Hindawi
Mobile Information Systems
Volume 2022, Article ID 7191312, 14 pages
https://doi.org/10.1155/2022/7191312

Research Article

Research on Security Model Design Based on Computational Network and Natural Language Processing

Junpu Yang

Information Center, Liaoning Provincial Party School of CPC, Shenyang 110004, Liaoning, China

Correspondence should be addressed to Junpu Yang; 1764200067@e.gzhu.edu.cn

Received 10 May 2022; Revised 12 July 2022; Accepted 1 August 2022; Published 31 August 2022

Academic Editor: Imran Shafique Ansari

Human logical thinking exists in the form of language, and most of the knowledge is also recorded and transmitted in the form of language. It is also an important and even core part of artificial intelligence. Communicating with computers in natural language is a long-standing pursuit of people. People can use the computer in the language they are most accustomed to and can also use it to learn more about human language abilities and intelligent mechanisms. The realization of natural language communication between humans and computers means that computers can not only understand the meaning of natural language texts but also express the intentions and thoughts given in natural language texts. This paper designs and studies a computational model for natural language processing (NLP) models for natural language processing. This paper aims to study the design of computing network security model based on natural language processing. This paper proposes three calculation models, which are based on the long-term and short-term memory neural network model (LSTM), FastText model, and text processing model (GCN) based on graph convolution neural network. Several natural language processing models are evaluated and analyzed using four indexes: accuracy, recall, exactness, and *F*1 vaule. Results show that the performance level of the GCN model is the best. The accuracy of the NLP recognition of this model reaches 86.66%, which is 2.93% and 1.55% higher than the accuracy of the LSTM model and the FastText model, respectively.

1. Introduction

1.1. Background. With the development of computer technology, especially artificial intelligence, artificial intelligence algorithms represented by machine learning and deep learning technology have made rapid progress in the fields of image processing and text classification, and they have been widely used in various fields of people's life. However, the research on the security of computer networks is not sufficient. Especially in the era of big data, people need to invest a lot of time in organizing and sorting the massive digital information. Natural language processing technology provides a good support for this. This technology is widely used in medical text classification and image recognition, semantic recognition and judgment, building information classification, intelligent system instruction recognition, and other fields. Moreover, scholars in different fields have also compiled many large-scale and informative real dictionary corpora and pay more attention to the extraction of meaningful information. However, most of the existing natural language processing methods do not combine the latest research results of artificial intelligence and deep learning. The original methods of constructing conflict dictionary and machine learning are not only costly but also complex and redundant.

Natural language processing includes natural language understanding and natural language generation. Achieving natural language communication between humans and machines means enabling computers to not only understand the meaning of natural language texts but also to express given intentions, thoughts, etc., in natural language texts.

1.2. Related Work. NLP has been widely used in various fields, and many scholars have also conducted research on its application in various fields. The research of Nobel et al.

describes the preprocessing and processing steps and highlights the important challenges that must be overcome to successfully implement free text mining algorithms using NLP tools and machine learning in small language fields. According to the eighth TNM classification system, based on tumor size, presence, and involvement items, a rule-based algorithm was constructed [1]. However, his algorithm ignores the removal of redundant text. Hence, the classification result may be mediocre. Lou et al. developed an algorithm that uses NLP technology and machine learning models to automatically detect free-text radiology reports with follow-up recommendations. The dataset used in his research is composed of 6000 free-text reports from the author's institution. On this dataset, he trained the Naive Bayes, decision tree, and maximum entropy model, and the results show that the score of the decision tree model is better than that of the other two [2]. However, his results may be because of omissions or delays in related texts. Tom et al. reviewed the important deep learning-related models and methods used in a large number of NLP tasks and provided their evolution process. It also summarizes and compares various models and puts forward a detailed understanding of the past, present, and future of deep learning in NLP [3]. The data presentation of the research results is not very clear. Using the method introduced in our recent work, Brooke et al. obtained information about six styles from a large number of texts of the Gutenberg project. Thus, he built a high coverage and fine-grained dictionary, including common multiword collocations. Using this information and the annotations to the two Modernist Texts, Brooke J confirmed that free indirect discourse does reflect the mixture of narrative and direct speech at the stylistic level. [4]. Compared with commonly used NLP models, his research methods are more complicated. The automatic extraction of keywords is an important research direction in text mining, NLP, and information retrieval. In this regard, Onan et al. research tested five statistical keyword extraction methods (keyword extraction based on the most frequent measurement, keyword extraction based on word frequency-inverse sentence frequency, keyword extraction based on co-occurrence statistics, and eccentricity keyword extraction and TextRank algorithm) and conducted a comparative analysis on the predictive performance of scientific text document classification algorithms and ensemble methods [5]. His research is quite informative for this article, however, it still needs to be simplified. Jung and Lee compare and analyze a method of using the building information model (BIM) to automatically classify the building information model (BIM) cases in construction projects and deploy natural language processing (NLP) and common unsupervised text classification learning [6]. The model he studied can make semantic prediction, and it can also provide new ideas for this article.

2. Related Methods of NLP Calculation Model Design Research

2.1. Feature Engineering of NLP. Data preprocessing refers to some processing performed on the data before the main processing. The irregularly distributed measurement network is converted into a regular network through interpolation to facilitate computer operations. Feature engineering is data preprocessing. The original language data contains a lot of noise information and meaningless data. Text preprocessing is the process of obtaining meaningful value from the text dataset. It generally includes several steps of word division, word embedding, feature extraction, and classification [7]. In layman's terms, feature engineering is to convert the language that users can read into information that the computer can understand. The degree of refinement of feature engineering processing determines the upper limit of the performance of the algorithm model. The text classifier is a process of summarizing the information that the computer can understand into a more concrete, reusable, and transferable knowledge base. Researchers need to adjust the model parameters to continuously approach this performance limit [8].

2.1.1. Word Segmentation and Word Embedding Model. In the English context, spaces are generally used as semantic gaps for word segmentation. The general preprocessing process includes two steps: text word segmentation and stop word removal [9]. Next, we will introduce the word embedding algorithm-word vector algorithm (Word2Vec). The word vector model is based on the assumption that the similarity between words is measured by whether their adjacent words are acquainted, which is based on the principle of "distance similarity" in linguistics.

Word2Vec is a typical word embedding model based on the distribution hypothesis, i.e., words with similar contexts have similar meanings. This model can get the distributed representation of words, which mainly includes two structures: CBOW and Skip-Gram. The structure diagram is shown in Figure 1. The former is to predict the middle word through the vocabulary before and after the middle word in the input and output layer, and the latter is to predict its context vocabulary through the input middle word [10].

2.1.2. CBOW Word Vector Update Process. The final output of the CBOW model is the predicted middle word. Assuming that the size of the word vector in the training dataset is A, the number of hidden layer neurons is M, and the words in the given input context are vectors encoded by One-Hot. It is characterized by $\{y1, y2, y3, \ldots, yA\}$, and the weight between the input layer and the hidden layer is represented by a matrix W with a dimension of A·M [11]. For the hidden layer, the following is satisfied:

$$h = y^T W = A_{w_I}^T. \quad (1)$$

The hidden layer refers to the layers other than the input layer and the output layer in the multilevel feedforward neural network. The hidden layer does not directly receive external signals, nor does it directly send signals to the outside world. It is only required when the data is nonlinearly separated. The weights of the hidden layer and the output layer are represented by a matrix W' of dimension M·A, and the prediction score can be obtained by multiplying the hidden layer vector and the weight matrix.

