should be addressed to Heyong Wang; wanghey@scut.edu.cn

With the development of computer science and information science, text classification technology has been greatly developed and its application scenarios have been widened. In traditional process of text classification, the existing method will lose much logical relationship information of text. The logical relationship information of a text refers to the relationship information among different logical parts of the text, such as title, abstract, and body. When human beings are reading, they will take title as an important part to remind the central idea of the article, abstract as a brief summary of the content of the article, and body as a detailed description of the article. In most of the text classification studies, researchers concern more about the relationship among words (word frequency, semantics, etc.) and neglect the logical relationship information of text. It will lose information about the relationship among different parts (title, body, etc.) and have an influence on the performance of text classification. Therefore, we propose a text classification algorithm—fusing the logical relationship information of text in neural network (FLRIOTINN), which complements the logical relationship information into text classification algorithms. Experiments show that the effect of FLRIOTINN is better than the conventional backpropagation neural networks which does not consider the logical relationship information of text.

1. Introduction

In recent years, with the development of science and technology, natural language processing (NLP) has been greatly developed due to its extensive application scenarios [1–4]. Text classification is an important branch of NLP. The application fields of text classification are extremely wide, such as opinion mining [5], social emotion detection [6], and educational knowledge recognition [7].

With the development of application scenarios, higher requirements are put forward for text classification. In order to improve the effect of text classification, many researchers adopt the way of fusing different information in text classification. While the logical relationship information of text is often overlooked, fusing logical relationship information of text can supplement the information lost and can be an effective way to improve the performance of text classification.

From the perspective of the complexity of data structure, text can be divided into two forms: one is text without structure (usually without obvious titles, abstracts, etc.), such as commodity reviews, and the other is text with obvious structure (usually with obvious titles, abstracts, etc.), such as patents, news, and papers. The most typical object of text without structure is short text [8–10]. Text with an obvious structure, due to its clear structure, often appears in formal written expressions and has a wide range of applications. Many researchers have studied the text with obvious structure: D'hondt et al. [11] emphasize the effect of phrases (versus single word) in patent and news classification by incorporating the text's statistical phrases and linguistic phrases into the word bag model. Hu et al. [12] focus on the improvement of patent keyword extraction algorithm by using distributed Skip-gram model and propose a new text classification keyword extraction method to improve the effect of the text classification algorithm. Patent is a kind of

text with obvious structure. But D'hondt and Hu only focus on the role of words in text classification. In other words, they focus more on the relationship among words of text. They did not take advantage of the logical relationship information in patent.

However, the logical relationship information of text is very important. The logical relationship information of text refers to the way the whole article is written. The concepts of logical relationship information of text include the relationship among various parts of the article. When reading, humans interpret different parts of the text in different ways, such as understanding the general content of the text through the title and abstract and understanding the details of the text through the body information. That is, the title provides the core idea of the text, the abstract describes the brief summary of the text, and the body provides the details of the text. The relationship of title, abstract, and body is the logical relationship of text. Human uses the logical relationship information of text to better interpret textual content. And also, it is beneficial for text classification to integrate different logical relationship information because it can better simulate the mode of human thinking.

In this regard, we improve the BP neural network and propose a text classification algorithm (fusing logical relationship information of text in neural network, FLRIO-TINN), which processes title and body, respectively, and supplements the logical relationship information that is generally neglected in text classification. Comparing the effect of text classification of FLRIOTINN with the conventional backpropagation neural network, experiments show that the performance of FLRIOTINN is better.

The main contribution of our paper is as follows: (1) our paper provides a good processing flow of text classification. We combine the characteristics of LDA model and neural network classifier to improve the classification performance. (2) Our paper provides a new perspective of text classification. We have pioneered the fusion of logical relationship information into a neural network for text classification. (3) Our paper verifies the validity and importance of taking the human thinking mode as a reference in the field of deep learning and artificial intelligence, which is conducive to further research on deep learning and artificial intelligence.

The rest of the paper is organized as follows: the second section will discuss current studies on text classification and research related to logical relationship information, the third section will introduce the processing framework in this paper, the fourth section describes the experiments of this article, and the fifth section summarizes the work.

2. Related Work

2.1. Text Classification. The major process of text classification includes preprocessing, text representation, feature selection, weighting, and classification. Among them, text representation, feature selection, weighting, and classification algorithms become the focus of text classification research.

2.1.1. Text Representation. There are three mainstream research directions in text representation. The first is based on bag of word (BOW). BOW can trace back to Harris's 1954 article on "Distributional structure" [13] and widely used in text classification. Researchers make a lot of optimization on the basis of BOW. Sarker and Gonzalez [14] add additional information (change phrases, Syn-set expansion, Unified Medical Language System (UMLS) semantic types, and concept IDs) related to the text classification application to the original text representation which is based on BOW. It expands the space of text representation and enriches the information. Although BOW plays an important role in text representation, text representation based on BOW cannot cover all the text information, such as semantic information.

The second research direction is based on Word2Vec. In the formulation of word vectors, each word is represented by a vector which is concatenated or averaged with other word vectors in a context, and the resulting vector is used to predict other words in the context [15]. The most famous representation method based on word vector is Word2Vec. Word2Vec was created and published in 2013 by a team of researchers led by Tomas Mikolov at Google. It is a text representation method that uses text semantics to represent words as a vector form. Based on this, Zhang et al. [16] make good use of Word2Vec to improve the accuracy of classification.

The third research direction is based on network. Network is a recent advancement of text representation. Written texts can be modeled as networks in several ways [17]. One possibility is to map texts into a word adjacency network (WAN), which links adjacent words [18, 19]. Amancio et al. [20] study the symmetry of word adjacency networks and find that specific authors prefer particular types of symmetric motifs. Based on this, the researchers carry out a text classification of authorship of books in a data set comprising books written by 8 authors. The performance of the classification of authorship of books is pretty good. de Arruda [21] use adjacency network to text representation and propose a network model which can describe the local topological/dynamical properties of function words. The results show that the accuracy can achieve up to 95%, which is much higher than similar networked approaches. There are also other kinds of networks which are based on other characters of text. Akimushkin et al. [22] propose a methodology based on the dynamics of word co-occurrence networks representing written texts to model small chunk of text to grasp stylistic feature. With an optimized supervised learning procedure based on a nonlinear transformation performed by Isomap, 71 out of 80 texts are correctly classified using the K-nearest neighbor algorithm.

2.1.2. Feature Selection and Weighting. Feature selection is a significant step in text classification used to reduce the computational cost and improve the performance of classification. Wang et al. [23] focus on a utility-based feature selection method and measure the usefulness of terms from the point of expressing the author's main ideas. With the further study of latent Dirichlet allocation (LDA), many

researchers notice its useful in feature selection. Li et al. [24] use LDA topic for feature selection to solve the problem of semantic similarity measurement in SVM and enhance the performance in text classification.

In terms of weighting, there are more relevant studies. Ren and Sohrab [25] implement a new inverse class space density frequency (ICS delta F) for the traditional TF-IDF, generating the TF-IDF-ICS delta F method. It shows that the TF-IDF-ICS delta F weighting approach is promising over the compared well-known baseline term weighting approach. Yang et al. [26] improve TF-IDF by judging whether words appeared in title, abstracts, and keyword. If a word appears in the title, abstract, and keyword, indicating that the word is more important, the word is given more weight. Sun et al. [27] transform the title, abstract, keyword, and the text of the paper into four parts that are title, abstract, keyword, and general feature. Assign a different weight to each part. It shows that the average precision is enhanced. Li et al. [28] use the weight estimation algorithm which fuses the information of the depth and breadth of Wikipedia network for web page classification, and it also gets good results.

2.1.3. Classification Algorithm.

The research on classification algorithm can be divided into two kinds of research. One is to make an improvement of traditional algorithms, such as SVM, Naïve Bayes, and KNN. Xiang et al. [29] take the title and abstract of the patent text as two instances, respectively, use undirected graph and SVM to predict the unmarked patent text. Experiments show that the proposed method has better performance compared to traditional SVM and KNN. Jiang et al. [30] improve Naïve Bayes by a new way, called deep feature weighting (DFW), which estimates the conditional probabilities of naive Bayes by deeply computing feature weighted frequencies from training data. They integrate the additional frequency information into the naive Bayes algorithm, which has achieved the purpose of improving the quality of the model. Pang et al. [31] combine KNN- and centroid-based classifiers to overcome the problems of the sensitivity of the algorithm to imbalanced class distributions and irrelevant or noisy-term features in KNN.

The other kind of research is based on neural network. Zheng and Zheng [32] apply Word2Vec to generate word vectors automatically and a bidirectional recurrent structure to capture contextual information and long-term dependence of sentences. Jin also makes use of a convolutional neural network to capture the key component in texts. Ren and Deng [33] propose a multistream network based on recurrent neural network. The multistream network mainly consists of the basal stream, which retains the original sequence information, and background knowledge-based streams, which are composed of keywords and co-occurred words.

2.2. Logical Relationship Information.

The structure of the text can be divided into titles, abstracts, and body. Every part has a unique role in expressing the idea of text. Readers can get an overview of the text by reading titles and abstracts and understand the details by reading body. It is a common way for people to access information. This way of understanding, which is based on logical relationship information, is also extremely important for text classification.

Fusion of logical relationship information is beneficial to improve the performance of text classification. Some researchers have made corresponding comparisons. Langlois et al. [34] study whether to use full text or to use partial text such as title, abstract, and keyword in text classification. Langlois compares decision trees, SVM, KNN, and Naive Bayes with each other. It shows that the classification effect of the full text is not significantly better than using part of text information. Mai et al. [35] compare the full text with title. They use multilayer perceptron (MLP), convolutional neural network (CNN), and recurrent neural network (RNN) as classifier. It proves that the performance of text classification by using title can be as good as using full text. These researchers only use the structure information as a method of segmenting text, and studies show that using part of the text can obtain good classification results as using full text. This proves that there is important information in different logical parts (title, abstract, body, etc.), and it is helpful for text classification.

2.3. Summary.

In the research of text representation, the text representations based on BOW, word vector, and network are more likely to deal with the quantitative relationship among words. The relationship among different logical parts (title, abstract, etc.) cannot be fused and well handled.

