Hindawi
Scientific Programming
Volume 2021, Article ID 9976001, 17 pages
https://doi.org/10.1155/2021/9976001

Research Article

Network Public Opinion Monitoring System for Agriculture Products Based on Big Data

He Liu,[1,2] Zekun Yu,[1] Xiangzhi Zhong,[1] and Helong Yu [1,2]

[1]*College of Information Technology, Jilin Agricultural University, Changchun 130118, Jilin, China*
[2]*Jilin Precision Agriculture and Big Data Engineering Research Center, Changchun 130118, Jilin, China*

Correspondence should be addressed to Helong Yu; yuhelong@jlau.edu.cn

Received 9 March 2021; Revised 24 April 2021; Accepted 24 May 2021; Published 10 June 2021

Academic Editor: Imran Sarwar Bajwa

The influence of online public opinion on agricultural product safety on the society is increasing. In order to correctly guide the direction of online public opinion on agricultural products, help the agricultural sector turn from passive to active public opinion, timely prevent the spread of negative public opinion, and reduce the negative impact on public opinion hot events, it is especially important to improve the ability of monitoring agricultural products' network public opinion. This research is based on big data technology to develop an agricultural products' network public opinion monitoring system that can collect, process, and analyze data in real time, discover and track hot topics, and calculate and visualize the polarity of public sentiment. The use of big data technology to increase the processing speed aims to strengthen the public's supervision of the public opinion on the network security of agricultural products and provide an effective basis of the decision-making of relevant departments.

1. Introduction

Solving the quality and safety of agricultural products is to better improve and protect people's livelihood. The quality of agricultural products will be quickly spread through various media and social networks. The Internet has almost become the main channel of information dissemination. The risk of public opinion on the quality and safety of agricultural products is generally caused by negative public opinion. The transmission and dissemination of various emotions, attitudes, and opinions on the quality and safety of agricultural products by netizens through the Internet will reduce the efficiency of the government's emergency response, reduce the government's credibility, mislead the public's perceptions, and causepotentialdangers such as chaotic socio-economic order [1]. After the spread of agricultural product safety issues, they were amplified and hyped. Without verifying the authenticity of the problem, it seriously affected consumers and industrial economy and brought unnecessary trouble to the quality and safety supervision of agricultural products. This increased the difficulty of supervision. And, it is difficult to respond positively and guide public opinion on a timely manner.

Use the agricultural product public opinion monitoring system to obtain timely public opinion issues and conduct correct guidance. As a soft power, Internet public opinion guidance plays an increasingly important role. It effectively controls the direction of public opinion, adjusts the content of public opinion, and grasps the size of public opinion. Manipulate the presence or absence of public opinion to realize the communication between the leader and the public [2]. In the era of big data, advanced computer technology should be used to conduct public opinion research [3, 4] As the number of monitored websites increases, the situation is complex, and the content is wide, and manual analysis of public opinion has been difficult to deal with. The use of information technology to establish a network public opinion monitoring and analysis system has turned public opinion into active guidance [5, 6].

With the rapid development of the Internet and the ever-increasing amount of information, it is necessary to use big data technology to solve the processing speed and storage bottlenecks of traditional public opinion monitoring in the context of the big data era. The application of big data enables deeper analysis and more accurate

prediction of social public opinion. Use the Hadoop open source platform to build a big data foundation, realize distributed storage of data, realize distributed computing and processing data with MapReduce and Spark, perform text processing on the collected data, and use appropriate algorithm models to classify and cluster text information to complete the text emotional tendency analysis and topic discovery and tracking, and the research has a certain degree of innovation..Grasping the development status of public opinion information on agricultural products, and providing real-time and effective public opinion analysis services for relevant government departments, is of great significance for guiding the correct direction of public opinion on agricultural products and eliminating the adverse effects of public opinion on the safety and quality of agricultural products.

2. Research Content and Methods

This research takes the agricultural product public opinion monitoring system as the main research object. On the basis of the existing public opinion monitoring system, it solves the problem of using big data distributed technology to realize the public opinion monitoring of agricultural products in the vertical field from the massive Internet information. It uses big data technology to store data; uses big data computing power to process and analyze data; and uses Chinese natural language processing technology, including Chinese word segmentation and text classification and text clustering algorithms to process and calculate text and to mine data such as topics and sentiment tendency in public opinion information. These technologies can improve the processing efficiency of existing public opinion systems. The system module diagram is shown in Figure 1.

Based on the existing public opinion system, this research analyzes the needs of agricultural public opinion monitoring and designs a public opinion monitoring system. The functions are divided into four parts: public opinion information collection, public opinion information processing, public opinion information analysis, and public opinion service. Information collection first uses the distributed crawler Scrapy-Redis to collect the HTML pages in the pre-prepared seed URL, stores the collected data into the database HBase, and then uses the content extraction algorithm to extract the content of the saved HTML pages and store them in HBase again. Finally the data in HBase is synchronized to SolrCloud by Zookeeper, so as to establish an index to provide efficient retrieval function. Write the Spark program in Python and use jieba to segment the extracted content of Chinese. Calculate the word vectored through word2vec, and perform natural language processing, text classification, and text clustering on the captured information on topic recognition and tracking and sentiment judgment of public opinion. Use Django, Bootstrap, and other technologies to construct the display function of the agricultural product network public opinion system to realize public opinion warning and data display.

3. Key Algorithms of Agricultural Product Network Public Opinion Monitoring System Based on Big Data

Text feature selection and extraction are also research hotspots. Text feature selection is to find words with strong distinctions. For example, after preprocessing coarse word segmentation, it will filter function words, pronouns, and stop words in the word segmentation results. These words generally appear frequently but have no clear meaning. By reducing the useless features of text, the efficiency of text processing can be improved, such as text classification; use text features' selection algorithms to select words that have an effect on distinguishing categories as text features. Text features' selection algorithms include the following.

3.1. Document Frequency (DF). The frequency of words appearing in the document collection is called document frequency (DF), which is calculated in the following formula:

$$\mathrm{DF}(t_i) = \frac{\text{The number of documents where the word } t_i \text{ appears}}{\text{Total number of documents in the document set}}. \tag{1}$$

Set the upper threshold f_u and lower threshold f_d of document frequency (DF), and count the document frequency of words of the document collection. The document frequency is lower than f_d ($\mathrm{DF}(t_k) < f_d$), the word is not representative, and the word is removed from the text feature space. The document frequency is higher than f_u ($\mathrm{DF}(t_k) > f_u$), the word is not representative, and the word is removed from the text feature space [7]. The final texts' feature space retains the words $f_u < \mathrm{DF}(t_k) < f_d$.

3.2. Chi-Square Test (CHI). The chi-square tests first give the hypothesis and calculate the theoretical value based on the hypothesis. The correct rate of the theoretical value is judged by the deviation from the observed value and the theoretical value. If the correct rate is large, the hypothesis of the theoretical value is considered correct. The deviation calculation is shown in the following formula:

$$\chi^2 = \sum_{i=1}^{k} \frac{(A_i - E_i)^2}{E_i}. \tag{2}$$

In the above equation, A is the observed value, E is the theoretical value, and k is the number of observed values. The closer the value of χ^2 is to 0, the more likely the hypothesis we make is correct. When the deviation is larger, the hypothesis we make is more incorrect. The numerical standard for measuring the magnitude of the deviation is measured by the chi-square distribution.