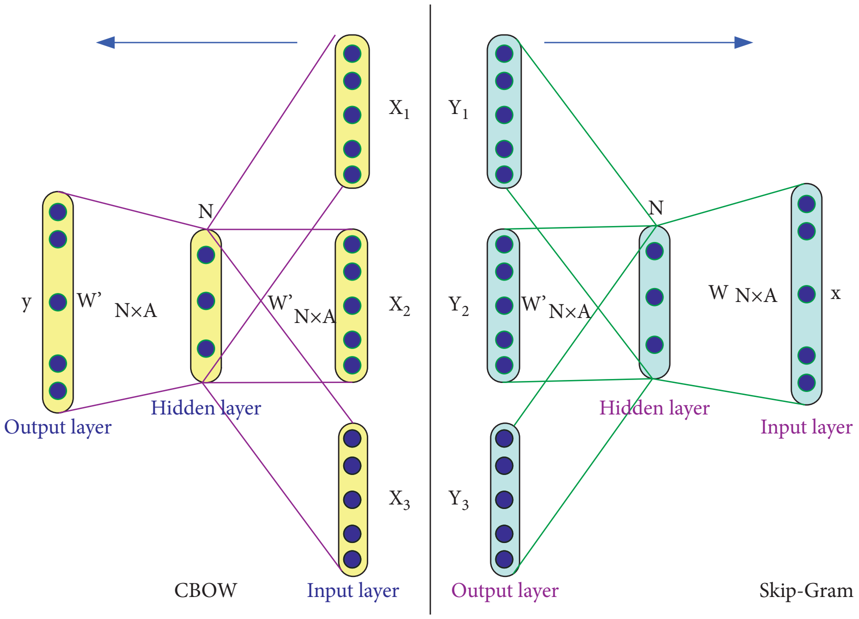


Figure 1: Word2Vec word embedding model.

$$s_j = A'^{T}_{w_j} \bullet h. \tag{2}$$

Backpropagation algorithm, referred to as BP algorithm, is a learning algorithm suitable for multilayer neuron networks, which is based on the gradient descent method. The output value of the neural network is followed by a layer of linear classifier to obtain the posterior probability of the word.

$$F\left(w_j | w_I\right) = x_j = \frac{e^{s_j}}{\sum_{q=1}^{A} e^q} = \frac{e^{A'_{w_O} A_{w_I}}}{\sum_{q=1}^{A} e^{A'_{w'_j} A_{w_I}}}. \tag{3}$$

In each training batch, dropout randomly drops some neurons (note that it is temporarily dropped), so that it does not work during forward propagation and does not update parameters during back propagation. In the next training batch, restore these neurons and repeat the process. The previous article is an introduction to forward propagation, and then back propagation is introduced. The training goal of the neural network here is to find the maximum value of the posterior probability of the output layer, i.e., given the document context information and hidden layer weight matrix, calculate the maximum value of the posterior probability of each word in the vocabulary, and finally, predict the target word. The above formula can be transformed as follows:

$$\begin{aligned} \text{Max}\left[F\left(w_j | w_I\right)\right] &= \text{Max}\left(x_{j*}\right) \\ &= \log\left[\text{Max}\left(x_{j*}\right)\right] \\ &= s_{j*} - \log\left(\sum_{q-1}^{A} e^s q = -P\right). \end{aligned} \tag{4}$$

A loss function or a cost function is a function that maps a random event or the value of its related random variable to a non-negative real number to represent the “risk” or “loss” of the random event. Defining the minimization target P as the loss function, and j^* as the index of the target word in the output layer. The neural network first propagates back from the output layer to the hidden layer, and the loss function is tricked to calculate the weight update formula.

$$\frac{\partial P}{\partial s_j} = x_j - r_j = e_j, \tag{5}$$

where r_j means that the j^{th} output of the output layer is assigned a value of 1 when the target word is output, and it is 0 in other cases. Continue to seek partial derivatives to get the following:

$$\frac{\partial P}{\partial w'_{ij}} = \frac{\partial P}{\partial s_j} \cdot \frac{\partial s_j}{\partial w'_{ij}} = e_j h_i. \tag{6}$$

Let ρ be the learning rate of the gradient descent method. h_i is the i^{th} neuron in the hidden layer. Use the stochastic gradient descent method to solve the following:

$$\begin{cases} w'_{ij} = w'_{ij} - \rho \bullet e_j \bullet h_i, \\ A'_{w_j} = A'_{w_j} - \rho \bullet e_j \bullet h_j. \end{cases} \tag{7}$$

Weight is a relative concept, which is for a certain indicator. The weight of an indicator refers to the relative importance of the indicator in the overall evaluation. Updating weights can be achieved by calculating the difference between the predicted value and expected value of the neural network.

$$\frac{\partial P}{\partial h_i} = \sum_{j=1}^{A} \frac{\partial P}{\partial s_i} \bullet \frac{\partial s_i}{\partial h_i} = K_i. \tag{8}$$

D K_i is an M-dimensional vector, which represents the sum of the word prediction error in the vocabulary and the product of the word vector of the output value, decomposing the output value of the hidden layer.

$$
\begin{aligned}
h &= y^{T} W = A_{w_I}^{T},\\
h_i &= \sum_{b=1}^{A} y_b \bullet w_{bI}.
\end{aligned}
\tag{9}
$$

Find the partial derivative of w_{bI} to get the following:

$$
\begin{aligned}
\frac{\partial P}{\partial w_{ki}} &= \frac{\partial P}{\partial h_i} = \frac{\partial h_i}{\partial w_{ki}} = K_i \bullet y_b,\\
\frac{\partial P}{\partial W} &= y \bullet K = y * K^{T}.
\end{aligned}
\tag{10}
$$

The update formula of the weight matrix is as follows:

$$
A_{w_I} = A_{w_I} - \rho \bullet K. \tag{11}
$$

After multiple iterations of training, the word vector update ends when the prediction error is equal to 0.

2.2. Optimization of the Operation Efficiency of the Word Embedding Model. In machine learning, especially deep learning, Softmax is a very common and important function, especially in multiclassification scenarios. In the above process, the number of Softmax calculation is very large as its calculation involves all the contents of all datasets in the dictionary. To reduce the computational complexity of the model, researchers and scholars have proposed several optimization schemes. Commonly used optimization schemes are the negative sampling method and the tomographic Softmax method [12]. When cleaning data to construct positive and negative samples, because of the delayed reporting of logs, when constructing samples in the problem of click events, the exposed unclicked data is often mistaken for negative samples. The core principle of the negative sampling method is to increase the prediction probability of positive samples, while reducing the prediction probability of negative samples. It is derived from the noise comparison estimation algorithm, which uses a random sample set to predict words outside the target (that is, the target negative sample) to improve the training speed of the model. This method only needs to calculate the probability of positive samples and several negative samples after each iteration, which will greatly reduce the calculation amount of the model [13]. Hierarchical Softmax uses Hoffman trees to optimize the model calculation process. The Hoffman tree has the shortest weighted path length. The hierarchical Softmax method uses the number of times each word appears in the corpus as the weight to construct a Hoffman tree. Frequent words are close to the root node and low-frequency words are far away from the root node [14].

2.3. Semantic Disambiguation Based on IFD. Word sense disambiguation is sometimes called word sense tagging, and its task is to determine the specific meaning of a polysemous word in a given context. Because of the ambiguity and complexity of natural language, there are many ways to describe the same meaning, and the computer will recognize it as different meanings. Therefore, IFD is used to eliminate the ambiguity of word segmentation [15]. Taking the example of "cup" to analyze its concept from different source information. The concept of "cup" in the IFD dictionary is shown in Figure 2. The same color means the same nature, and the properties are summarized to form the general meaning of "cup" concept.

Using $Z = \{z_1, z_2, \ldots, z_k\}$ to represent the IFD dictionary, the elements after word segmentation can find the one-to-one corresponding GUID in the dictionary Z, and the word $H = \{h_1, h_2, \ldots, h_k\}$ is obtained after semantic disambiguation. The step expression based on this method is as follows:

$$
\begin{aligned}
H &= \{h_1, h_2, \ldots, h_k\}\\
&= \begin{cases} h_i, & w_i = z_j,\\ w_i, & w_i \neq z_i. \end{cases}
\end{aligned}
\tag{12}
$$

This formula means that if the extracted i^{th} word segmentation result can be described in the dictionary Z, the word segmentation result will be stored and assigned, otherwise it will not be disambiguated [16]. The semantic disambiguation process of word segmentation results is shown in Figure 3.