In the research of feature selection and weighting, most researchers focus on the improvement of existing methods of feature selection and weight, such as TF-IDF. Except these researchers, some of the researchers realize the importance of logical relationship information. They did not further supplement the text logical relationship information into the text classification algorithm. If the classification algorithm is not adjusted accordingly, the logical relationship information cannot act on the text classification algorithm. The logical relationship information is not used effectively, and this is an inadequate application of logical relationship information. It is difficult to maximize the role of logical relationship information.

In the research of classification algorithm, the studies based on traditional algorithm mostly focus on improving the algorithm by fusing extra information, and the studies based on neural network mostly focus on improving the algorithm by changing the structure of the network from the point of view of algorithm. No researcher correlates the changes in neural network structure with the fused information.

Text logical relationship information is important and helpful for text classification. If we handle the different logical parts (title, abstract, etc.), respectively, in classification algorithm, the different processing in the algorithm can be thought as the different interpretation in human mind. It is a better way to simulate the mode of human thinking.

Based on this idea, a neural network with strong advantages in information fusion for processing different logical parts, respectively, is used. We propose fusing text logical relationship information of text in neural network (FLRIOTINN), which processes title and body, respectively. The experiment shows that the classification accuracy of FLRIOTINN is higher than conventional backpropagation neural network in which text logical relationship information is not used.

3. Methodology

3.1. General Processing Flow.
The general processing flow is given in Figure 1.

There are three stages.

In the first stage, the data sets are preprocessed in order to get more effective data for the final classification performance. The data sets in the form of text are obtained from the online public data set and segmented by white space, which is the most common character between English words. After the segmentation, the article in the data sets is in the form of the bag of words. Then, the stop words in the segmented data are removed. The removed stop words include the common stop words [36] (such as "a" and "an") and user-defined stop words. After stop words removal, the preprocessed data are obtained.

For example, there is an article with the title of "my house" and the body of "I have a TV and a computer in my house." After word segmentation and stop words removal, the outcome of the title will be {"my," "house"} and the outcome of the body will be {"I," "have," "TV," "computer," "my," "house"}.

In the second stage, the preprocessed data are represented in the form of a word frequency vector.

LDA and deduplication are used to get the text feature dictionary. As shown in Figure 2, the preprocessed data are classified according to class labels. And articles which are in the same category are treated as a collection (D_i) to extract LDA topic words. After the LDA topic words corresponding to each category are obtained, the topic words in different categories are gathered together and deduplicated to obtain the text feature dictionary (Dic, with the length of d).

And then, the preprocessed data are vectorized through the text feature dictionary (Dic). The title and body of the preprocessed data are separated. Each article is divided into three parts and is decomposed into the form of "title + body + class label." Then, the text feature dictionary (Dic) is used to vectorize the body and title by word frequency, and the class label is vectorized by "one-hot".

For instance, suppose the text feature dictionary is (my, house, have, TV, computer), and there are four categories; the preprocessed text mentioned above is vectorized as follows.

The title collection ({"my," "house"}) is represented as the title vector $(1, 1, 0, 0, 0)$. The body collection ({"I," "have," "TV," "computer," "my," "house"}) is represented as the body vector $(1, 1, 1, 1, 1)$. If the article belongs to the second category and there are four categories, the vectorized class label is $(0, 1, 0, 0)$.

Through all this, the vectorized data of every article are obtained.

In the third stage, neural networks are trained. The vectorized data are input into the proposed FLRIOTINN for training.

3.2. Neural Network Structure

3.2.1. Conventional Backpropagation Neural Network for Text Classification.
The conventional backpropagation neural network for text classification (hereinafter referred to as CBPNNTC) is widely used. The CBPNNTC [37] always takes the feature vector of the full text as the input and has the similar structure as the ordinary backpropagation neural network [38]. The structure [39] of the conventional backpropagation neural network for text classification is shown in Figure 3.

As shown in Figure 3, d denotes the length of the text feature dictionary (Dic), c is the total number of categories, W denotes the weight of the network, b denotes the bias of the network, and F denotes the activation function.

In the CBPNNTC, depending on the performance of network, there could be two or more layers. Each layer has several hidden layers except the input layer.

In CBPNNTC, the input layer usually contains a frequency vector (I_0) of the full text. The frequency vector (I_0) of the full text is obtained by combining the vector of the title, body, and the other parts of the text (if the text contains the other parts).

For example, the text with the title frequency vector $(0, 1, 1, 0, 1, 1, 0)$ and body frequency vector $(1, 2, 3, 1, 1, 2, 4)$ will be represented as $I_0 = (0+1, 1+2, 1+3, 0+1, 1+1, 1+2, 0+4) = (1, 3, 4, 1, 2, 3, 4)$. The CBPNNTC takes $(1, 3, 4, 1, 2, 3, 4)$ as input.

In the other layers, there are many hidden layers. The specific calculation in the hidden layer is as follows:

$$I_{n+1} = F_{n+1}(W_n * I_n + b_n). \qquad (1)$$

F_{n+1} denotes the activation function of the $n + 1^{\text{th}}$ hidden layer. W_n is the weight of the $n + 1^{\text{th}}$ hidden layer. b_n is the bias of the $n + 1^{\text{th}}$ hidden layer.

Several hidden layers make up a single layer:

$$
\begin{aligned}
I_{n+1} &= F_{n+1}(W_n * I_n + b_n), \\
I_{n+2} &= F_{n+2}(W_{n+1} * I_{n+1} + b_{n+1}), \\
I_{n+x+1} &= F_{n+x+1}(W_{n+x} * I_{n+x} + b_{n+x}).
\end{aligned}
\qquad (2)
$$

I_n is the input of the layer. I_{n+x+1} is the output of the layer. The layer consists of x hidden layers. The structure of second layer is the common structure of the subsequent layers.

3.2.2. FLRIOTINN.
In the CBPNNTC, it cannot identify the logical relationship information in the text because the different logical parts (title and body) are merged in I_0. The neural network cannot obtain the word frequency information of different logical parts from I_0.
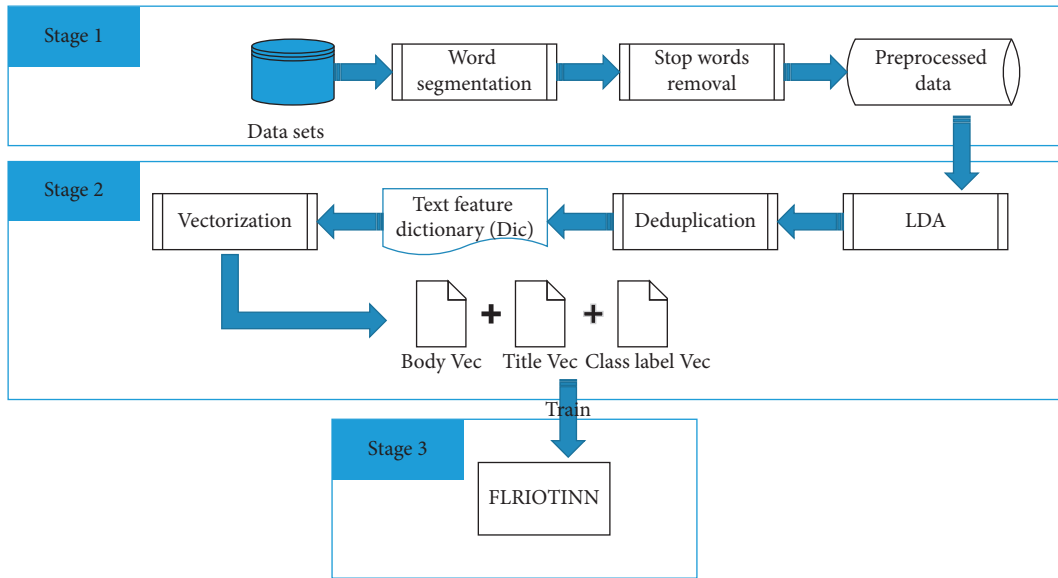
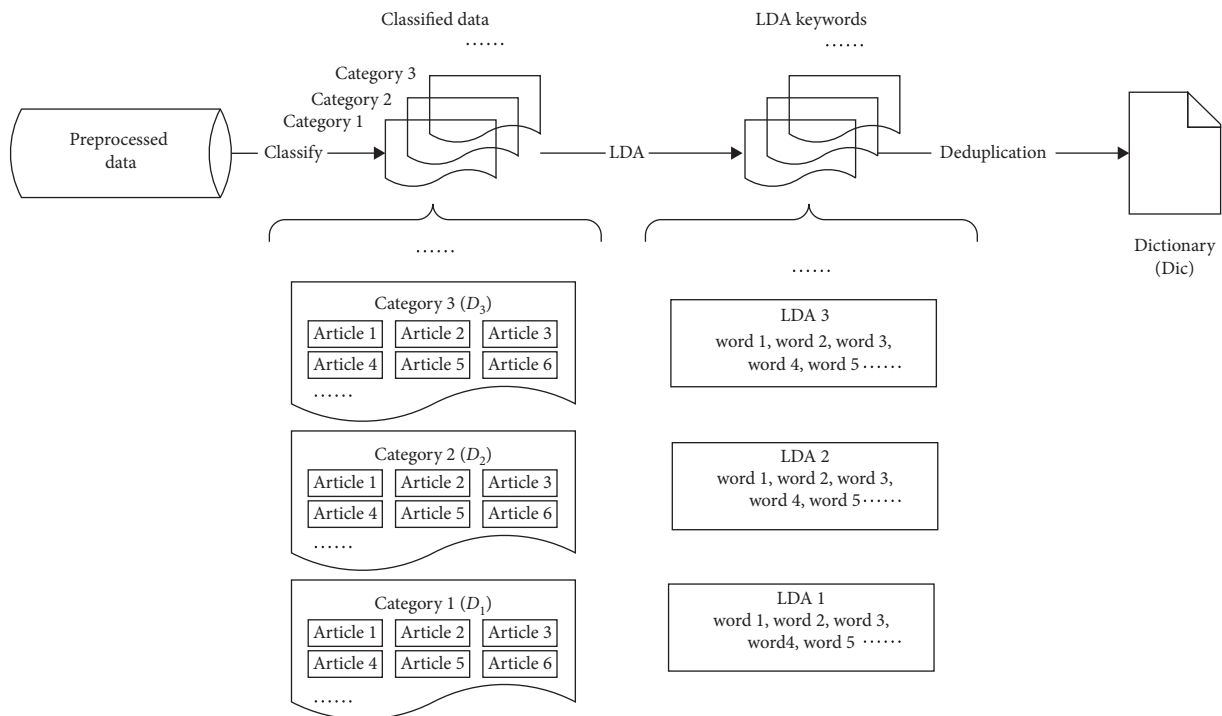FIGURE 1: General processing flow.