In the feature selection of text classification, the chi-square test is used to measure the correlation between categories and words. There is a set T containing feature words and a set C containing category labels.t_i belongs to the set T, and c_j belongs to the set C. The chi-square test can measure the correlation between t_i and c_j, assuming that t_i

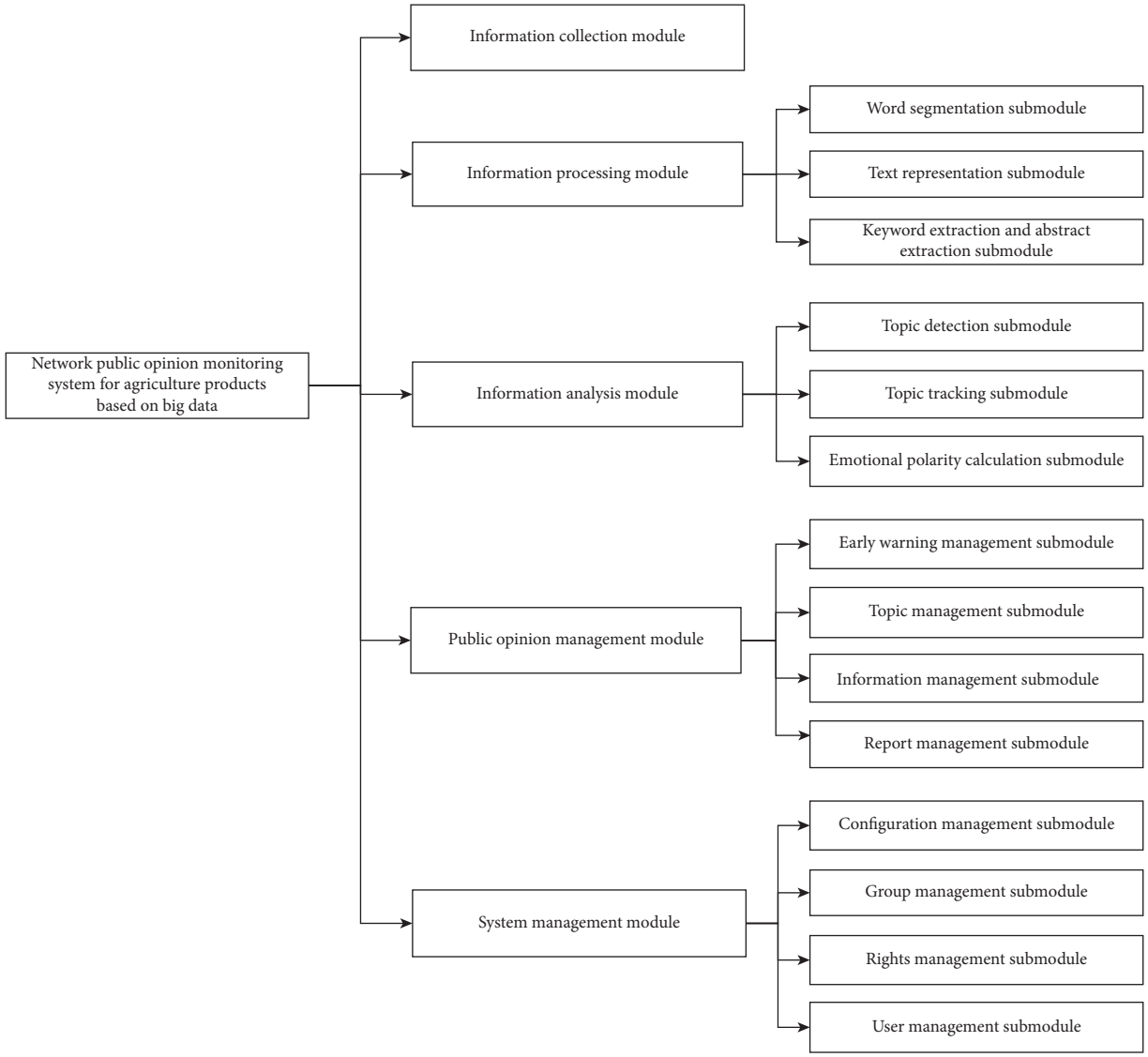

Figure 1: System module diagram.

and c_j conform to the chi-square distribution of the first degree of freedom. χ^2 of t_i and c_j is calculated as follows:

$$\chi^2(t_i, c_j) = \frac{N(AD - CB)^2}{(A + C)(B + D)(A + B)(C + D)}. \quad (3)$$

In the above equation, A is the number of texts in the text collection that contains t_i and belongs to c_j, B is the number of texts in the text collection that contains t_i but does not belong to c_j, C is the number of texts that do not contain t_i but belongs to c_j, D is the number of texts that do not include t_i or c_j in the collection, and N is the number of texts in the text collection. When the value of $\chi^2(t_i, c_j)$ is larger, it means that the correlation between t_i and c_j is stronger. When selecting features, the most relevant words with each category can be selected as text features for text processing such as classification.

3.3. Information Gain (IG). The information gain measures the importance of t_i words according to the amount of information brought by the word t_i to the classification system, so as to select feature words. The difference between the amount of information when the system contains t_i and when t_i is not included is the gain that this t_i word brings to the entire classification system. The information gains consider the effect of the word on all categories rather than a single category. There are a set T containing featrue words and a set C containing category labels. t_i belongs to the set T, and c_j belongs to the set C; there are m categories in the C category

set, and the amount of information is obtained by calculating the information entropy. The gain is calculated as follows:

$$G(t_i) = -\sum_{j=1}^{m} p(c_j)\log p(c_j) - p(t_i)\left[-\sum_{j=1}^{m} p(c_j|t_i)\log p(c_j|t_i)\right] - p(\overline{t_i})\left[-\sum_{j=1}^{m} p(C_j|\overline{t_i})\log p(C_j|\overline{t_i})\right]. \tag{4}$$

$p(c_j)$ represents the probability of occurrence of category c_j in the text set, $p(t_i)$ represents the probability of occurrence of the text containing the word t_i in the text set, $p(c_j|t_i)$ represents the conditional probability of the text that contains t_i and belongs to the c_j category in the text set, $p(\overline{t_i})$ represents the probability of occurrence of texts that do not contain the word t_i in the text set, and $p(C_j|\overline{t_i})$ represents the conditional probability of texts that do not contain t_i but belong to the c_j category in the text set.

3.4. Mutual Information (MI). Mutual information is based on the basis of information theory. The frequency of occurrence in class c_j is higher, while the words t_i that appear less frequently in other categories $\overline{c_j}$ have greater mutual information with class c_j. Through the above principles, the relevance of words and categories can be measured. The mutual information of the word t_i and the category c_j is calculated as follows:

$$I(t_i, c_j) = \log\frac{p(t_i, c_j)}{p(t_i)p(c_j)} = \log\frac{p(t_i|c_j)}{p(t_i)} \approx \log\frac{AN}{(A+C)(A+B)}. \tag{5}$$

A is the number of texts in the text collection that contains t_i and belongs to c_j, B is the number of texts in the text collection that contains t_i but does not belong to c_j, C is the number of texts that do not contain t_i but belongs to c_j, D is text number of texts that do not include t_i or c_j in the collection, and N is the number of texts in the text collection.

Both chi-square test and mutual information of text feature selection have the problem of low-frequency word defects. They only consider the case where the text contains t_i without considering the number of times t_i appears in the text, which makes the algorithm's selection of low-frequency words have selection errors. The text feature selects representative words in the text set as features and reduces the number of features, thereby reducing the dimensionality of the space vector, achieving the dimensionality reduction of the text vector, reducing the pressure of computer operations, and greatly improving the efficiency of text processing.

The text feature selection retains the important features of the text. The text feature extraction calculates the weight of each feature of each text to measure the different weights of different texts under the same feature. The text feature extraction algorithms are as follows.