2.4. NLP Machine Learning Calculation Model

2.4.1. NLP Calculation Model Based on LSTM Algorithm. LSTM is a time recurrent neural network, which is specially designed to solve the long-term dependence problem of the general recurrent neural network. LSTM (long short-term memory neural network) can overcome the problems of gradient disappearance and explosion and can be used for text training to achieve text representation that combines char-level and word-level [17]. All RNNs have a chained form of repeating neural network modules. In standard RNNs, this repeated structural module has only a very simple structure, such as a tanh layer. It contains an input layer, several hidden layers, and an input layer. It contains many neurons, also called storage units. The structure diagram is shown in Figure 4. Each storage unit has three "gates," namely, forget gate h_k, input gate i_k, and output gate t_k. These gates can maintain and adjust the state of the storage unit R_k [18]. At each step k, each gate structure receives the input x_k at this time and the output p_{k-1} from the output unit at time $k-1$ in the previous step.

The LSTM neural network is similar to the traditional feedforward network, and its training process is as follows: the first step is to determine the network structure and loss function. The second step is to initialize the input parameters and calculate the accuracy of the model through the loss function. The third step is to update the parameters, bias terms, and weights. The gradient information needs to be obtained by deriving the parameters through the loss function and then combined with the model learning rate to

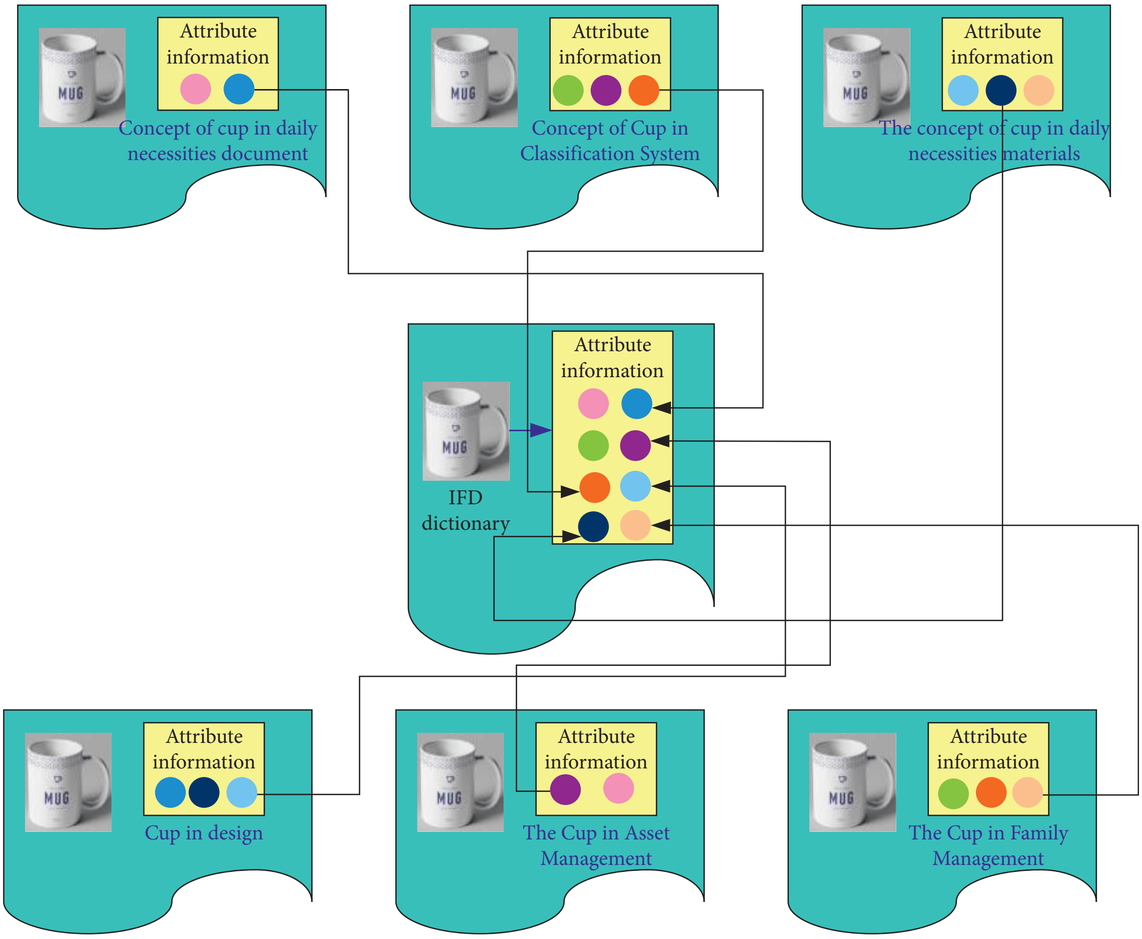

Attribute information
Concept of cup in daily necessities document
Attribute information
Concept of Cup in Classification System
Attribute information
The concept of cup in daily necessities materials
MUG
Attribute information
IFD dictionary
Attribute information
Cup in design
Attribute information
The Cup in Asset Management
Attribute information
The Cup in Family Management

Figure 2: The concept of "cup" in the IFD dictionary.

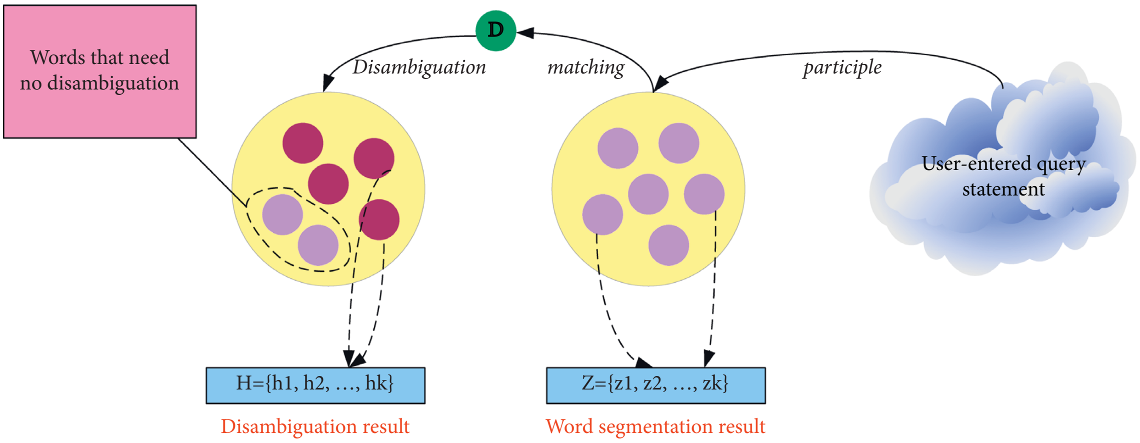


Figure 3: Semantic disambiguation process of word segmentation results.

determine it. When the gradient reaches the corresponding target accuracy, the modeling is completed [19].

2.4.2. NLP Calculation Model Based on FastText Calculation. The FastText model also has only three layers: input layer, hidden layer, and output layer. The input is a number of words represented by vectors, and the output is a specific target. The FastText model is a fast and efficient text classification representation and classification model proposed by Facebook. It performs better in languages with rich morphology and highlights the efficient training speed on large datasets [20]. It is similar to the CBOW structure in the Word2Wec model. It also has input, hidden, and output layers, and it uses Softmax to optimize the model structure. The structure is shown in Figure 5.

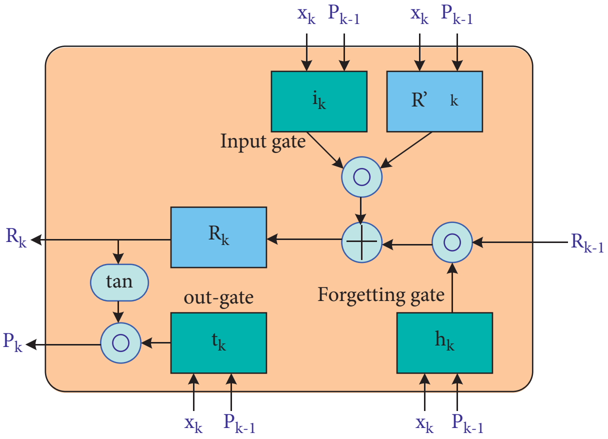


Figure 4: Long and short-term memory neural network structure.