FIGURE 2: LDA dimension reduction process.

The number in I_0 indicates the total number of times the corresponding words in Dic appear in the text. But in the CBPNNTC, the neural network cannot identify how many times the word appears in the title and how many times the word appears in the body or in other logical parts. When we cannot distinguish different logical parts, we cannot fuse the relationship information.

Based on this, FLRIOTINN is proposed, and it take title vector, body vector, and other logical parts vector as input separately in order to fix how many times the word appears in the title and how many times the word appears in the body or in other logical parts.

FLRIOTINN inputs different logic structure vectors into different sublayers for processing.

For example, the text with the title frequency vector $(0, 1, 1, 0, 1, 1, 0)$ and body frequency vector $(1, 2, 3, 1, 1, 2, 4)$ will input into FLRIOTINN directly. The title frequency vector is input into the first sublayer first layer of FLSINN and the body frequency vector is input into the second sublayer of first layer of FLSINN. The weights and bias in
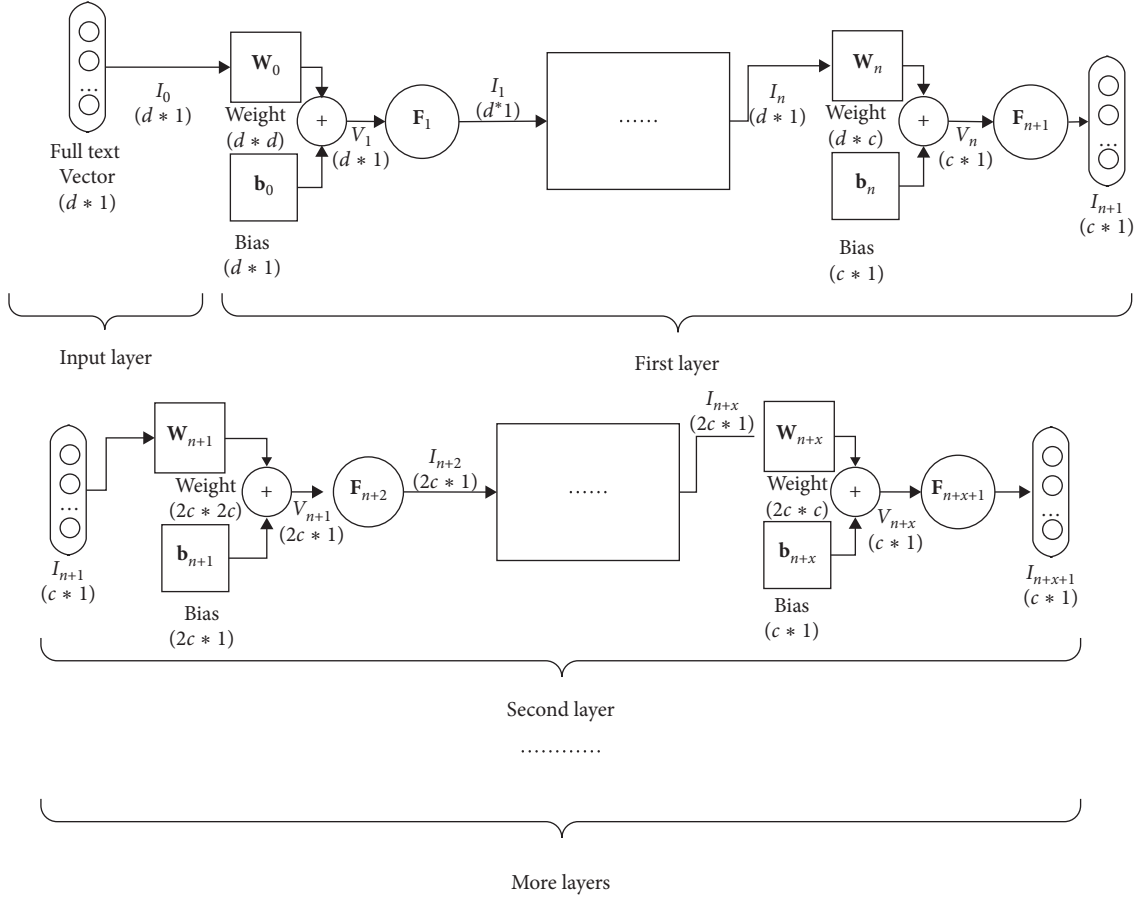
FIGURE 3: Conventional backpropagation neural network for text classification. d: length of Dic; c: total number of categories; **w**: weight; **b**: bias; **F**: activation function.

different sublayers are trained according to the characteristics of different logical parts. The training process can be seen as the formation of different thinking for different logical parts. After the process of first layer, the different information extracted from different logical parts should combined for classification.

The specific structure of FLRIOTINN is shown in Figure 4.

In the FLRIOTINN, layer 0 is the input layer, which inputs the information of different logical parts (title, body, etc.), respectively. Depending on the number of logical parts, there could be more than two inputs:

$$T_0 = \text{Title Vec},$$
$$B_0 = \text{Body Vec}. \tag{3}$$

Title Vec and Body Vec are the title frequency vector and body frequency vector.

The first layer is the logical information processing layer, which is divided into two or more sublayers corresponding to the input. The structure in the red rectangle in Figure 4 is regarded as a sublayer. The first layer is used to process different logical parts (title, body, etc.) and extracts the features (T_{n+1}, B_{n+1}, etc.) of different logical parts.

Each sublayer has the same structure. There are several hidden layers in each sublayer. The specific calculation of a sublayer is as follows (take the first sublayer as example):

$$T_1 = F_1 (W_{11} * T_0 + b_{11}),$$
$$T_2 = F_2 (W_{12} * T_1 + b_{12}), \tag{4}$$
$$T_n = F_n (W_{1n} * T_{n-1} + b_{1n}).$$

T_0 is the input of the sublayer. T_i, B_i $(i = 1, 2, \ldots, n)$ is the output of the hidden layer in each sublayer. T_n is also the output of the sublayer. F_n denotes the activation function of the nth hidden layer of every sublayer. W_{1n} denotes the weight of the nth hidden layer of the first sublayer. b_{1n} denotes the bias of the nth hidden layer of the first sublayer.

In the second layer, the input (I_0) of the second layer is generated by connecting T_n B_n and other logical parts of processed vectors.

For example, if $T_n = \{t_{n1}, t_{n2}, t_{n3}, \ldots, t_{ni}\}$, $B_n = \{b_{n1}, b_{n2}, b_{n3}, \ldots, b_{ni}\}$, and there are no other logical parts processed by the network, then $I_0 = \text{connect}(T_n, B_n) = \{t_{n1}, t_{n2}, t_{n3}, \ldots, t_{ni}, b_{n1}, b_{n2}, b_{n3}, \ldots, b_{ni}\}$.

In this way, I_0 contains much more information comparing to the input of CBPNNoTC. The specific calculation of the second layer is as follows:
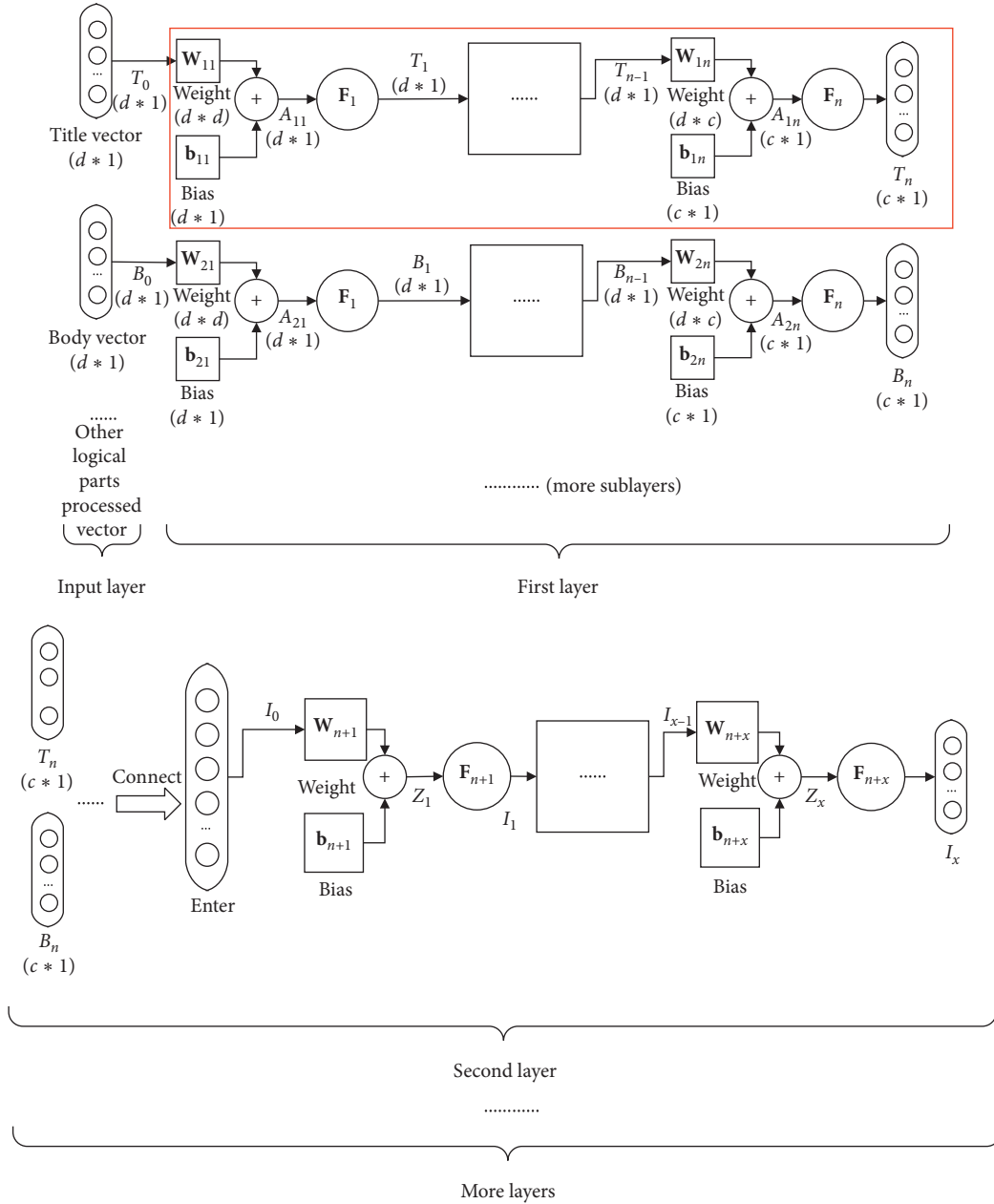
FIGURE 4: FLRIOTINN. d: length of Dic; c: total number of categories; w: weight; b: bias; F: activation function.