3.4.1. TF-IDF Algorithm. The TF-IDF algorithm calculates the weight of the feature by integrating the frequency of a single word in a single text and the document frequency of the word. The calculation formula is as follows:

$$w_{i,j} = tf_{i,j} * \mathrm{id}f_i. \tag{6}$$

$tf_{i,j}$ is the frequency of the feature item, $n_{i,j}$ is the frequency of the word t_i in the text d_j, and $w_{i,j}$ is the weight of the word t_i in the text d_j. The calculation formula of the word frequency $(tf_{i,j})$ is shown in the following formula:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}. \tag{7}$$

$\mathrm{id}f_i$ is the inverse document frequency [8], $\left|\{j: t_i \in d_j\}\right|$ is the number of texts containing the word t_i, and $|D|$ is the total number of texts in the text collection, and the calculation formula is as follows:

$$\mathrm{id}f_i = \log\frac{|D|}{1 + \left|\{j: t_i \in d_j\}\right|} + 0.01. \tag{8}$$

The result of the TF-IDF algorithm is to analyze the weight of a single word in a single text. The text is long or short. When it is necessary to compare and calculate with each other, such as calculating cosine similarity, the numerical deviation of different vectors seriously affects the calculation result. The result vector of TF-IDF is normalized, and each component of the vector is limited to the range of [0, 1]. The normalization formula is as follows:

$$w_{i,j} = \frac{w_{i,j}}{\sqrt{\sum_{i=1}^{T} w_{i,j}^2}}. \tag{9}$$

3.4.2. Word2vec Algorithm. The word2vec algorithm obtains a fixed-dimensional word vector through the training of the text set. The traditional one-hot coded word vector has a large dimension and is too sparse, which can easily cause a memory disaster.

With the rise of deep neural networks (DNN), DNN is used to train word vectors to process the relationship between words, but the vocabulary is generally millions, and the output layer of DNN needs to calculate the output probability of each word. The amount of calculation is huge, and this process is very time-consuming. The DNN Model is shown in Figure 2.

Word2vec uses the CBOW or skip-gram model (see Figure 3). The CBOW model predicts the target word through context, but does not use the traditional DNN model. It uses a simple method of summing all input word vectors and averaging as the map from the input layer to the hidden layer, and the Huffman tree is used to replace the neurons from the hidden layer to the output layer. The leaf nodes of the Huffman tree function as the output layer neurons. The number of leaf nodes is the size of the vocabulary. The internal nodes play the role of hidden layer neurons, and the Huffman tree is called the hierarchical softmax model. The leaf node with higher weight of the Huffman tree is closer to the root node, and the code is shorter, while the leaf node with lower weight is far from the root node, and the code is longer to ensure the shortest weighted path. The mapping from the hidden layer to the

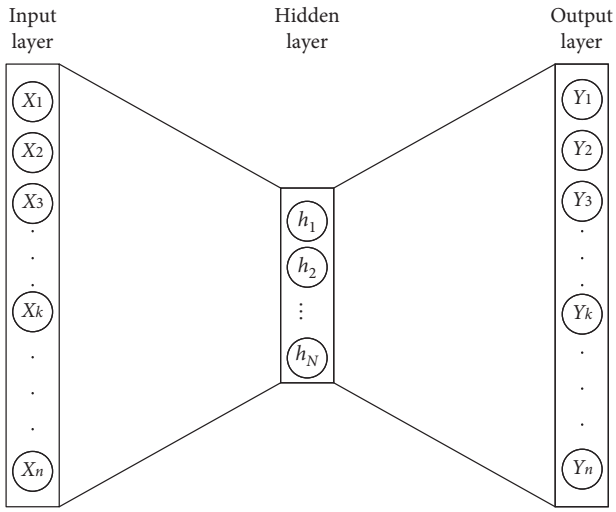

Figure 2: DNN model.

output layer in word2vcec follows the Huffman tree step by step, and the word vector of the root node is the word vector of the mapping from the input layer to the hidden layer, and then, the binary logistic regression method is used to specify the left side. The tree walk is a negative class (coded as 1), walking along the right subtree is a positive class (coded as 0), χ_w is the word vector of the internal node, and θ is the model parameter of the logistic regression of the internal node that needs to be obtained from the training sample. The method to distinguish $p(+)$ and $p(-)$ is to use the sigmoid function:

$$p(+) = \sigma\left(\chi_w^T\theta\right) = \frac{1}{1 + e^{-\chi_w^T\theta}}, \tag{10}$$

$$p(+) = 1 - p(-). \tag{11}$$

Using the Huffman tree, the calculation amount n of DNN is reduced to $\log_2 n$, where n is the size of the vocabulary, and the closer the word with higher weight is to the root node, the shorter the time to reach the goal. The goal of solving the Huffman tree is to find the word vector of the leaf node and the θ model parameters of the internal nodes. Suppose the word vector of the hidden layer mapping is χ_w, the number of summary points passed from the root node to the leaf node where the target word is w is l_w, the ith node passed is recorded as l_w, and the corresponding Huffman code is recorded as $d_i^w (d_i^w \in \{0,1\}, i \in \{2,3,4,5,\ldots,l_w\})$, and the internal node model parameters are expressed as $\theta_i^w (i \in \{1,2,3,4,5,\ldots,l_w - 1\})$. The log likelihood function of w is given as follows:

$$L = \prod_{j=2}^{l_w}\left(\left(1 - d_j^w\right)\log\left[\sigma\left(\chi_w^T\theta_{j-1}^w\right)\right] + d_j^w\log\left[1 - \sigma\left(\chi_w^T\theta_{j-1}^w\right)\right]\right). \tag{12}$$

Calculate the gradient expressions of θ_{j-1}^w and χ_w as follows:

$$\frac{\partial L}{\partial\theta_{j-1}^w} = \left(1 - d_j^w - \sigma\left(\chi_w^T\theta_{j-1}^w\right)\right)\chi_w, \tag{13}$$

$$\frac{\partial L}{\partial\chi_w} = \sum_{j=2}^{l_w}\left(1 - d_j^w - \sigma\left(\chi_w^T\theta_{j-1}^w\right)\right)\theta_{j-1}^w. \tag{14}$$

According to the gradient expression, the stochastic gradient ascent method can be used to iteratively solve θ_{j-1}^w and θ_{j-1}^w. Initially, the θ parameters of internal nodes and all word vectors χ_w are initialized randomly.

Solving the CBOW model based on the Huffman tree, the dimension of the word vector is assumed to be M, the context size of the CBOW model is $2c$, and there are c words in the front and c words in the back. From the input layer to the projection layer (hidden layer), find the $2c$ word vectors around w and take the average value in the following formula:

$$\chi_w = \frac{1}{2c}\sum_{i=1}^{2c}\chi_i. \tag{15}$$

From the projection layer to the output layer, we update our θ_{j-1}^w and χ_w through the gradient ascent method. χ_w is obtained by adding the $2c$ word vectors and averaging. The update of χ_w is to $2c$ word vectors $\chi_i (i \in \{1,2,\ldots,2c\})$ is updated, the update formula of θ_{j-1}^w is shown in formula (21), the update formula of χ_i is shown in formula (22), and η is the step size of the gradient ascent method:

$$\begin{aligned}\theta_{j-1}^w &= \theta_{j-1}^w + \eta\left(1 - d_j^w - \sigma\left(\chi_w^T\theta_{j-1}^w\right)\right)\chi_w,\\ \chi_i &= \chi_i + \eta\sum_{j=2}^{l_w}\left(1 - d_j^w - \sigma\left(\chi_w^T\theta_{j-1}^w\right)\right)\theta_{j-1}^w, \quad i \in \{1,2,\ldots,2c\}.\end{aligned} \tag{16}$$

Iteratively, update θ_{j-1}^w and $2c$ word vectors $\chi_i (i \in \{1,2,\ldots,2c\})$ until the gradient convergence ends the iterative calculation.