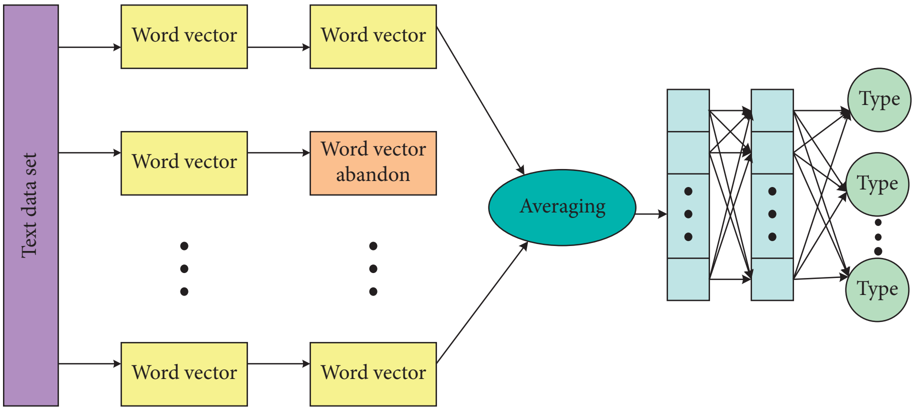


Figure 5: FastText model.

Input the word vector K $(k_1, k_2, \ldots, k_M)$ that has undergone word embedding processing, and the hidden layer calculation formula is as follows:

$$H = \frac{\sum_{j=1}^{M} k_j}{M}. \tag{13}$$

There are three main categories of linear classifiers: perceptron criterion function, SVM, Fisher criterion. The document is represented by the mean value of the word vector in the input layer, and then the output value is input into the linear classifier, and the Softmax is used to establish the mapping. The loss function is as follows:

$$L = \frac{-\sum_{j=1}^{M} \sum_{i=1}^{N} 1\left(x_j = i\right) \log\left(x'_{ji}\right)}{M}, \qquad x'_j = \operatorname{softmax}(H_i). \tag{14}$$

Normalize the Softmax function.

$$\operatorname{softmax}(H_i) = \sigma(H_i) = \frac{e^{H_j}}{\sum_{i=1}^{N} e^{H_i}}. \tag{15}$$

Complex language information conceals the complex emotional needs of users. If the model focuses too much on specific training data and misses key information, it will easily lead to overfitting. The general solution method adds a positive term to the loss function. Add regularization to the loss function L.

$$D = -\log(w_0|w_1) + \mu \sum_{i=1}^{A} w'_i + \lambda \sum_{j=1}^{A} w'_j. \tag{16}$$

At this time, the word vector update method is as follows:

$$w_{kj} = w_{kj} - \rho \sum_{i-1}^{A} (x_i - t_i) \bullet w'_{ij} - \lambda w_{kj}. \tag{17}$$

2.4.3. NLP Calculation Model (GCN) Based on Graph Convolutional Network Algorithm. A convolutional neural network (CNN) is a feedforward neural network whose artificial neurons can respond to surrounding units within a partial coverage. The graph convolutional neural network directly applies the multilayer neural network to the graph structure data and embed the graph according to the neighboring points. A layer of GCN can be defined as follows:

$$\begin{cases} G(1) = \varphi(\tilde{N} X W), \\ \tilde{N} = D^{-1/2}. \end{cases} \tag{18}$$

The adjacency matrix is denoted by N, $\tilde{N}$ is the symmetric normalization, W is the weight matrix, ϕ is the activation function, and $G(1)$ is the next hidden state of each fixed point after one iteration.

The convolution process is shown in Figure 6. After constructing the graph model through convolution, the neural network is subsequently added according to the downstream tasks. The formula is as follows:

$$F = g(Y, P) = \mathrm{softm}ax(\tilde{N}\phi(\tilde{N}XW)). \tag{19}$$

Y represents all fixed-point feature sets, P represents the adjacency matrix corresponding to the vertex, and $g(\cdot)$ represents the state update function.

2.5. NLP Model Evaluation Indicators. NLP is an acronym for neurolinguistic programming. The evaluation strategy is used to evaluate the quality of the established NLP model. To accurately reflect the recognition effect of the model from multiple angles, multiple evaluation indicators are generally used to evaluate the NLP classification performance of the calculation model. Common evaluation indicators are introduced below.

2.5.1. Precision. The algorithm formula is as follows:

$$\mathrm{precision} = \frac{N_{\mathrm{correct}}}{N_{\mathrm{correct}} + N_{\mathrm{false}}}. \tag{20}$$

2.5.2. Recall Rate. Recall is relative to the text dataset to be classified. It represents the ratio of the text information correctly classified by the calculation model to the text information that should be classified correctly. The algorithm formula is as follows:

$$\mathrm{recall} = \frac{N_{\mathrm{correct}}}{N_{\mathrm{correct}} + N_{\mathrm{nc}}}. \tag{21}$$

Among them, $N_{\mathrm{correct}} + N_{\mathrm{nc}}$ is the amount of information that should be classified correctly.

2.5.3. Accuracy. The accuracy rate is relative to all classifications of NLP results. Assuming that there are three types of text datasets, namely, A, B, and C, the accuracy rate means that the total number of accurately identifying categories A, B, and C accounts for the total number of datasets. The algorithm is as follows:

$$\mathrm{accuracy} = \frac{N_{Ac} + N_{Bc} + N_{Cc}}{\left(N_{Ac} + N_{Bc} + N_{Cc}\right) + \left(N_{Af} + N_{Bf} + N_{Cf}\right)}. \tag{22}$$

Among them, N_{Af}, N_{Bf}, and N_{Cf} are not the amount of information that should be recognized but not recognized in categories A, B, and C, respectively.

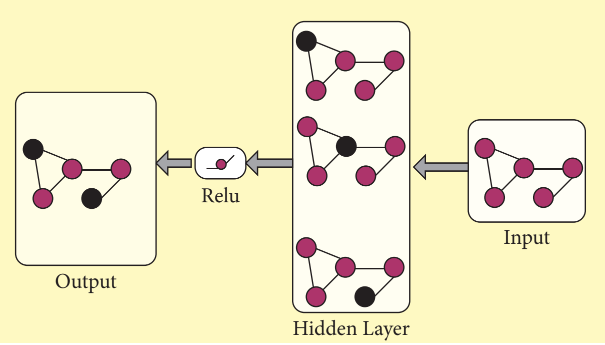


Figure 6: Graph convolutional neural network architecture.

2.5.4. F1 Value (F1_Score). Since the accuracy and recall rate examine the performance of different dimensions of the calculation method, there is a certain contradiction, i.e., the recall rate will be lower when the accuracy rate is high, and the accuracy rate will be low when the recall rate is high. Therefore, the F-value index is also used to measure the performance of the method, so that both the accuracy of the algorithm model and the recall rate can be evaluated. The calculation method is as follows:

$$F1 = \frac{2 * \mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}. \tag{23}$$

3. Significance and Innovation

3.1. Significance. With the development of database and data warehouse technology, the amount of data accumulated by people has increased at an unprecedented rate. Whether it is commerce, enterprises, scientific research, or government agencies, they have accumulated a large number of data stored in different forms. How to obtain meaningful information more conveniently has always been a research hotspot. The emergence of deep learning neural network and its application in natural language processing can aggregate edge information to update the vector representation of nodes, pay attention to more detailed and rich text features, and effectively help text understanding. This paper hopes to improve the efficiency and accuracy of text classification and recognition through the research on the calculation model of natural language processing and helps mine the relationship information between texts to improve the effect of natural language processing-related tasks.

3.2. Innovation. With the massive growth of network information and the continuous updating of intelligent requirements, text processing has become a more and more important research direction. The innovation of this paper is that to make the experiment more reasonable and effective, a comprehensive text dataset is selected. The data comes from the news dataset of Sogou laboratory, and the dataset is preprocessed. The results show that the anti-over-fitting effect will be better if the anti-over-fitting algorithm is not carried out, and the comprehensive analysis should be combined with its word vector model. The performance of three natural language processing models is compared with the loss function.