$$I_0 = \text{connect}(T_n, B_n, \ldots),$$
$$I_1 = F_{n+1}(W_{n+1} * I_0 + b_{n+1}),$$
$$I_2 = F_{n+2}(W_{n+2} * I_1 + b_{n+2}), \qquad (5)$$
$$I_x = F_{n+x}(W_{n+x} * I_{x-1} + b_{n+x}).$$

$T_n, B_n, \ldots$ are the inputs of the second layer. "..." denotes the vector of the other logical parts, if there are other logical parts except title and body. I_x is the output of the second layer. There are x hidden layers in the layer.

There may be more layers in the network, but the structure of the layer after the second layer is similar to the structure of the second layer. The only difference between the second layer and the subsequent layers is the input. The subsequent layers only take one vector, which is the output of the previous layer, as input. There is no connection operation. For instance, the third layer takes I_x as input, and I_x do not need to connect with other vectors.

The core of FLRIOTINN is the processing of the first layer. Unlike only one processing flow in each layer in the conventional backpropagation neural network, two or more processing sublayers are set in the first layer of the FLRIOTINN. Different logical parts of text are processed by different sublayers. The weights in the two or more sublayers are completely independent, and more effective weights for the final topic are trained. Through this structure, the different logical parts can be processed, respectively, and the information of different logical parts can be interpreted properly. It can better simulate the mode of human thinking, which is to interpret different logical parts in different ways.

Compared with other classification algorithms, FLRIO-TINN is very convenient for the expansion of different logical parts in the first layer (sublayers can be added directly to the existing neural network structure). The subjective weighting of human beings is avoided. The weights are completely trained by the neural network, and the relationship information of text logical parts is effectively utilized.

4. Experiment

4.1. Model

4.1.1. FLRIOTINN. FLRIOTINN is shown in Figure 5.

The FLRIOTINN conducted in the experiment applies 3 layers: input layer, the first layer, and the second layer.

In the input layer, the Title Vec and the Body Vec, which are the outputs of stage 2, will input into network as T_0 and B_0.

In the first layer, there are two sublayers and three hidden layers in each sublayer. The first sublayer is used to process the title vector of the article, and the second sublayer is used to process the body vector of the article. And the linear and nonlinear activation functions are used to make the information obtained by the neural network larger. After the process of the three hidden layers, the first layer outputs two vectors (T_3, B_3), representing the information extracted by the neural network from the title and the body, respectively.

In the second layer, T_3 and B_3 are connected to get the input (I_0) of the second layer. And then, I_0 will be processed by two hidden layers to get the output (OUT).

The specific calculation is as follows:

First layer

Hidden layer 1 in the first layer:

$$
\begin{aligned}
A_{11} &= W_{11} * T_0 + b_{11}, \\
T_1 &= F_1 (A_{11}), \\
A_{21} &= W_{21} * B_0 + b_{21}, \\
B_1 &= F_1 (A_{21}).
\end{aligned}
\tag{6}
$$

F_1 is a linear activation function, $F_1 a = a$.

Hidden layer 2 in the first layer:

$$
\begin{aligned}
A_{12} &= W_{12} * T_1 + b_{12}, \\
T_2 &= F_2 (A_{12}), \\
A_{22} &= W_{22} * B_1 + b_{22}, \\
B_2 &= F_2 (A_{22}).
\end{aligned}
\tag{7}
$$

F_2 is a sigmoid activation function, $F_2 a = \text{sigmoid} a = (1/1 + e^{-a})$.

Hidden layer 3 in the first layer:

$$
\begin{aligned}
A_{13} &= W_{13} * T_2 + b_{13}, \\
T_3 &= F_3 (A_{13}), \\
A_{23} &= W_{23} * B_2 + b_{23}, \\
B_3 &= F_3 (A_{23}).
\end{aligned}
\tag{8}
$$

F_3 is a sigmoid activation function, $F_3 a = \text{sigmoid} (a)$.

In summary, the calculation of the first layer can be expressed as

$$
\begin{aligned}
T_3 &= F_3 (W_{13} * F_2 (W_{12} * F_1 (W_{11} * T_0 + b_{11}) + b_{12}) + b_{13}), \\
B_3 &= F_3 (W_{23} * F_2 (W_{22} * F_1 (W_{21} * C_0 + b_{21}) + b_{22}) + b_{23}).
\end{aligned}
\tag{9}
$$

These activation functions include both the linear change (F_1) of the original information and the nonlinear change (F_2, F_3), which make the information extracted from the first layer as rich as possible and make the eigenvectors $(T_3 \text{ and } B_3)$ T_3 and B_3 contain as much basic information as possible.

Second layer

T_3 and B_3 obtained are $c * 1$-dimensional vectors, and c denotes the number of categories. That is to say, the information of title and body is compressed through the processing of the first layer. In order to get the final result, the information of title and body is merged. The input vector I_0 is obtained by connecting T_3 and B_3. At this time, I_0 is a vector of $2c * 1$ dimension, which is input into the second layer. In order to ensure the richness of information, the activation function in the second layer is set as a combination of linear (F_4) and nonlinear (F_5). The specific calculation is as follows.

Hidden layer 1 in the second layer:

$$
\begin{aligned}
Z_1 &= W_3 * I_0 + b_3, \\
I_1 &= F_4 (Z_1).
\end{aligned}
\tag{10}
$$

F_4 is a linear activation function, $F_4 a = a$.

Output layer in second layer:

$$
\begin{aligned}
Z_2 &= W_4 * I_1 + b_4, \\
\text{OUT} &= F_5 (Z_2).
\end{aligned}
\tag{11}
$$

F_5 is a softmax function, $F_5 (\mathbf{a}) = \text{soft max} (\mathbf{a})$.

The second layer of computation can be expressed as

$$
\text{OUT} = F_5 (W_4 * F_4 (W_3 * I_0 + b_3) + b_4).
\tag{12}
$$

OUT represents the probability distribution of the classification instance.

4.1.2. Conventional Backpropagation Neural Network. In order to verify whether the processing method of distinguishing logical parts is effective, the conventional backpropagation neural network is taken as a comparison (hereinafter referred to as CBPNN). As shown in Figure 6, the processing flow of CBPNN is designed, which applies 3 layers and has the similar structure with FLRIOTINN for comparison.

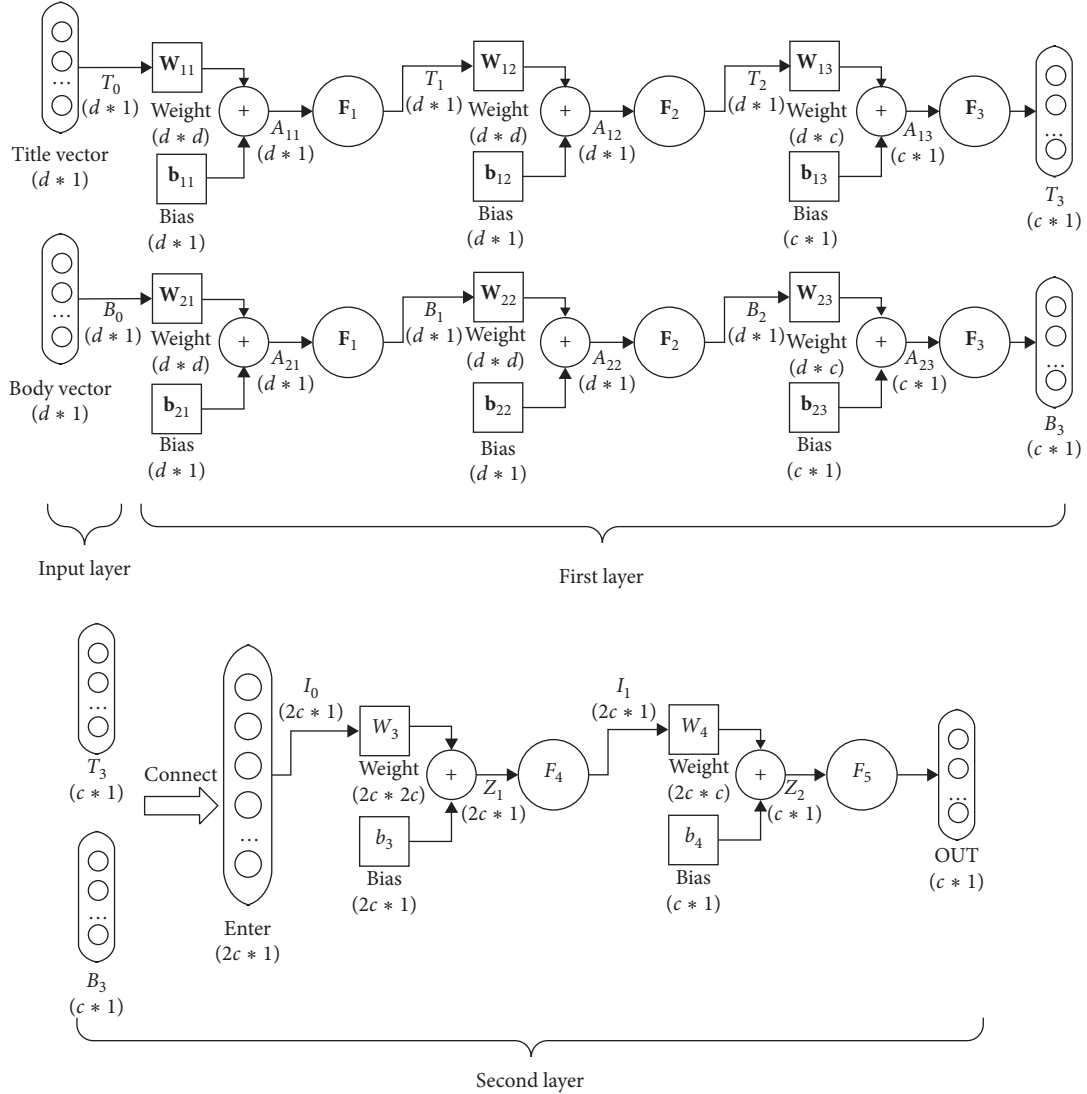As can be seen from Figure 6, the input of the conventional neural network will add the title $(T_0 = \text{TitleVec})$