The skip-gram model is solved based on the Huffman tree. The input layer is the word vector of w, and the word vector of w is directly mapped to the projection layer (hidden layer). The output of the skip-gram model is a context size of $2c$ word vectors $\chi_i (i \in \{1,2,\ldots,2c\})$, and the skip-gram model does not iteratively update the input like the CBOW model, but iteratively updates the $2c$ outputs.

Huffman tree l_w encounters rare words with low weight, and l_w will be very large, and it takes continuous iterative samples until the gradient converges. To solve the complex calculation of Huffman tree rare words, negative sampling (NegativeSampling) can be used to solve the word2vec model. In the negative sampling, a total of $2c$ words before and after the central word w are recorded as context(w),

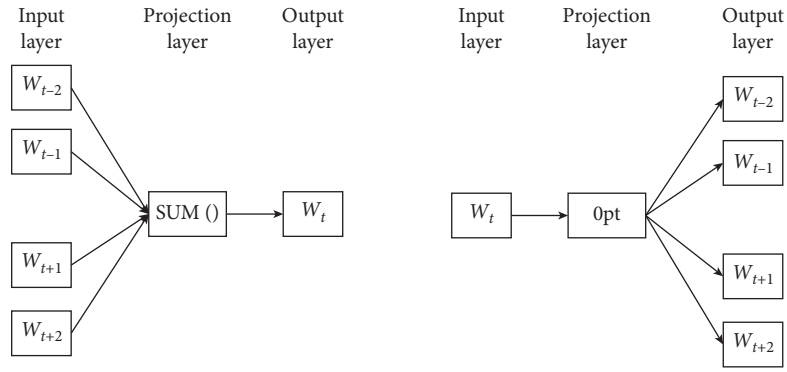

Figure 3: CBOW and skip-gram.

where w and context(w) constitute a positive example recorded as $\mathrm{POS}(w) = \{(\mathrm{context}(w), w)\}$; through negative sampling, we obtain neg central words wi ($i \in \{1, 2, \ldots, \mathrm{neg}\}$) that are different from w, so that context(w) and wi constitute neg negative examples, denoted as $\mathrm{NEG}(w) = \{(\mathrm{context}(w), w_i), i \in \{1, 2, \ldots, \mathrm{neg}\}\}$. Take the positive example and the neg negative examples as a sample set $\mathrm{ALL}(w) = \mathrm{POS}(w) \cup \mathrm{NEG}(w) = \{(\mathrm{context}(w), w_i), i \in \{0, 1, 2, \ldots, \mathrm{neg}\}\}$, when $i = 0$, wi is the positive example w. Perform binary logistic regression to get the model parameter θ_i corresponding to each word w_i ($i \in \{0, 1, 2, \ldots, \mathrm{neg}\}$) and the word vector of each word. The whole process is simpler than the Huffman tree. The log likelihood function of w is given in the following formula:

Table 1: Semantic similarity of word vectors.

Swamp rice	
Words	Cosine
Paddy field	0.686254
Rice	0.620230
Hybrid rice	0.602272
Paddy	0.585341

$$L = \sum_{c=0}^{\mathrm{neg}} y_i \log\left(\sigma\left(\chi_{w_0}^T \theta^{w_i}\right)\right) + (1 - y_i)\log\left(1 - \sigma\left(\chi_{w_0}^T \theta^{w_i}\right)\right), \quad y_i = 1, i = 0; y_i = 0, i \in \{1, 2, \ldots, \mathrm{neg}\}. \tag{17}$$

The update formulas of θ^{w_i} and χ_{w_0} are shown in equations (18) and (19). Similarly, the update of χ_{w_0} of the CBOW model and skip-gram model is synchronized to the context $2c$ word vectors using negative sampling:

$$\theta^{w_i} = \theta^{w_i} + \eta\left(y_i - \sigma\left(\chi_{w_0}^T \theta^{w_i}\right)\right)\chi_{w_0}, \tag{18}$$

$$\chi_{w_0} = \chi_{w_0} + \eta \sum_{i=0}^{\mathrm{neg}} \left(y_i - \sigma\left(\chi_{w_0}^T \theta^{w_i}\right)\right)\theta^{w_i}. \tag{19}$$

The similar words and similar values of rice are shown in Table 1. The similar words of Chinese medicinal materials and rice are obviously clustered together in Figure 4.

In order to compare the text representations of word2vec, tfidf (word frequency-inverse frequency), and bow (bag of words), the two-dimensional view of the text vector under random three types of text representations of tfidf, bow, and word2vec is calculated and drawn, respectively. Word2vec means that the text is the average of all word vectors of the text. The result is shown in Figure 5. The text vectors of tfidf and bow have obvious overlapping parts, and the boundaries of the three types of text represented by word2vec are more obvious.

Experimental comparison shows that word2vec has a stronger expression of text semantics and, at the same time, solves the high-dimensional sparse problem of tfidf and bow vectors, and bow text vectors perform poorly. In this paper, word2vec combined with tfidf will be used as the text feature input of the classification model.

3.5. Text Classification. Text classification categorizes texts of unknown categories into known categories, which involves manually classifying and labeling known text sets, using the labeled text of the training set combined with the text features of the unknown text to distinguish the text category. Text classification algorithms have methods based on traditional machine learning and deep neural network learning. Traditional machine learning has naive Bayes, *K*-nearest neighbor algorithm (KNN), support vector machine (SVM), neural network, etc. Deep neural network learning has fastText model, TextCNN model, TextRNN model, etc.

3.5.1. Naive Bayes (NB). The naive Bayes classifier is a probabilistic classifier that uses a bag-of-words model for text features and uses the frequency of each word as a document feature. Assume that the category $C = \{c_1, c_2, c_3, \ldots, c_m\}$ in the labeled text set has m categories of text. There is a text d to be classified, looking for the classification of d:

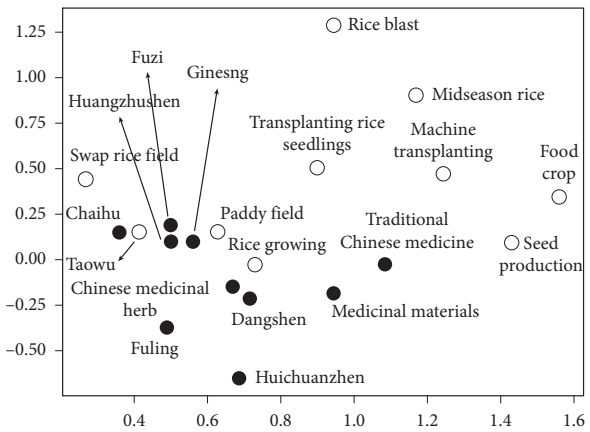

Figure 4: Vector two-dimensional representation of rice and Chinese medicinal materials.

$$p(c_i|d) = \frac{p(d|c_i)p(c_i)}{p(d)}, \quad i \in \{1, 2, \ldots, m\}. \tag{20}$$

Through the statistics of the training set text, it is easy to calculate the probability of a word in each category, but the probability may be small, and the product result will become smaller and smaller. The logarithmic function is introduced, $p(c_i|d)$:

$$p(c_i|d) = \log p(c_i) + \sum_{j=1}^{n} \log p\left(w_j|c_i\right), \quad i \in \{1, 2, \ldots, m\}. \tag{23}$$

3.5.2. K-Nearest Neighbor Algorithm (KNN). To classify text d, find the k texts closest to text d in the training text set. The classification of text d is based on the classification labels of these k texts. In simple terms, most of the classification labels of k texts belong to a certain category. Then, the text d also belongs to this category [9, 10]. The distance between the text d to be classified and the training sample can be calculated by Euclidean distance or cosine similarity [11, 12].