4. Experiment and Analysis of the NLP Computing Model

4.1. Experimental Design

4.1.1. Experimental Environment Configuration. This experiment will compare the model performance of the NLP computational methods proposed in this paper. The hardware configuration of the experiment is as follows: AMD R6-5600 CPU, 16 GB memory. The software configuration of the experiment is Python and the deep learning framework PyTorch under Windows 10 system. The following is the specific configuration of the experimental parameters: among them, the dataset is divided into a training set and a test set. The learning rate is 0.0001. The batch data size is 64. The total number of cycles is 200. The loss function uses the cross-entropy loss function, and the optimizer is SGD.

4.1.2. Corpus Preparation. To reflect the versatility and effectiveness of the algorithm, this paper selects a dataset with a strong comprehensive text type. The source of the experimental data in this paper is the news data of Sogou Lab, and preprocessing is carried out on this basis. Because of the large amount of data in the original dataset, to save the training time of the neural network model, part of the data was randomly selected as the experimental dataset in this experiment, including 6 types of news data, such as entertainment, finance, culture, and health. To avoid unbalanced sample distribution affecting the experimental results, each category is evenly distributed in the dataset, as shown in Table 1.

Table 1: Description of experimental data.

Type	Training	Test	Verification
Entertainment	4000	800	400
Finance	4000	800	400
Culture	4000	800	400
Health	4000	800	400
Politics	4000	800	400
Science and technology	4000	800	400

4.1.3. Experimental Steps. The experiment in this article will first perform statistical analysis on the accuracy of the test set and training set. Then, use the data of the validation set to perform an experiment to compare the performance of the Word2Vec-CBOW word embedding model proposed above and the model optimized with hierarchical Softmax. Then, compare the several natural language classification models (NLP calculation model based on LSTM algorithm, NLP calculation model based on FastText calculation, NLP calculation model based on graph convolutional network algorithm (GCN)) mentioned in the article, and finally, experiments will be carried out on the degree of fit and classification accuracy of the calculation method with relatively better performance. The evaluation indicators of various NLP calculation methods roughly include precision, recall, accuracy, and $F1$ value ($F1$_score).

4.2. Experimental Results and Analysis

4.2.1. Accuracy of Training Set and Test Set. Using Sogou Lab news data as the dataset, several algorithms in the training set and the test set are simulated and compared, and all algorithms are trained 100 times. The experimental results are shown in Figure 7, where the abscissa and ordinate, respectively, represent training times and accuracy rate. As shown in the figure, among the three types of algorithms, GCN has the best text classification accuracy in both the test set and the training set. The classification accuracy of LSTM is lower, which is the worst performance among the three algorithms. At the same time, from the perspective of convergence speed, GCN's NLP model based on the graph convolutional neural network obtains the optimal value faster in both training and test processing, as shown in Figure 7(b). In the test processing, the optimal value of the processing classification accuracy is obtained by only 6 iterations, which is at least 10 cycles earlier than other algorithms. Therefore, the performance of the GCN algorithm for NLP is the best among the three algorithms, and it takes less time and is more suitable for practical applications.

Table 2 shows the evaluation index results of all algorithms used in the experiment after 100 trainings, including accuracy, recall, precision, and $F1$ value. Experimental data shows that, compared with the LSTM and FastText algorithms, the evaluation index of the GCN algorithm is higher. It shows that the algorithm performance of the graph convolutional neural network model is better. The classification accuracy of the GCN algorithm reaches 89.94%, the recall rate reaches 88.16%, the accuracy rate reaches 90.28%, and the $F1$ value reaches 89.99%.

4.2.2. Fitting Effect Analysis. Overfitting will affect the accuracy of neural network classification. The commonly used methods to prevent overfitting include the Dropout algorithm that adds regularization terms and nonrandom probabilities to the loss function. To verify their effectiveness in preventing overfitting in the training and testing process, the Sogou laboratory dataset is also used. In this paper, the accuracy and $F1$ value of several NLP calculation models that use the loss function to increase the regularization term and the nonrandom probability of the Dropout algorithm are used to perform experimental statistics on the classification and recognition accuracy and $F1$ value of the other parameters. Compared with the average accuracy rate and average $F1$ value data without overfitting prevention processing, the comparison result is shown in Figure 8. The left side of the figure is the accuracy rate comparison chart, and the right side is the $F1$ value comparison chart. The red in the figure indicates the experimental results of the anti-over-fitting algorithm, and

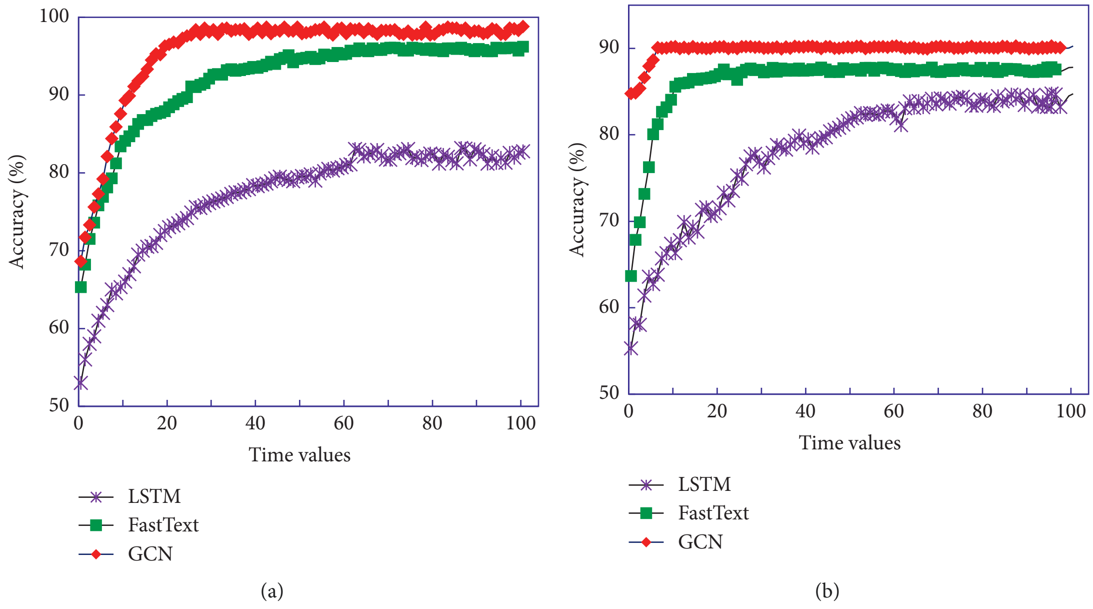


Figure 7: Accuracy result graph of training set and test set. (a) Accuracy of training. (b) Accuracy of test.

Table 2: Evaluation index results of each algorithm.

Type	Precision (%)	Recall (%)	Accuracy (%)	$F1$_score (%)
LSTM	82.31	81.98	80.65	81.73
FastText	87.66	85.44	84.32	86.82
GCN	89.94	88.16	90.28	89.99