FIGURE 5: FLRIOTINN. d: length of Dic; c: total number of categories; **w**: weight; b: bias; **F**: activation function.

and the body ($B_0 = \text{BodyVec}$) to get a word frequency vector of the whole document for input (I_0).

The input information of the text is processed uniformly. The different logical parts of the text are not identified. That is to say, the logical relationship information is not fused into the neural network.

The specific calculation is as follows.

Input layer:

$$I_0 = C_0 + T_0. \tag{13}$$

First layer:

$$I_3 = F_3 \left(W_2 * F_2 \left(W_1 * F_1 \left(W_0 * I_0 + b_0 \right) + b_1 \right) + b_2 \right). \tag{14}$$

F_1 is a linear activation function, $F_1 a = a$.

F_2 is a sigmoid activation function, $F_2 a = \text{sigmoid}(a)$.

F_3 is a sigmoid activation function, $F_3 a = \text{sigmoid}(a)$.

Second layer:

$$\text{OUT} = F_5 \left(W_4 * F_4 \left(W_3 * I_3 + b_3 \right) + b_4 \right). \tag{15}$$

F_4 is a linear activation function, $F_4 a = a$.

F_5 is a softmax function, $F_5 a = \text{soft max}(a)$.

OUT represents the probability distribution of the classification strength.

4.2. Data Set. Three text classification data sets (20-news-group, Reuters 21578, and Reddit self-posts) are selected from public data sets. All the data in 20-news-group, part of data in Reuters 21578, and Reddit self-posts are selected for experiment. The specific categories are shown in Table 1.

As shown in Table 1, all 20 categories of data in 20-news-group are selected, with 1000 documents in each category, totaling 20,000 documents. 20-news-group is a data set donated in 1999-09-09. It contains 20 categories of news and is widely used for text classification.
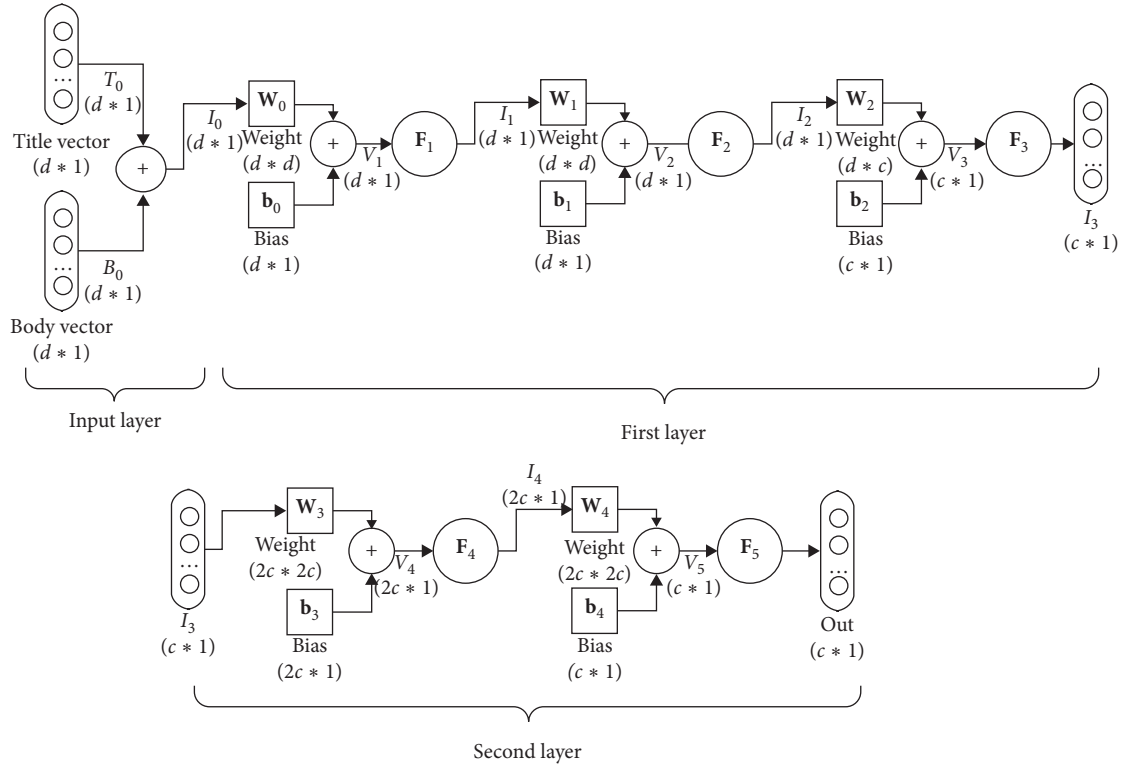
FIGURE 6: Conventional backpropagation neural network. d: length of Dic; c: total number of categories; **w**: weight; b: bias; **F**: activation function.

For Reuters 21578 data set, it contains 21,578 news from Reuters newswire in 1987. Since 1991, it appeared as Reuters 22173 and was assembled and indexed with 135 categories by personnel from Reuters Ltd. in 1996. We selected four categories (crude, grain, trade, and interest) for the experiment, totaling 2042 documents.

For the Reddit self-posts, the aim of the data set was to create an interesting, large text classification problem with many classes. The data consists of 1.013M self-posts, posted from 1013 subreddits (1000 examples per class). In the Reddit self-posts data set, five categories of data are selected from the first-level label: company/website, animals, arts, books, and advice/question. 5000 documents are selected from each category, totaling 25,000 documents.

All the documents of the data set mentioned above have the clear logical parts of title and body.

4.3. Performance Evaluation.

Accuracy, precision, recall, F1-measure, ROC, micro-ROC, and macro-ROC curves are used for evaluation.

ROC is often used to evaluate the effect of binary classification. In this paper, a ROC curve is made for each category, and AUC (area under curve) is calculated to evaluate the classification effect of the classifier in each category. Micro-ROC and macro-ROC are used for the general evaluation of classifiers. Assuming that there are m classification texts and n categories, the predicted matrix is $m * n$. The micro-ROC splices the above matrix backward

according to the categories to get the matrix of $m * n * 1$ and then draws a ROC curve based on it. The specific transformation is as Figure 7. The macro-ROC makes the ROC curve for each category, respectively, and then takes the average.

4.4. Results

4.4.1. Overall Performance of Models.

In order to evaluate the overall performance of the model, this section evaluates the model by the accuracy, micro-ROC, and macro-ROC, which are relatively comprehensive indicators

LDA is an important dimension reduction method, in which the number of topics generated by each category of text is the most important parameter. In order to get a better classification effect, we control the number of topics produced by LDA and carry out several experiments on three data sets using two networks, respectively. The average of accuracy is shown in Table 2.

In terms of accuracy, the proposed model (FLRIOTINN) is superior to CBPNN in three data sets. As can be seen from Table 2, the number of LDA topics does have an impact on the accuracy of text classification, and the accuracy of different data sets will fluctuate under different LDA topic numbers. Based on this, in Section 4.4.2, where the performance of models in each category is listed, experiment results with high accuracy are showed.

The AUC (area under curve) of micro-Roc and macro-Roc of two models of different LDA topic numbers for

Table 1: Data set.

Data set	Category	Total
20-news-group	alt.atheism, comp.graphics comp.os.mswindows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware, comp.windows.x misc.forsale, rec.autos, rec.motorcycles rec.sport.baseball, rec.sport.hockey sci.crypt, sci.electronics, sci.med, sci.space, soc.religion. Christian, talk.politics.guns talk.politics.mideast, talk.politics.misc, talk.religion.misc	20000
Reuters 21578	Crude, grain, trade, interest	2042
Reddit self-posts	Company/website, animals arts, books, advice/question	25000

	Category 1	Category 2	······	Category n
Text 1	a_{11}	a_{12}	...	a_{1n}
Text 2	a_{21}	a_{22}	...	a_{2n}
Text 3	a_{31}	a_{32}	...	a_{3n}
······	...	...	...	...
Text m	a_{m1}	a_{m2}	...	a_{mn}

	Category
Text 1	a_{11}
Text 2	a_{21}
Text 3	a_{31}
... ...	...
Text m	a_{m1}
Text 1	a_{12}
Text 2	a_{22}
... ...	...
Text m	a_{m2}
... ...	...
Text 1	a_{1n}
... ...	...
Text m	a_{mn}
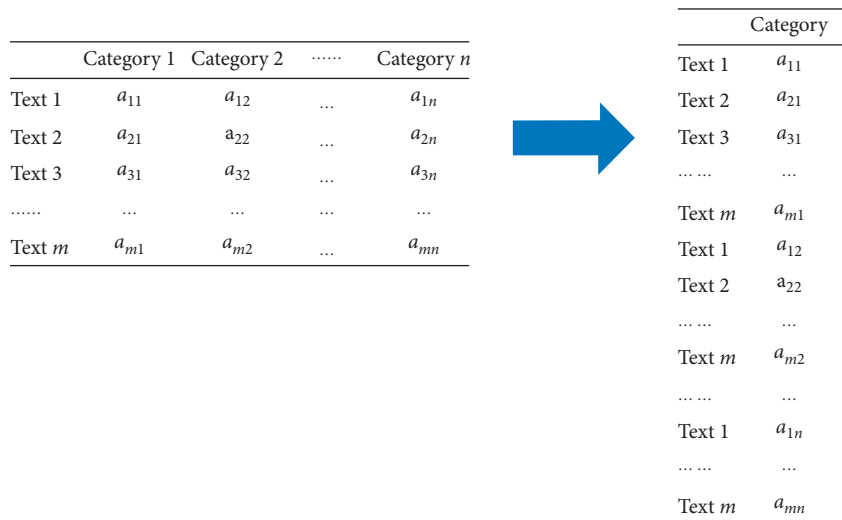
Figure 7: Micro-ROC transformation.