The advantages of KNN are suitable for automatic classification with relatively large sample size, but for small sample sizes, it is easy to cause misclassification. When the number of classifications of the training samples is unbalanced, the prediction accuracy of the text to be classified in a small number of categories is low.

3.5.3. Support Vector Machine (SVM). The SVM algorithm is a general learning method proposed by Vapnik and Bell Labs group in 1995, which is based on VC statistics and the principle of structural risk minimization [13]. The basic idea of the SVM classification method is to find a hyperplane in the n-dimensional space under the condition of linear

Formula (23) calculates the probability value of the text d under $C = \{c_1, c_2, c_3, \ldots, c_m\}$. Text d belongs to the category with the largest probability value. Assuming that the set of words in text d is $\{w_1, w_2, w_3, \ldots, w_n\}$, the calculation denominator $p(d)$ for each category is the same and can be omitted. Equation (22) is further simplified as follows:

$$p(c_i|d) = p(w_1, w_2, w_3, \ldots, w_n|c_i)p(c_i), \quad i \in \{1, 2, \ldots, m\}. \tag{21}$$

Since naive Bayes assumes that the attributes (feature items) are mutually independent, formula (22) can be obtained:

$$p(w_1, w_2, w_3, \ldots, w_n|c_i) = p(w_1|c_i)p(w_2|c_i)p(w_3|c_i)\cdots p(w_n|c_i), \quad i \in \{1, 2, \ldots, m\}. \tag{22}$$

The disadvantage of naive Bayes text classification is its conditional independence assumption. It assumes that words are independent and has no correlation. The text is regarded as a bag-of-words model, ignoring the influence of the word order on text classification. N-gram can be introduced to naive Bayes. The model improves text classification, and the following formula is improved to formula (30) (assuming that the set of w is arranged in the text order, using the 2-gram model):

$$p(c_i|d) = \log p(c_i) + \log p(w_1|c_i) + \sum_{j=1}^{n} \log p\left(w_j|w_{j-1}, c_i\right), \quad i \in \{1, 2, \ldots, m\}. \tag{24}$$

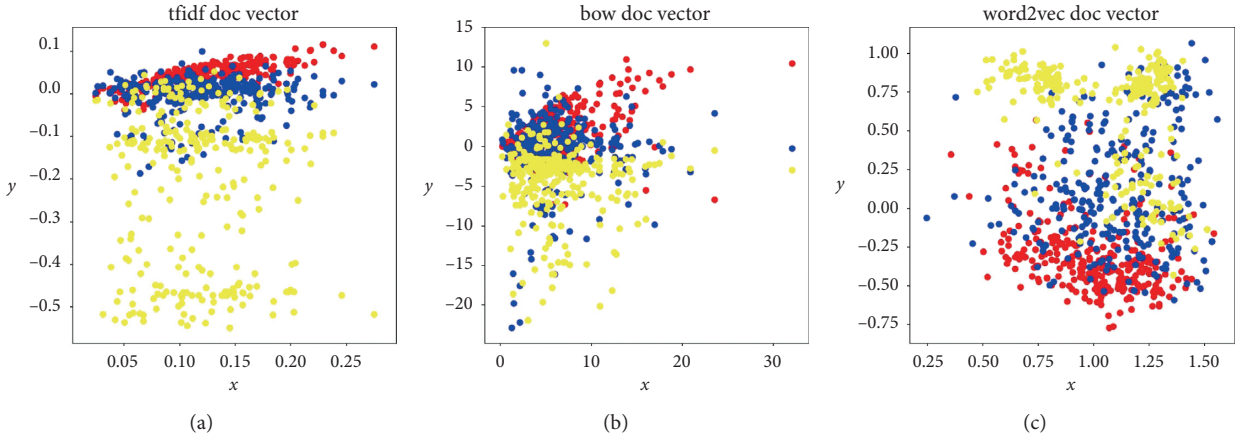

Figure 5: Document vector representation.

separability, distinguish two types of samples in the space, and solve the multiclassification problem by transforming it into a two-classification problem and then solve it.

In order to judge the performance of the classification algorithm, the necessary evaluation of the classification algorithm is performed, and the accuracy, precision, and recall are used to evaluate the performance of the model classification. The formulas are as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{25}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{26}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{27}$$

In the formula, P and N in FP, FN, TP, and TN represent the judgment result of the model, and T and F evaluate whether the judgment result of the model is correct. FP is false positive, which means that the prediction is of this type, but the actual number is not the number of this type; FN is false negative, which means that the prediction is not of this type, but is actually the number of this type; TP is true positive, which means that the prediction is of this type, and it is also actually the number of this category; TN is true negative, which means that the prediction is not of this category, and it is not actually the number of this category. Considering the accuracy rate and recall rate comprehensively, calculate the F-Score (harmonic mean); the β weight is 1, and the $F1$-Score value is calculated. The larger the value, the better the model classification performance. The formula of $F1$-Score is as follows:

$$F1 - \text{Score} = \frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}. \tag{28}$$

Plotting the true positive rate (TPR) and false positive rate (FPR) curve ROC is also a method to evaluate the classification model. The area under the ROC curve is AUC (Area under the ROC curve). The larger the AUC area, the better the classification effect.

3.5.4. TextCNN. TextCNN is the application of the convolutional neural network (CNN) in text classification. CNN initially achieved great success in the image field. CNN mainly captures local features. The CNN sentence classification model proposed by Kim [14] is shown in Figure 6.

The input layer of TextCNN is the word vector matrix in the text. Assuming that the text has n words and the word vector dimension is k, then the size of this matrix is $n \times k$. The word vector here can directly use the word vector calculated by word2vec, or it can be used as the embedding layer of the CNN model to participate in the back propagation algorithm for parameter optimization. The hidden layer of TextCNN is composed of a convolutional layer and a pooling layer. The convolutional layer has several different convolution kernels. The input matrix is subjected to convolution operations with several different convolution kernels to obtain several feature vectors. The pooling layer completes the work of reducing the dimensionality of the feature vector. There are usually average pooling and maximum pooling operations. TextCNN text classification generally chooses maximum pooling to compress each feature vector and select the maximum value of each feature vector. The output layer uses the softmax function to normalize the output vector and output the probability of each class. In the TextCNN experiment, you can arbitrarily combine multilayer convolution and pooling to achieve different experimental effects.

3.5.5. TextRNN. Recurrent Neural Network (RNN) introduces the concept of time series into the network structure, which has stronger adaptability in time series data analysis [15–17]. RNN processing time series data can save historical information and apply the information of the previous layer to the information of the lower layer. RNN training has the problems of gradient disappearance and gradient explosion [18]. Hochreiter and Schmidhuber. improved RNN, that is, long-term and short-term neural network (LSTM) [19], which can realize long-distance dependent information. The RNN model structure of LSTM is mostly used in text processing, as shown in Figure 7. LSTM adds cell state and gating unit to the structure of the original RNN to

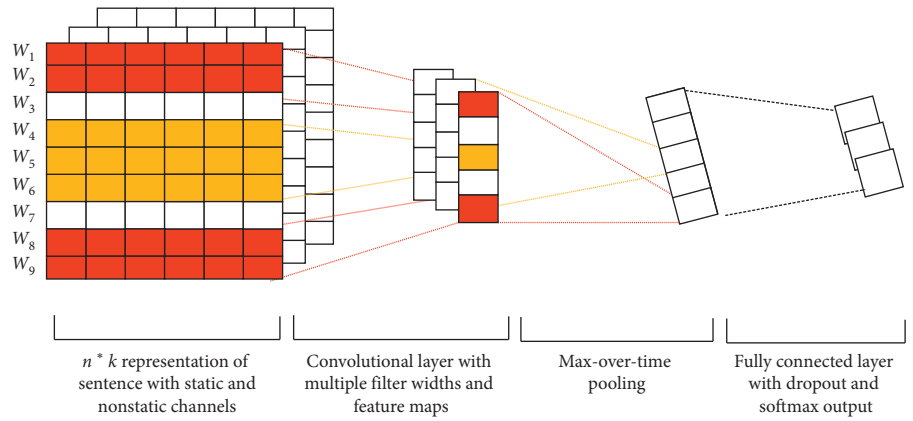

Figure 6: TextCNN model.