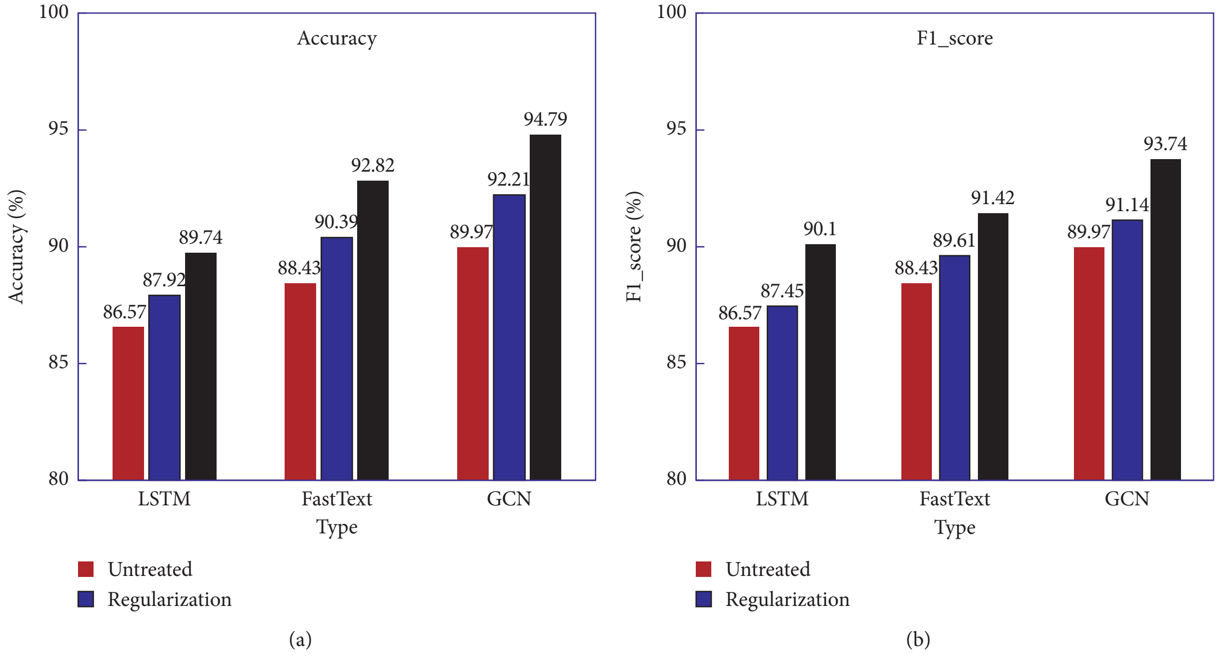
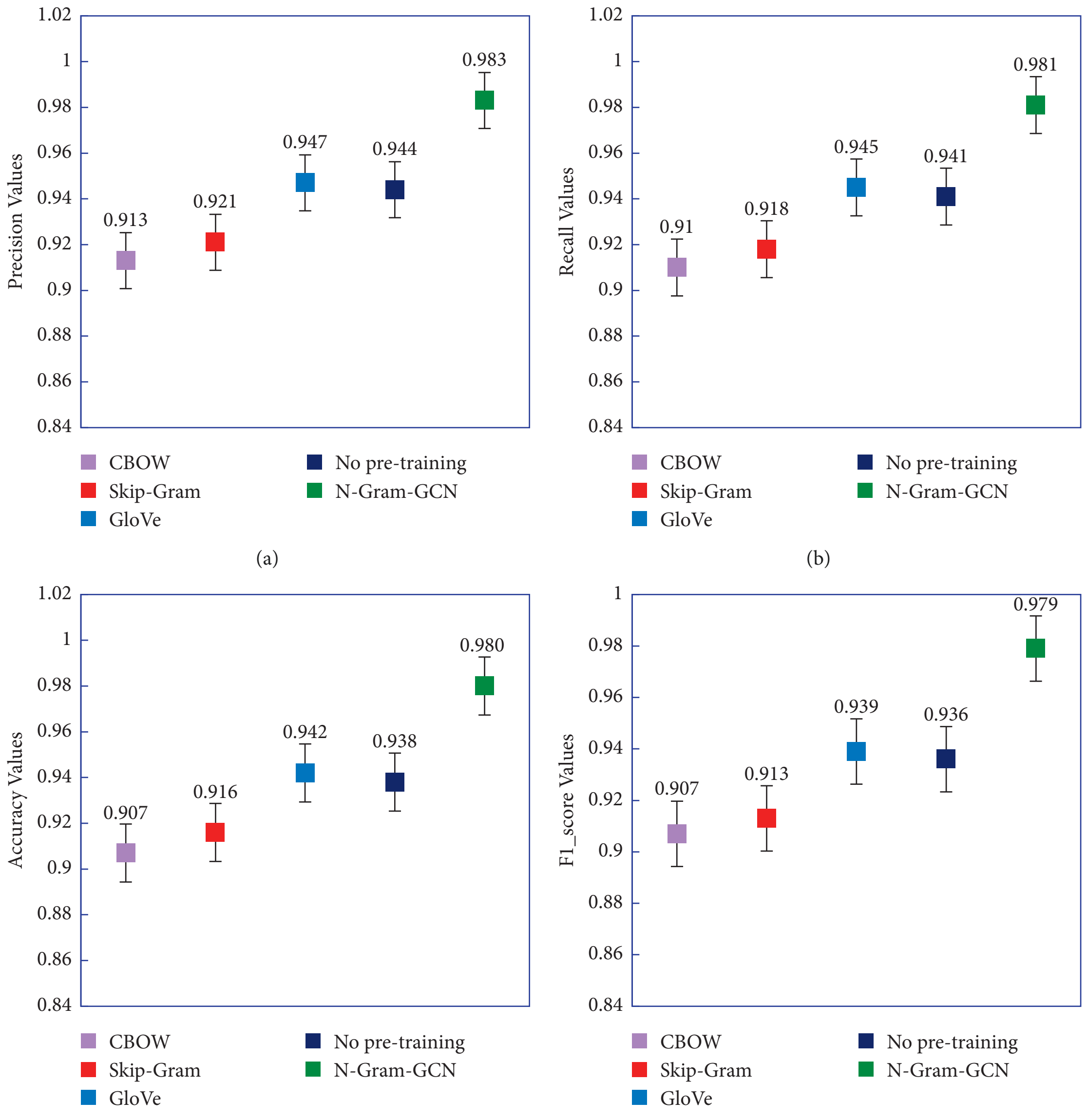


Figure 8: Anti-over-fitting effect diagram.

the blue and black are the results of the algorithm with increased regularization and dropout (dropout method) anti-over-fitting. From the experimental results, it can be seen that the accuracy and $F1$ value of several algorithms that have undergone anti-over-fitting processing are higher than those without anti-over-fitting processing. It can be concluded that for the three types of NLP calculation methods of LSTM, FastText, and GCN, the two anti-over-



(a)

(b)

(c)

(d)

Figure 9: Experimental results of the word embedding model. (a) Precision. (b) Recall. (c) Accuracy. (d) $F1$_score.

fitting methods proposed above are compared, and the effect of the Dropout algorithm is significantly better than the method of increasing the regularization term.

4.2.3. Comparative Analysis of the Performance of NLP Computing Models

(1) Performance Comparison of NLP Word Embedding Models. The experiment compares the effects of various word embedding methods on the performance of the GCN network NLP model. The parameter configuration is consistent with the previous training and test processing. The experimental results verify that the GCN word embedding model based on N-Gram features improves the task effect. The experiment consists of three parts, namely, the performance of the model when using Word2Vec word vectors (CBOW and Skip-Gram), GloVe word vectors, and without pretraining word vectors. The experimental data results on the test set are shown in Figure 9. Figures 9(a)–9(d) represent the experimental results of the index accuracy, recall, precision, and $F1$ value, respectively.

It can be seen from the figure that the task effect of using GloVe word vector is better than that of Word2Ve word vector, however, the gap is not very large. For the two different structures in the Word2Vec model, the effect of Skip-Gram is slightly better than that of the CBOW model. Without pretraining the word vector, the word embedding process needs to be combined with the training of the deep learning model, that is to say, the word embedding matrix will be used as a part of the deep learning model parameters, and it will be continuously optimized during the training process. It can also be clearly seen from the figure that the word vector effect obtained without pretraining the word vector (cotraining of the