Table 2: The average of accuracy.

Data set	Topic num	FLRIOTINN	CBPNN
20-news-group	30	0.7549	0.7210
	40	0.7680	0.7200
	50	0.7604	0.7189
Reuters 21578	30	0.9264	0.5571
	40	0.9215	0.5179
	50	0.9199	0.8774
Reddit self-posts	30	0.8798	0.8569
	40	0.8882	0.8635
	50	0.8844	0.8762

different data sets are shown in Figures 8 and 9 (abscissa naming rule is datasets_LDA topic number):

It can be seen that, in the performance of micro-Roc and macro-Roc, the overall performance of FLRIOTINN is better than that of CBPNN.

4.4.2. Performance of Models in Each Category. In order to find out the performance of each model in different
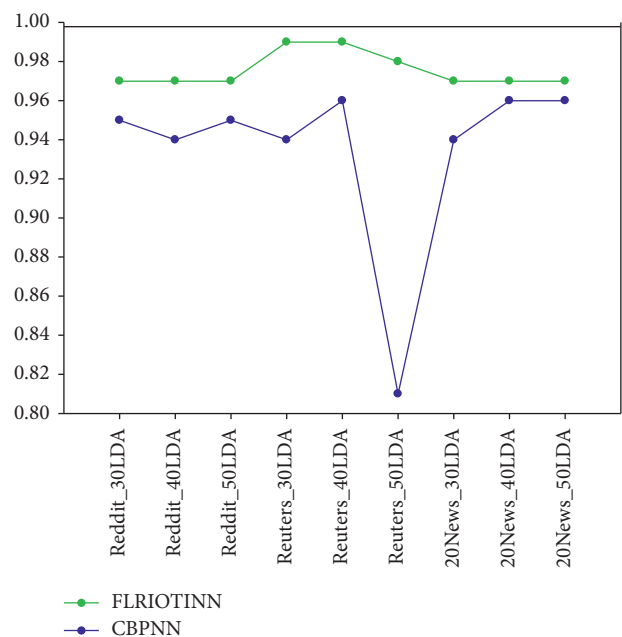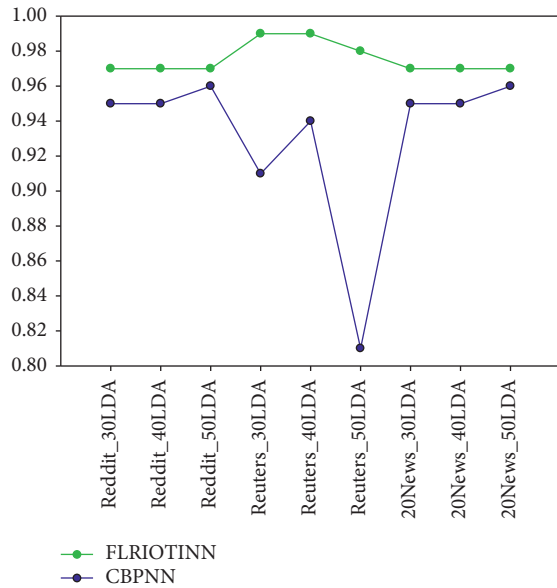

Figure 8: Micro-Roc.

Figure 9: Macro-Roc.

categories, this section measures the performance of models in each category by four evaluation indicators: precision, recall, F1-measure, and ROC. Abscissa is the categories' name. The vertical coordinate of the blue point represents the value of the index of CBPNN in the corresponding category. The green point represents the FLRIOTINN.

In 20-news-group, when the topic number is 40, FLRIOTINN achieves the highest accuracy and CBPNN is similar to its highest accuracy. Select the experiment of LDA topic number 40 in the 20-news-group, compare precision, recall, F1-measure, and ROC in each category, as shown in Figures 10–13.

As shown in the figures, FLRIOTINN performs better than CBPNN in most of the categories in terms of precision, recall, F1-measure, and ROC. Especially in ROC and F1-measure, FLRIOTINN is better than CBPNN in almost all classes

In Reuters 21578, when the topic number is 30, FLRIOTINN achieves the highest accuracy and have the distinct difference comparing to the experiment of FLRIOTINN with other topic numbers. Select the experiment of LDA topic number 30 in Reuters 21578, and compare precision, recall, F1-measure, and ROC curves in each category, as shown in Figures 14–17 (the missing value represents a divisor of 0).

As can be seen from the figures, FLRIOTINN performs better in precision, recall, F1-measure, and ROC than CBPNN in most classes and FLRIOTINN shows less difference among different classes.

In Reddit self-posts, FLRIOTINN achieves the highest accuracy when the topic number is 40. But it is very close when the topic number is 50, and when the topic number is 50, CBPNN achieves its highest accuracy. Consider all of this, select the experiment of LDA topic number 50 in Reddit self-posts and compare precision, recall, F1-measure, and ROC curves in each category, as shown in Figures 18–21.
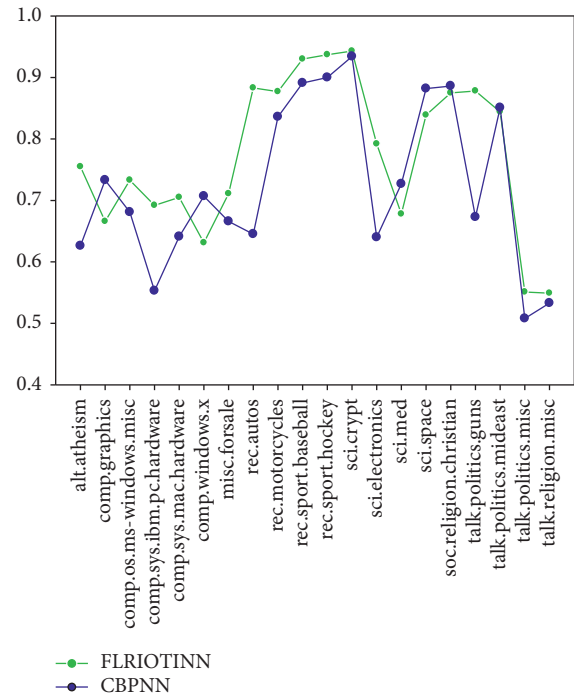

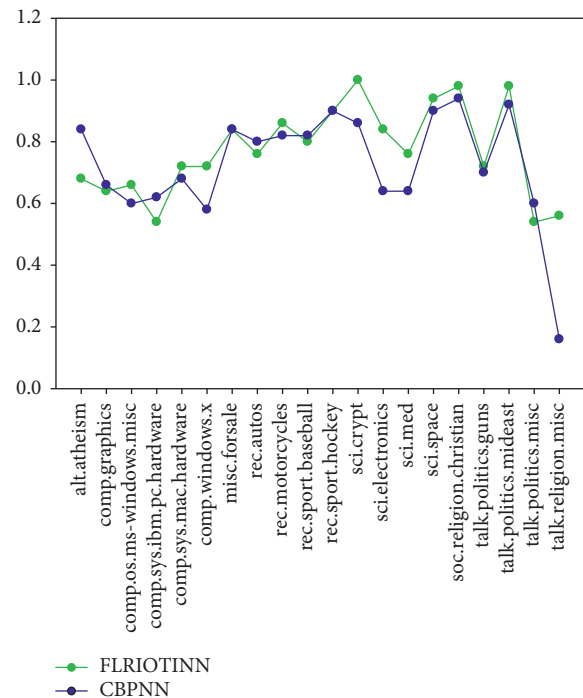Figure 10: Precision in 20-news-group.

Figure 11: Recall in 20-news-group.

Overall, among precision, recall, F1-measure, and ROC, the performance of FLRIOTINN is better than that of CBPNN. FLRIOTINN has higher precision, recall, F1-measure, and ROC in most classes.