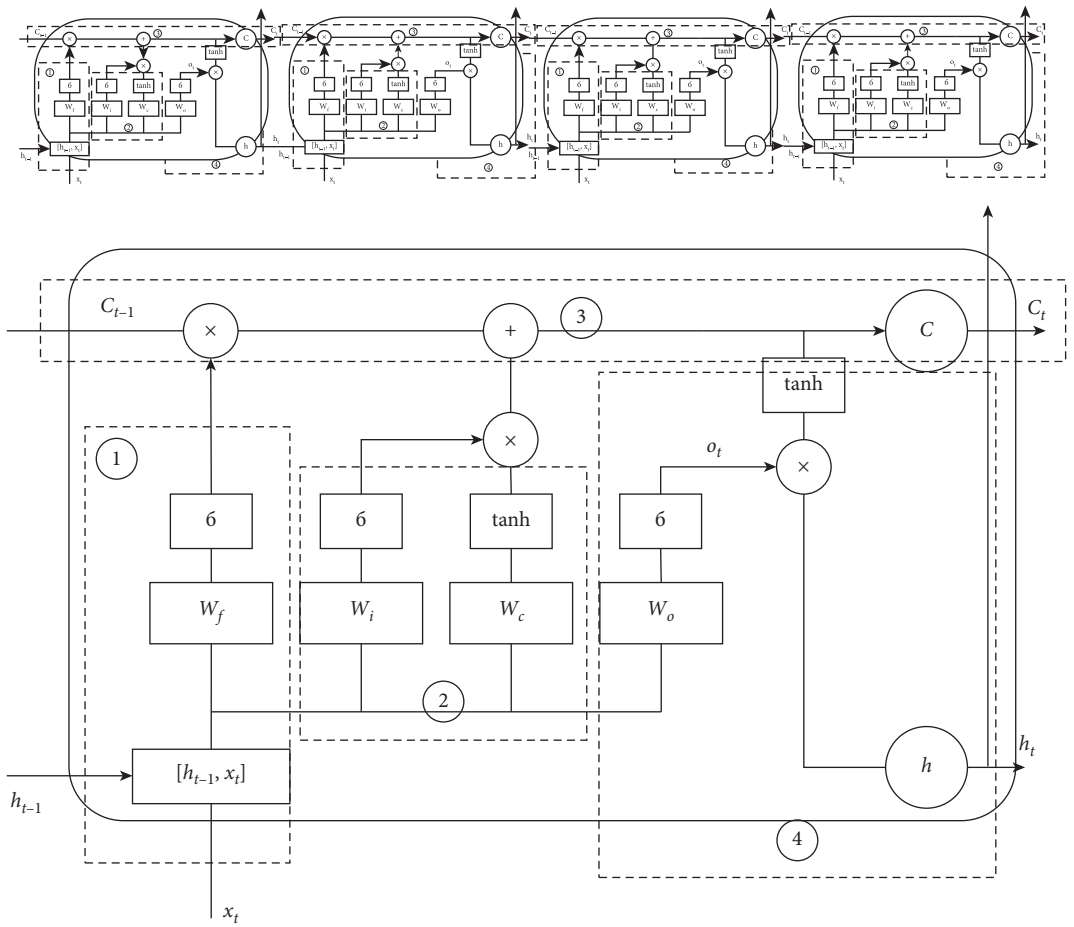

Figure 7: LSTM structure.

complicate the structure of the unit (hidden layer). Information can be added or deleted through the structure of the unit's input gate, forget gate, and output gate and can selectively send message.

In the text classification task, LSTM is connected by multiple cells. The input χ_t of each cell corresponds to a word in the text. h_t of the last cell is output to the fully connected softmax layer, and the classification result y is output.

3.6. Text Clustering. Text clustering is the process of automatically categorizing text collections. The classification of text collections is not determined in advance, but is obtained from the data itself. Text clustering is to maximize the similarity within classes and minimize the similarity between classes. Text clustering is an unsupervised learning method with a certain flexibility and high automatic processing ability [20]. According to the thought clustering algorithm, it

can be divided into partition clustering, hierarchical clustering, density clustering, and so on.

3.6.1. Divide Clusters. Dividing and clustering uses the split method to construct a dataset (N length) into K clusters ($K < N$). K-means belongs to the division clustering method. First, select the K initial centroids of the number of categories expected by the user, and randomly select the K centroids. Through distance calculation, the text is classified into the class of the closest mass point and the centroid of this class is recalculated; repeat the process until the position of the centroid does not change; then, the final result of clustering is obtained. The similarity calculation can use methods such as Euclidean distance to calculate the text vector to obtain the distance. The smaller the distance, the higher the similarity of the data. The K value of K-means needs to be determined in advance. For unsupervised tasks, the actual number of classifications of the dataset is not known. It is difficult to obtain the value of K. Generally, a rough estimate is obtained through the evaluation of clustering results and other hierarchical clustering. Based on the classic K-means algorithm, Ding Ruoyao introduced the idea of level-based, density-based, and partition-based to solve the problem of how many and how to choose the initial cluster center [21]. Update the centroid; if there are too many abnormal points, the centroid will be biased toward the coordinates of the abnormal points, resulting in a bad clustering effect. K-means uses Euclidean distance to measure the similarity of sample data, and the clustering results obtained are biased towards convex distribution, which is not friendly to nonconvex data clustering. And, the initial centroid is randomly selected, and the initial centroid has a certain influence on the clustering effect.

3.6.2. Hierarchical Clustering. Hierarchical clustering uses hierarchical decomposition to process a given dataset until the expected conditions are met. Hierarchical clustering has two schemes, "bottom-up" and "top-down." BIRCH adopts balanced iterative protocol and clustering, scanning the dataset in a single pass, and using the clustering feature tree to help fast clustering. The BIRCH algorithm does not need to input the category number K value. If the K value is not input, the number of tuples of the final clustering feature tree is the final K; otherwise, the tuples of the clustering feature tree will be mergedaccording to the input K value combined by distance. The BIRCH algorithm has fast clustering speed. It only needs to scan the training set once to build a clustering feature tree and identify noise points, but it does not perform well on high-dimensional feature data clustering.

3.6.3. Density Clustering. Compared with clustering based on distance calculation, density calculation solves the shortcoming that distance calculation can only find "quasi-circular" clusters. As a density clustering algorithm, DBSCAN is more suitable for convex distribution data than K-means and BIRCH, and it is also suitable for nonconvex distribution data. DBSCAN has the advantages of fast clustering speed, effective processing of noise points, and discovery of spatial clustering of arbitrary shapes, but the DBSCAN algorithm is not a completely stable algorithm.