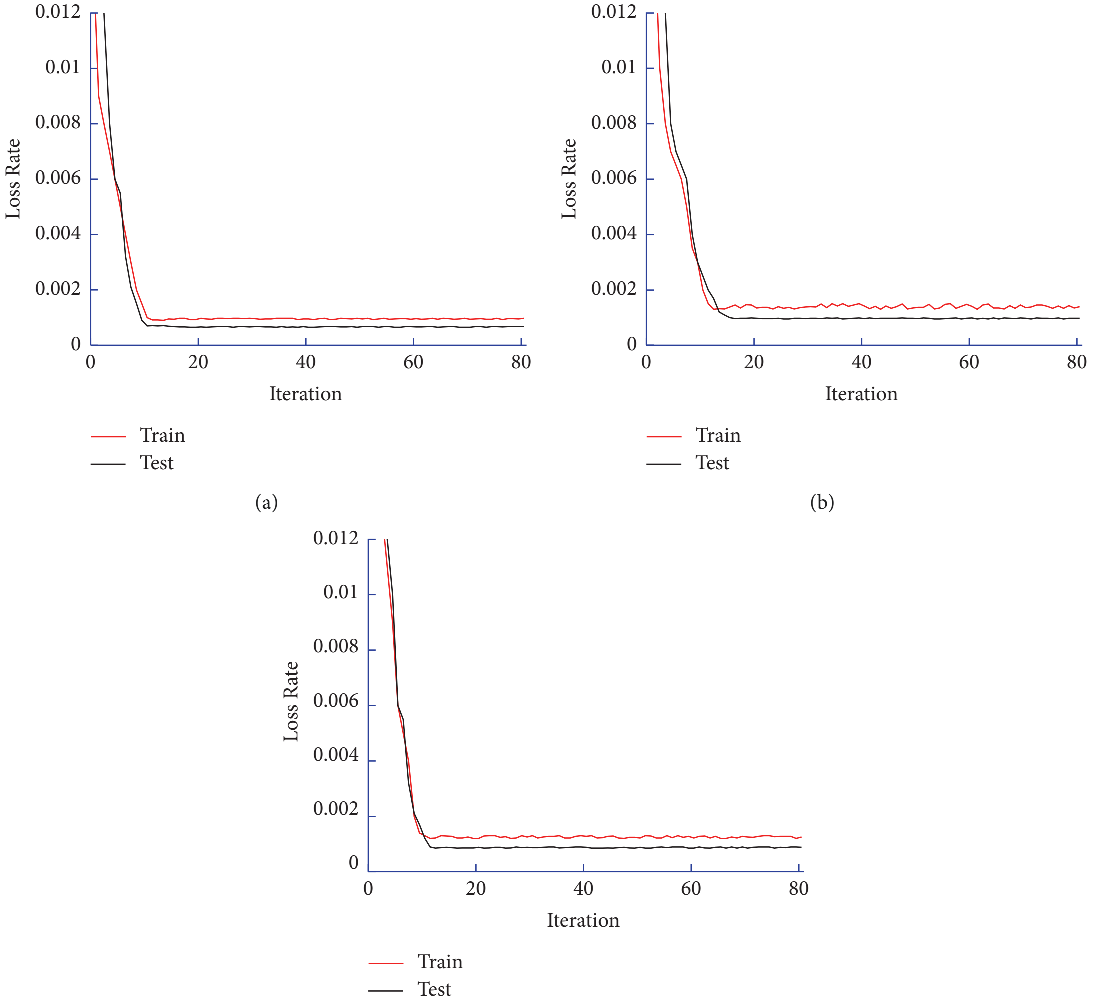


Figure 10: Algorithm loss function curve. (a) GCN. (b) Fasttext. (c) LSTM.

deep learning model) is better than the word vector effect of GloVe and Word2Vec. The reason may be that the word embedding learning is closely related to the target task learning in the common training process, and supervised word embedding training is carried out. The GCN based on the *N*-Gram feature's experiment on the test set has achieved the best task effect. All index values are the highest among several word vectors, and the accuracy, recall, precision, and *F*1 values are all above 0.97. Vector expression is the most efficient of several methods, probably because word structure is considered during the training process of this method.

(2) Comparative Analysis of the Performance of NLP Models. First of all, this article conducts experiments on the training levels of the several deep learning models mentioned above. The data used is the news dataset of Sogou Lab. Here, the training set and the test set are divided according to 4 : 1, the number of iterations is 80, and every 500 sample data is a batch for training. Use Adam as the model optimizer. The learning rate is still 0.001, and the training dataset is randomly scrambled. Cross-entropy is used as the loss function to test various NLP models, and the experimental results are shown in Figure 10. Figures 10(a)–10(c), respectively, show the loss function curves of the GCN, FsatText, and LSTM deep learning models on the test set and training set when the number of task processing iterations is 80. The red represents the training set curve and black represents the test set curve. As a result, it can be seen that the loss function of the three types of deep learning networks after the iterative is faster and tends to converge quickly. The fastest convergence rate is the computational model based on the graph convolutional network (GCN), followed by LSTM, and finally, FastText, for the comparison of the loss function value. It can also be clearly seen that the loss function value of the GCN after reaching the stability is the smallest, FastText is the second, and LSTM is the last. Based on the above analysis, the performance of GCN's NLP model based on the graph convolutional network will be better and the learning speed will be faster.

Finally, this article analyzes the results of NLP calculation methods and selects the experimental training data to randomly allocate 2400 news texts in six categories, and

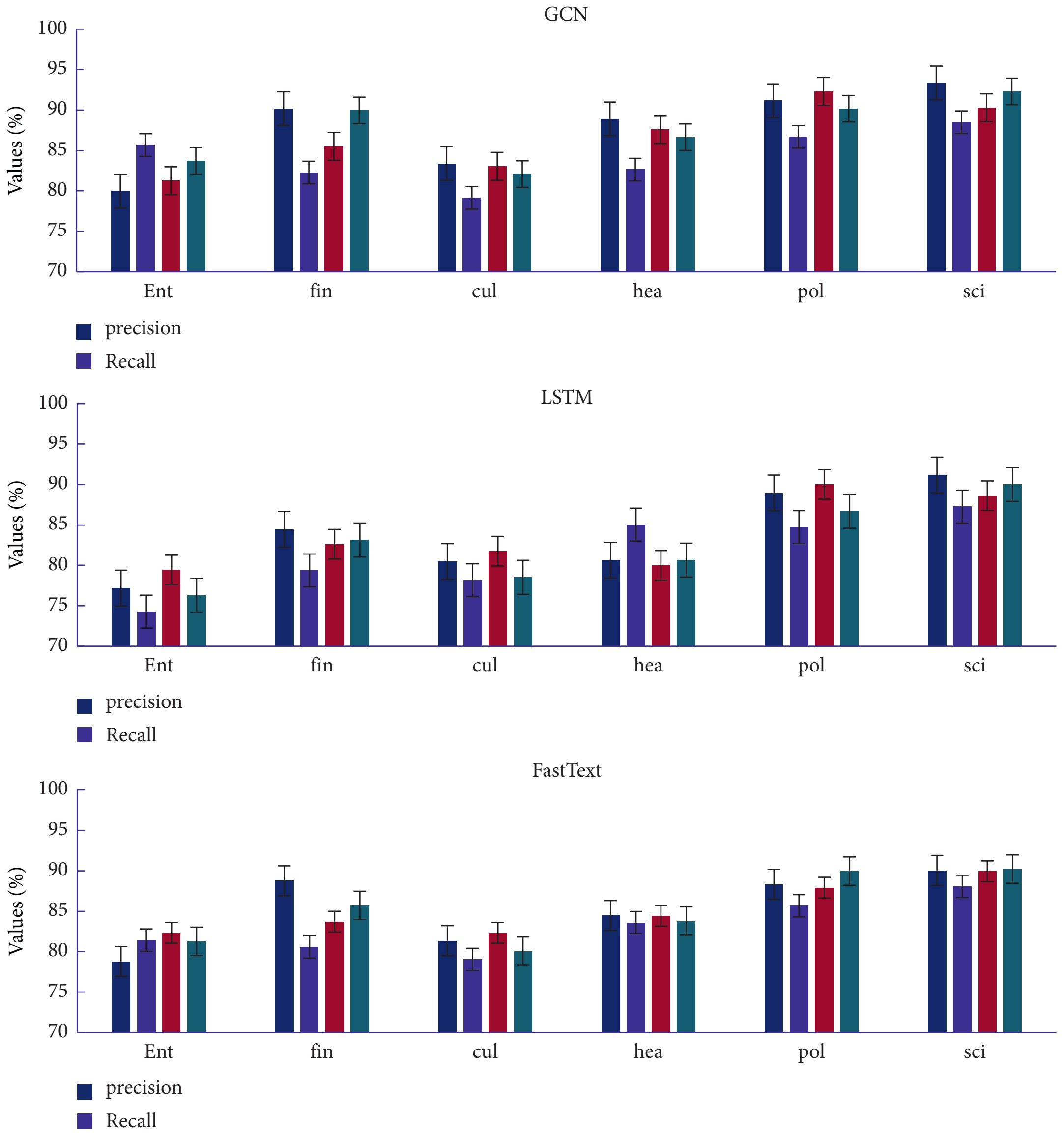


Figure 11: Algorithm processing evaluation index results.

verifies that the test set sample data is 600 news data randomly assigned using GCN, LSTM, and FastText NLP computing methods. The experimental results are shown in Figure 11. The figure shows the index results and text recognition results. The six news texts in the figure are replaced by the first three letters of each name, Ent-Entertainment, fin-finance, cul-culture, hea-health, pol-politics, and sci-science and technology.