4.5. Summary. From the experiments, we can see that the proposed model (FLRIOTINN) in this paper performs better
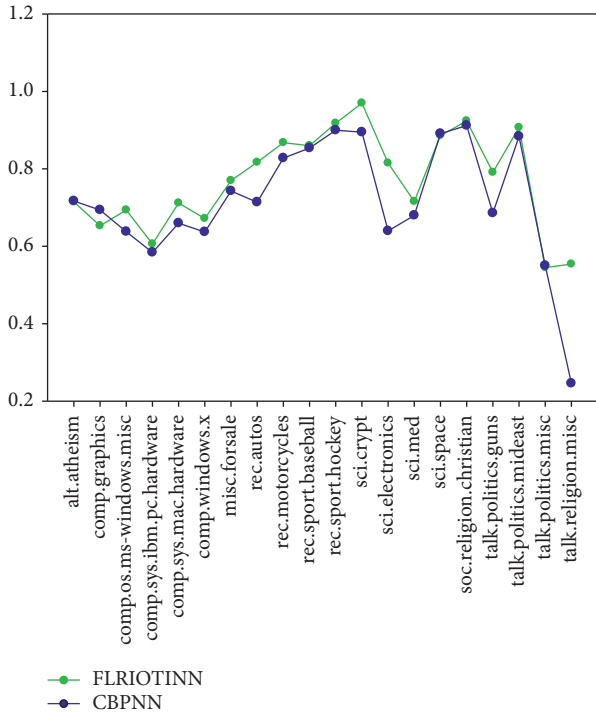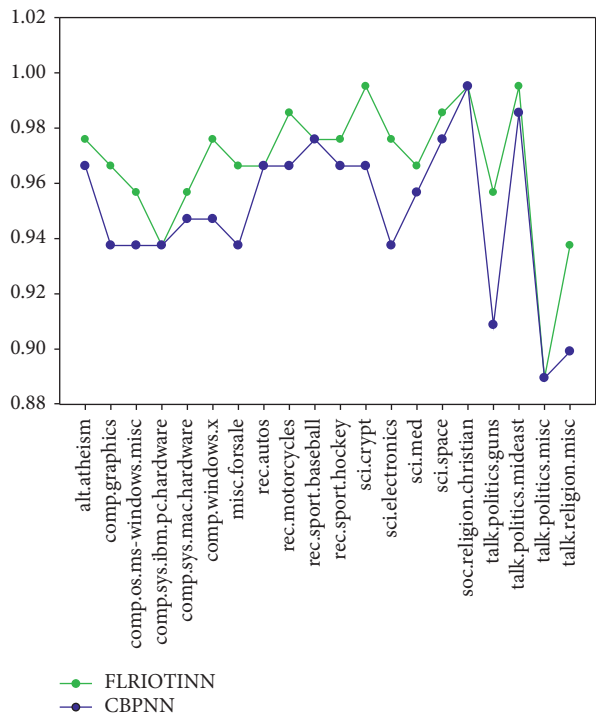
Figure 12: F1-measure in 20-news-group.
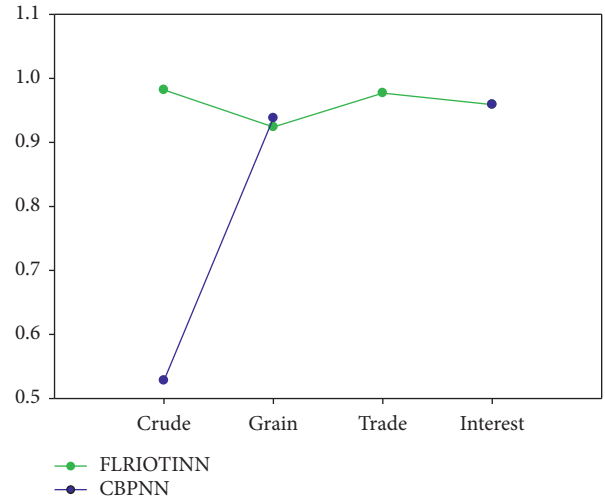

Figure 13: ROC in 20-news-group.

Figure 14: Precision in Reuters 21578.

Figure 15: Recall in Reuters 21578.
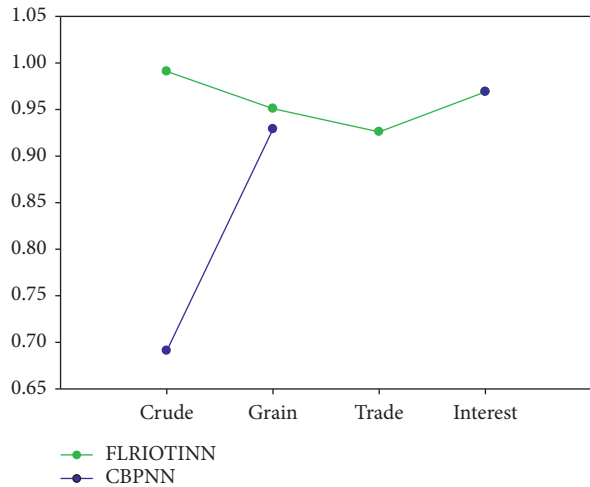
Figure 16: F1-measure in Reuters 21578.

in various evaluation indicators and different data sets than the conventional backpropagation neural network, which does not distinguish the logical parts. This shows that the idea of fusing logical relationship information in text classification is correct and effective.
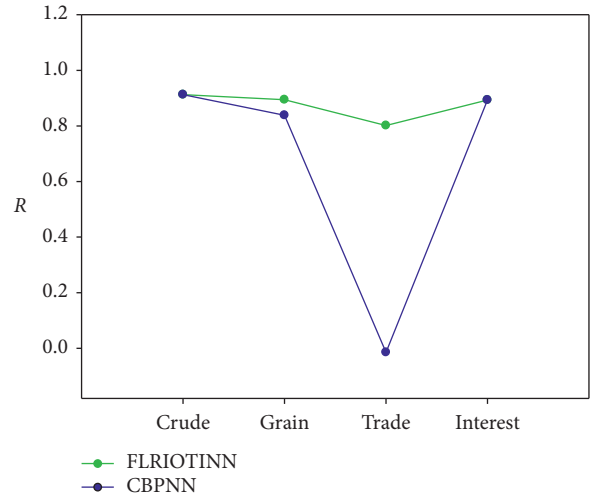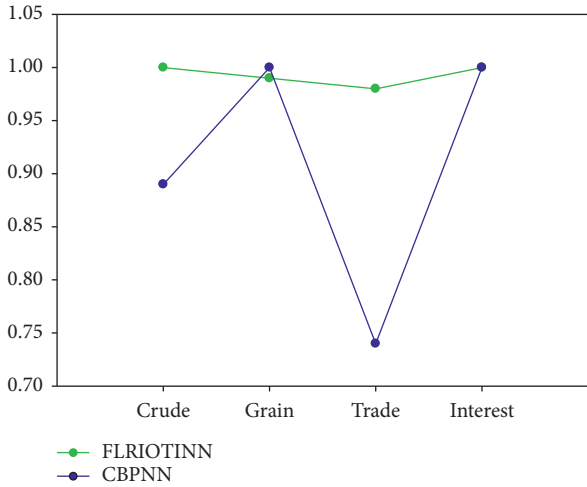
Figure 17: ROC in Reuters 21578.
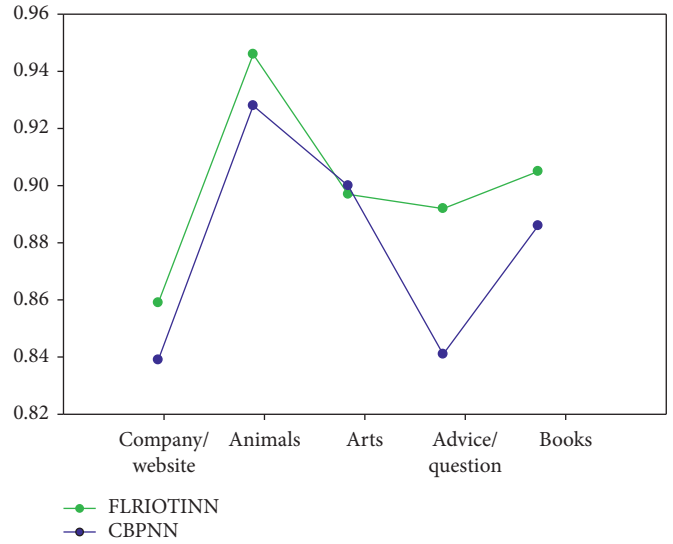

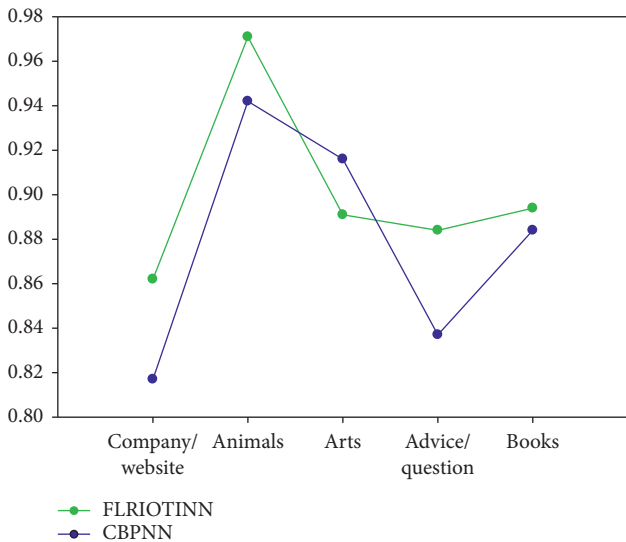Figure 18: Precision in Reddit self-posts.
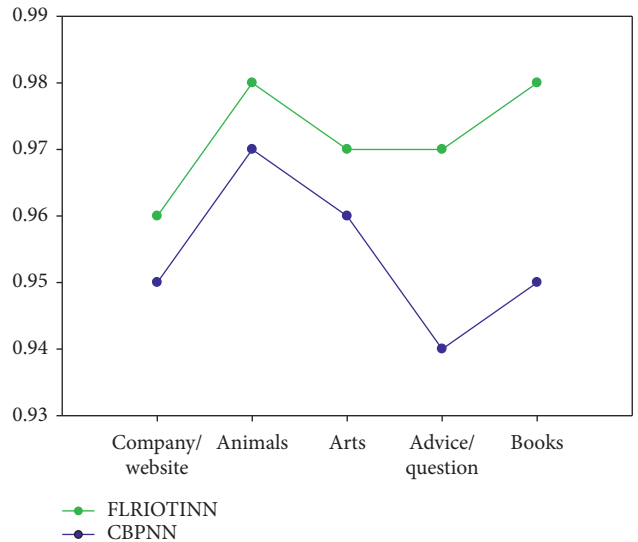

Figure 19: Recall in Reddit self-posts.