3.7. Theme Crawler Algorithm. The topic crawler uses the LSTM + CNN classification model to judge the topic relevance of the collected information, and further extracts links from related information pages to further crawl information. The experimental data contains 23,000 pieces of agricultural information collected and 25,000 pieces of Sogou news data. The model structure is shown in Figure 8.

Proceed as follows:

(1) Data input: fixed the matrix parameters of the embedding layer. The parameter is the word vector trained by word2vec. All texts are processed into fixed-length time series data and network input. The embedding layer becomes a two-dimensional matrix. Each row is A word.

(2) Model training: the embedding layer parameters do not participate in model training, and the word2vec obtained is used directly. All the text sequences in the training set are used as the input layer data of the network, and the two-dimensional time series data is converted into the LSTM layer through the embedding layer, and the output of the LSTM layer is used as the input of CNN. The convolutional layer consists of 3 layers of convolution. After the maximum pooling process, it is connected to the 3 layers of fully connected layers. The activation function uses Relu, and finally, the layer containing the softmax activation function is used to output the classification results. Using backpropagation to update the parameters of the entire network, in order to improve the generalization ability of the model and avoid overfitting, some neural connections (Dropout) are randomly discarded [22], and batch normalization (Batch Normalization) [23] is added.

(3) Model verification: use the test set and the trained model for evaluation. The test set is used as the input of the model to compare the real classification label and the classification output of the model to verify the prediction accuracy of the model and related performance parameters.

The evaluation of the model calculated the accuracy rate, recall rate, and $F1$-Score under each experiment and plotted the ROC curve to visualize the classification effect of the classification algorithm. Experiments compare the differences between LSTM + CNN and other classification algorithms and conduct experiments on support vector machines (SVM), polynomial naive Bayes (MultinomialNB), and convolutional neural networks (CNN). SVM uses a linear kernel function for multiple classification. MultinomialNB uses statistics-based classification methods for text classification. CNN has the characteristics of local perception, global sharing, and multiple convolution kernels. The CNN model experiment uses a convolutional layer with a three-layer convolution kernel of 128, a maximum

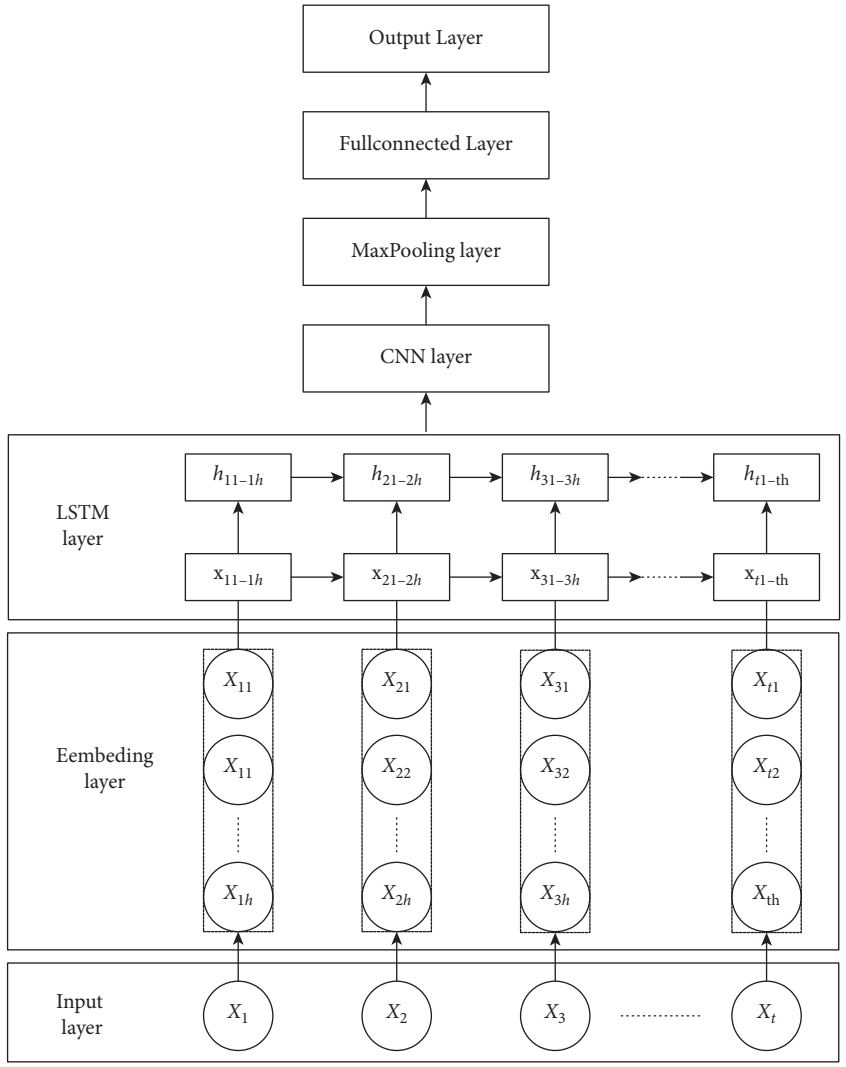

Figure 8: LSTM + CNN model structure.

pooling layer, and a three-layer fully connected layer (RELU activation function). Finally, the softmax layer outputs the classification results. The LSTM model experiment uses a layer of LSTM with 500 units to connect to a 3-layer fully connected layer (RELU activation function) to output the classification results through the softmax layer. The LSTM + CNN model test is fused into the structure of CNN and LSTM [24]. The sequence output of LSTM is used as the input data of CNN for text classification. The classification results are shown in Table 2.

From the data in Table 2 and Figure 9, it can be seen that the method based on CNN-LSTM is superior to traditional SVM and Bayesian in various indicators. The main reason is that tf-idf is used to represent text features in traditional classification. This feature expression does not make full use of contextual information, and part of the information is lost. The simple CNN and LSTM classification methods are not as accurate as the features extracted after the combination of LSTM-CNN in the extraction of information features.

Table 2: The classification of different classification methods (%).

Model	Accuracy
SVM	52.55
Bayes	92.78
CNN	94.86
LSTM	95.61
LSTM + CNN	98.21

3.8. Information Extraction Algorithm. The TextRank model can be expressed as a directed weighted graph $G = (V, E)$, where V is a set of points and E is a set of edges. The weight of the edge between any two points V_i and V_j is w_{ij}. For a given point V_i, $\ln(V_i)$ is the set of points pointing to Vi, $\mathrm{Out}(V_i)$ is the set of points pointed by Vi, and the scoring formula of this point is shown in the following formula:

$$\mathrm{WS}(V_i) = (1-d) + d * \sum_{V_j \in \ln(V_i)} \frac{w_{ij}}{\sum_{V_k \in \mathrm{Out}(V_j)} w_{kj}} \mathrm{WS}(V_j). \tag{29}$$