Through the result index comparison chart, we can clearly see the differences in the accuracy, recall, exactnes,s and *F*1 vaule of the three NLP models. Combining the data in the figure and the average value of the indicators in Table 3, it can be concluded that the GCN model has the best performance level on these four indicators. The average values of precision, recall, accuracy, and *F*1 score of this model are 87.82, 84.15, 86.66, and 87.48, respectively. From the recognition results of the six news texts, it can be seen that in the news texts of the entertainment (Ent) and culture (cul) categories, the three NLP methods reflect the worst recognition processing effect. The best recognition processing effect is reflected in current affairs (pol), technology (sci), and finance (fin).

Table 3: The average value of the identification results of the three processing methods.

	GCN	LSTM	FastText
Precision	87.82	83.81	85.30
Recall	84.15	81.47	83.07
Accuracy	86.66	83.73	85.11
F1_score	87.48	82.55	85.18

5. Discussion

Communicate with computers in natural language, which has long been pursued by people. Because it has both obvious practical significance and important theoretical significance, people can use the computer in the language they are most accustomed to, without spending a lot of time and energy to learn various computer languages that are not very natural and accustomed. People can also use it to further understand the mechanism of human language ability and intelligence. The purpose of this paper is to design a better-performing NLP calculation method model to explore the accuracy of the computer's understanding of the meaning of natural language texts. The core content of the article first introduces the relevant content of NLP feature engineering, including NLP word segmentation, word embedding, word classification and recognition, etc., and gives the basic model of word embedding, efficiency optimization methods, and semantic disambiguation methods. Secondly, several NLP machine learning models FastText, LSTM, and GCN, which are mainly studied in this article, are described in related theories. Then, it introduces several evaluation indicators of model processing. Finally, it is the experimental part, which is mainly divided into pre-experiment, fit analysis, word embedding model analysis, and NLP result analysis.

In the pre-experiment, this article deals with the test set and training set of the three types of algorithms, and GCN shows a good task effect in the experiment at this stage. In the analysis of the fitting effect, the article compares the effects of anti-overfitting algorithms, and it turns out that the effect of the Dropout algorithm is better than the effect of increasing the regularization term. In the word embedding model experiment, the results of the GCN task processing method using Word2Vec word vector, Glove word vector, N-Gran, and no pretraining word vector are compared, and the results show that the performance of GCN based on N-Gran feature vector is the best. In the final performance experiment, the index values of GCN are still the highest among several methods.

6. Conclusions

Letting computers use natural language texts to express given intentions and thoughts has always been the yearning and pursuit of researchers in the computer field, and some functions have been realized under the research of scientific researchers. The article still has many shortcomings. For example, the type of data resources used in the experiment is too single, there are still some imprecise points in the experimental procedures, and there are not many innovations in the design of text processing calculation methods and the establishment of the computer network security model. However, after experiments and research, the article has a more systematic grasp and understanding of the calculation methods of NLP and hopes to make a little contribution to the direction of computer machine learning text processing.

Data Availability

No data were used to support this study.

Conflicts of Interest

The author declares no conflicts of interest in this study.

References

[1] J. M. Nobel, S. Puts, F. C. H. Bakers, S. G. F. Robben, and A. L. A. J. Dekker, "natural language processing in Dutch FreeText radiology reports: challenges in a small language area staging pulmonary oncology," *Journal of Digital Imaging*, vol. 33, no. 4, pp. 1002–1008, 2020.

[2] R. Lou, D. Lalevic, C. Chambers, H. M. Zafar, and T. S. Cook, "Automated detection of radiology reports that require follow-up imaging using natural language processing feature engineering and machine learning classification," *Journal of Digital Imaging*, vol. 33, no. 1, pp. 131–136, 2020.

[3] Y. Tom, H. Devamanyu, P. Soujanya, and E. Cambria, "Recent trends in deep learning based natural language processing [review article][J]," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[4] J. Brooke, A. Hammond, and G. Hirst, "Using models of lexical style to quantify free indirect discourse in modernist fiction[J]," *Literary and Linguistic Computing*, vol. 32, no. 2, pp. 234–250, 2017.

[5] A. Onan, S. Korukoğlu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Systems with Applications*, vol. 57, no. Sep, pp. 232–247, 2016.

[6] N. Jung and G. Lee, "Automated classification of building information modeling (BIM) case studies by BIM use based on natural language processing (NLP) and unsupervised learning[J]," *Advanced Engineering Informatics*, vol. 41, no. AUG, pp. 100917.1–100917.10, 2019.

[7] M. Silberztein, F. Atigui, E. Kornyshova, and M. Farid, "Natural Language Processing and Information Systems," in *Proceedings of the 23rd International Conference on Applications of Natural Language to Information Systems, NLDB 2018*, pp. 389–396, Paris, France, June 2018.

[8] K. Shaalan, A. E. Hassanien, and F. Tolba, "Intelligent Natural Language Processing: Trends and Applications," in *Automatic text classification using neural network and statistical approaches*pp. 351–369, China, 2018.

[9] A. V. Karhade, M. E. Bongers, O. Q. Groot et al., "Natural language processing for automated detection of incidental durotomy," *The Spine Journal*, vol. 20, no. 5, pp. 695–700, 2020.

[10] N. Liu, Q. Wang, and J. Ren, "Label-embedding Bi-directional attentive model for multi-label text classification," *Neural Processing Letters*, vol. 53, no. 1, pp. 375–389, 2021.

[11] C. Lehmann, D. Fabbri, and M. Temple, "natural language processing for cohort discovery in a discharge prediction model for the neonatal ICU," *Applied Informatics*, vol. 07, no. 01, pp. 101–115, 2016.

[12] A. A. Abokhzam, N. K. Gupta, and D. K. Bose, "Efficient diabetes mellitus prediction with grid based random forest classifier in association with natural language processing," *International Journal of Speech Technology*, vol. 24, no. 3, pp. 601–614, 2021.

[13] H. Wang, J. He, X. Zhang, and S. Liu, "A short text classification method based on N-gram and CNN," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 248–254, 2020.

[14] Y. Liu, Y. Wan, and X. Su, "Identifying individual expectations in service recovery through natural language processing and machine learning," *Expert Systems with Applications*, vol. 131, no. OCT, pp. 288–298, 2019.

[15] A. Rago, J. A. Diaz-Pace, and C. Marcos, "Using semantic roles to improve text classification in the requirements

domain," *Language Resources and Evaluation*, vol. 52, no. 3, pp. 801–837, 2018.

[16] N. P. Whitehead, W. T. Scherer, and M. C. Smith, "Use of natural language processing to discover evidence of systems thinking," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2140–2149, 2017.

[17] H. Oyama, M. Komachi, and Y. Matsumoto, "Hierarchical annotation and automatic error-type classification of Japanese language learners' writing," *Journal of Natural Language Processing*, vol. 23, no. 2, pp. 195–225, 2016.

[18] X. Huang, J. Jiang, D. Zhao, F. Yansong, and H. Yu, "Natural Language Processing and Chinese Computing," in *Proceedings of the 6th CCF International Conference, NLPCC 2017*, pp. 318–328, Dalian, China, November 2017.

[19] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing [review article]," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[20] D. S. Carrell, S. Halgrim, D. T. Tran et al., "Using natural language processing to improve efficiency of manual chart abstraction in research: the case of breast cancer recurrence," *American Journal of Epidemiology*, vol. 179, no. 6, pp. 749–758, 2014.