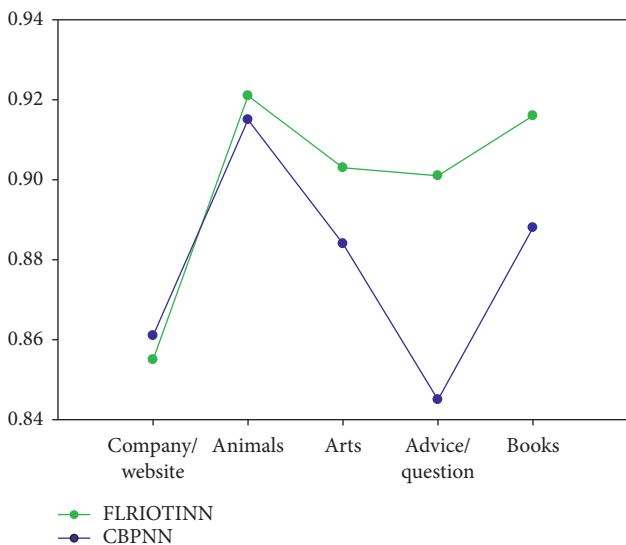Figure 20: F1-measure in Reddit self-posts.

Figure 21: ROC in Reddit self-posts.

5. Conclusion

In this paper, we propose a new way to deal with text classification. We use LDA to reduce the dimension effectively and then we distinguish the different logical parts of text and input them into different sublayers of the neural network for text classification. Experiments show that the experimental process designed in this paper can improve the accuracy of text classification. Compared with the conventional backpropagation neural network algorithm which does not fuse text logical relationship information, FLRIOTINN has better performance in three data sets (20-news-group, Reuters 21578, and Reddit self-posts). We prove that the logical relationship information is important and helpful for text classification, and the separate processing of different logical parts of text (title, body, etc.) in text classification helps to mine the information of text and fully understand and process the information.

In this paper, a neural network classification algorithm, which fuses text logical relationship information by processing the title and body, respectively, is designed to simulate the mode of human thinking which is interpreting different logical parts of text in different ways. This paper emphasizes the importance of text logical relationship information in text classification and the simulation of the mode of human thinking and combines the two ideas to form the text classification model proposed in this paper. The experiments confirm that it is effective to simulate the mode of human thinking by processing different logical parts of the text differently.

Data Availability

The Reddit self-posts data used to support the findings of this study have been deposited in the Kaggle repository. The Reuters 21578 data used to support the findings of this study have been deposited in the UCI repository. The 20-news-group data used to support the findings of this study have been deposited in the http://qwone.com/~jason/20Newsgroups/. The Reddit self-posts, Reuters 21578, and 20-news-group are collected from public data sets and can be accessed by following URL: Reddit self-posts (Kaggle): https://www.kaggle.com/mswarbrickjones/reddit-selfposts; Reuters 21578 (UCI): https://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection; 20-news-group: http://qwone.com/~jason/20Newsgroups/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

References

[1] E. Soysal, J. Wang, M. Jiang et al., "Clamp-a toolkit for efficiently building customized clinical natural language processing pipelines," *Journal of the American Medical Informatics Association*, vol. 25, no. 3, pp. 331–336, 2018.

[2] A. Park, A. L. Hartzler, J. Huh, D. W. McDonald, and W. Pratt, "Automatically detecting failures in natural language processing tools for online community text," *Journal of Medical Internet Research*, vol. 17, no. 8, p. e212, 2015.

[3] H. Wang and M. Hong, "Distance variance score: an efficient feature selection method in text classification," *Mathematical Problems in Engineering*, vol. 2015, Article ID 695720, 10 pages, 2015.

[4] J. S. Zhang and N. M. El-Gohary, "Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking," *Journal of Computing in Civil Engineering*, vol. 30, no. 2, Article ID 04015014, 2016.

[5] S. Sun, C. Luo, and J. Chen, "A review of natural language processing techniques for opinion mining systems," *Information Fusion*, vol. 36, pp. 10–25, 2017.

[6] Y. Rao, H. Xie, J. Li, F. Jin, F. L. Wang, and Q. Li, "Social emotion classification of short text via topic-level maximum entropy model," *Information & Management*, vol. 53, no. 8, pp. 978–986, 2016.

[7] T. Horakova, M. Houska, and L. Domeova, "Classification of the educational texts styles with the methods of artificial intelligence," *Journal of Baltic Science Education*, vol. 16, no. 3, pp. 324–336, 2017.

[8] L. Yan, Y. Zheng, and J. Cao, "Few-shot learning for short text classification," *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 29799–29810, 2018.

[9] D.-T. Vo and C.-Y. Ock, "Learning to classify short text from scientific documents using topic models with various types of knowledge," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1684–1698, 2015.

[10] L. Gao, S. Zhou, and J. Guan, "Effectively classifying short texts by structured sparse representation with dictionary filtering," *Information Sciences*, vol. 323, pp. 130–142, 2015.

[11] E. D'hondt, S. Verberne, C. Koster, and L. Boves, "Text representations for patent classification," *Computational Linguistics*, vol. 39, no. 3, pp. 755–775, 2013.

[12] J. Hu, S. B. Li, and G. C. Yang, "Patent keyword extraction algorithm based on distributed representation for patent classification," *Entropy*, vol. 20, no. 2, pp. 1–19, 2018.

[13] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[14] A. Sarker and G. Gonzalez, "Portable automatic text classification for adverse drug reaction detection via multi-corpus training," *Journal of Biomedical Informatics*, vol. 53, pp. 196–207, 2015.

[15] H. Zhang and G. Zhong, "Improving short text classification by learning vector representations of both words and hidden topics," *Knowledge-Based Systems*, vol. 102, pp. 76–86, 2016.

[16] D. Zhang, H. Xu, Z. Su, and Y. Xu, "Chinese comments sentiment classification based on word2vec and SVM$^{\text{perf}}$," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1857–1863, 2015.

[17] J. Cong and H. Liu, "Approaching human language with complex networks," *Physics of Life Reviews*, vol. 11, no. 4, pp. 598–618, 2014.

[18] D. R. Amancio, O. N. Oliveira Jr., and L. D. F. Costa, "Unveiling the relationship between complex networks metrics and word senses," *EPL Europhysics Letters*, vol. 98, no. 1, pp. 1–6, 2012.

[19] D. R. Amancio, "Probing the topological properties of complex networks modeling short written texts," *PLoS One*, vol. 10, no. 2, pp. 1–17, 2015.

[20] D. R. Amancio, F. N. Silva, and L. D. F. Costa, "Concentric network symmetry grasps authors' styles in word adjacency networks," *EPL Europhysics Letters*, vol. 110, no. 6, pp. 1–6, 2015.

[21] H. F. de Arruda, D. R. Amancio, and L. D. Costa, "Classifying informative and imaginative prose using complex networks," *EPL Europhysics Letters*, vol. 113, no. 2, pp. 1–11, 2016.

[22] C. Akimushkin, O. N. Oliveira, and D. R. Amancio, "Text authorship identified using the dynamics of word co-occurrence networks," *PLoS One*, vol. 12, no. 1, pp. 1–15, 2017.

[23] H. Wang, M. Hong, and R. Y. K. Lau, "Utility-based feature selection for text classification," *Knowledge and Information Systems*, vol. 61, no. 1, pp. 197–226, 2019.

[24] F.-G. Li, Y. Liang, and X.-Z. Gao, "Research on text categorization based on LDA-wSVM model," *Application Research of Computers*, vol. 32, no. 1, pp. 21–25, 2015.

[25] F. Ren and M. G. Sohrab, "Class-indexing-based term weighting for automatic text classification," *Information Sciences*, vol. 236, pp. 109–125, 2013.

[26] H. Yang, H. Cui, and H. Tang, "A text classification algorithm based on feature weighting," *Green Energy and Sustainable Development I*, vol. 1864, pp. 1–8, 2017.

[27] H. Sun, L. Yang, and D. Wu, "A multi-label classification approach based on ontology and structure weight strategy," *ICIC Express Letters*, vol. 7, no. 6, pp. 1055–1059, 2013.

[28] H. Li, Z. Xu, T. Li, G. Sun, and K.-K. Raymond Choo, "An optimized approach for massive web page classification using entity similarity based on semantic network," *Future Generation Computer Systems*, vol. 76, pp. 510–518, 2017.

[29] B. Xiang, G. Liu, and G. Yang, "Patent text classification method based on multi-instance learning," *Information Studies: Theory & Application*, vol. 41, no. 11, pp. 144–148, 2018.

[30] L. Jiang, C. Li, S. Wang, and L. Zhang, "Deep feature weighting for naive Bayes and its application to text classification," *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 26–39, 2016.

[31] G. Pang, H. Jin, and S. Jiang, "CenKNN: a scalable and effective text classifier," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 593–625, 2015.

[32] J. Zheng and L. Zheng, "A hybrid bidirectional recurrent convolutional neural network attention-based model for text classification," *IEEE Access*, vol. 7, pp. 106673–106685, 2019.

[33] F. Ren and J. Deng, "Background knowledge based multi-stream neural network for text classification," *Applied Sciences-Basel*, vol. 8, no. 12, pp. 1–18, 2018.

[34] A. Langlois, J.-Y. Nie, J. Thomas, Q. N. Hong, and P. Pluye, "Discriminating between empirical studies and nonempirical works using automated text classification," *Research Synthesis Methods*, vol. 9, no. 4, pp. 587–601, 2018.

[35] F. Mai, L. Galke, and A. Scherp, "Using deep learning for title-based semantic subject indexing to reach competitive performance to full-text," in *Proceedings of the 18th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, Fort Worth, TX, USA, June 2018.

[36] J.-B. Lee, T Lee, and H. P. In, "Automatic stop word generation for mining software artifact using topic model with pointwise mutual information," *IEICE Transactions on Information and Systems*, vol. E102.D, no. 9, pp. 1761–1772, 2019.

[37] W. Zhang, X. Tang, and T. Yoshida, "TESC: an approach to text classification using semi-supervised clustering," *Knowledge-Based Systems*, vol. 75, pp. 152–160, 2015.

[38] X. Yuanyou, X. Yanming, and Z. Ruigeng, "An engineering geology evaluation method based on an artificial neural network and its application," *Engineering Geology*, vol. 47, no. 1, pp. 149–156, 1997.

[39] N. M. Ranjan and R. S. Prasad, "Automatic text classification using BPLion-neural network and semantic word processing," *The Imaging Science Journal*, vol. 66, no. 2, pp. 69–83, 2018.