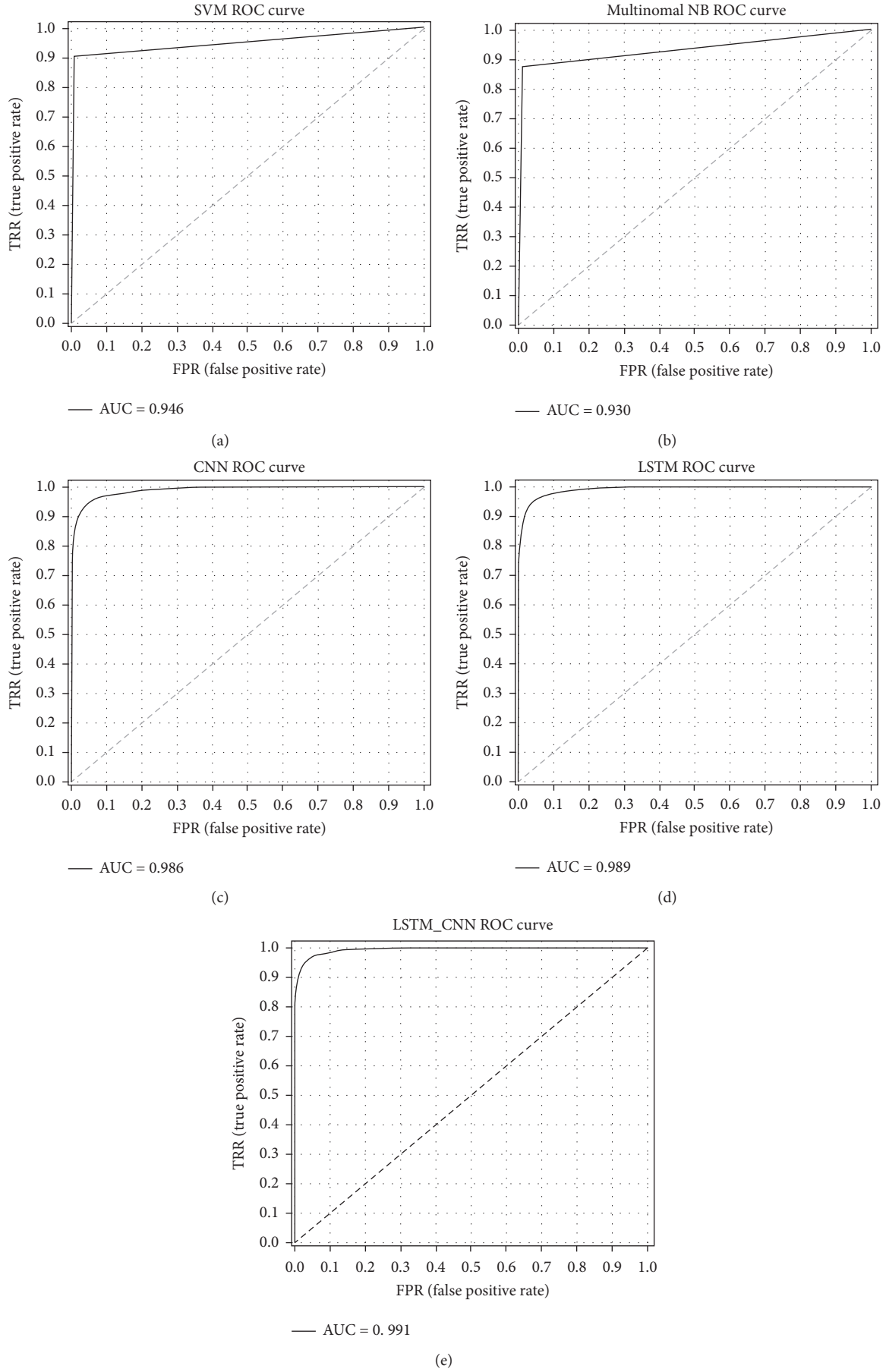

Figure 9: ROC curve.

Among them, d is the damping coefficient, with a value range of 0 to 1, which represents the probability of pointing from a specific point to any other point in the graph and generally takes a value of 0.85. At the beginning, each point has a random initial value, and the Markov transition matrix method is used to recursively calculate until the result is converged (the error is less than the threshold).

The system uses TextRank as the keyword and abstract extraction algorithm. Keyword extraction uses co-occurring vocabulary relations under a certain window to sort words and extract keywords. The main steps are as follows:

(1) Split the text into sentences.

(2) For each sentence, perform word segmentation and part-of-speech tagging, filter stop words, and retain specified part-of-speech words (such as nouns, verbs, and adjectives).

(3) Construct the word graph = (V, E), which is composed of the reserved words in step (2). Then, use the co-occurrence relationship to construct the E-edge set. There are edges between two points only if their corresponding words co-occur in a window of length K. K represents the window size, that is, at most, K words can co-occur.

(4) According to formula (1), iteratively calculate the score of each point until convergence.

(5) Reverse the score of each point to get the most important top words as candidate keywords.

(6) Mark the top candidate keywords in the original text. If adjacent phrases are formed, they are combined into multiword keywords.

Automatic summary extraction based on TextRank forms a summary by selecting sentences with higher importance in the text. The main steps are as follows:

(1) Divide the text into sentences to obtain $T = [S_1, S_2, \ldots, S_m]$, construct a graph $G = (V, E)$, where V is the sentence set, segment the sentence, and remove the stop words, $S_i = [t_{i,1}, t_{i,2}, \ldots, t_{i,n}]$, where $t_i \in S_i$

(2) Construct the edge set E. According to the content coverage between sentences, given two sentences S_i and S_j, the calculation is shown in the following formula:

$$\text{Similarity}\left(S_i, S_j\right) = \frac{\left|\{t_k\}\right|}{\log\left(\left|S_i\right|\right) + \log\left(\left|S_j\right|\right)}, \quad t_k \in t_i \wedge t_j. \tag{30}$$

If the similarity is greater than the set threshold, the two sentences S_i and S_j are considered to be related, and the edge set E is added, and the weight is set to the similarity value

(3) According to formula (1), iteratively solve each sentence score

(4) Reverse the scores and extract the top sentences with the highest importance as candidate abstract sentences

(5) Form a summary of candidate sentences according to requirements

For automatic extraction of keyword and abstract, the "meta" label and "title" label in the information page collected can be refered. The information extraction example is shown in Table 3.

3.9. Topic Detection Algorithm. This paper uses the adaptive incremental K-means clustering algorithm combined with the single-pass algorithm for topic detection. The algorithm steps are as follows:

(1) For each increment, set $N_i\,(i = 1, 2, \ldots, r)$ to determine whether the text S is the first text; if it is, then create the first topic for the text S; if not, compare similarity between text S and other topic centers

(2) According to the similarity between S and each topic, find the topic T with the highest similarity to the text S. The similarity calculation is shown in formula (31) [25]:

$$\text{sim}\left(\overrightarrow{x}, \overrightarrow{y}\right) = \frac{\sum_{i=1}^{m} w_{\chi_i} w_{y_i}}{\sqrt{\sum_{i=1}^{m} w_{x_k}^2} * \sqrt{\sum_{i=1}^{m} w_{y_k}^2}}, \tag{31}$$

where $\text{sim}\left(\overrightarrow{x}, \overrightarrow{y}\right)$ is the similarity between text $\overrightarrow{x}$ and $\overrightarrow{y}$, w_{χ_i} is the weight of feature word i in text $\overrightarrow{x}$, and w_{y_i} is the weight of feature word i in text $\overrightarrow{y}$, m is the total number of words of text $\overrightarrow{x}$ and $\overrightarrow{y}$.

(3) Using the single-pass algorithm, judge whether the similarity between the text S and the topic T is greater than the threshold θ; if it is greater than the threshold θ, the text S is included in the topic T; otherwise, use S to create a new topic and update the topic number K.

(4) Determine the number of texts to be processed for N_i. If it is not 0, continue to step (1) and process the next text. If it is 0, output the number of topics K and the clustering result at this time and continue to the next step.

(5) Calculate the average of K topics as the initial clustering center of the subsequent K-means algorithm.

(6) Calculate the cosine distance between each cluster center and all text according to formula (31), and classify the text to the cluster center with the smallest distance.

(7) Recalculate the mean of each cluster as the new cluster center.

(8) Judge the change between the new cluster center and the previous cluster center; if it is less than the

Table 3: Information extraction example.

Content	With the large number of farmed freshwater fish on the market, the price of Shouguang freshwater fish in Shouguang City, Weifang City, and Shandong Province has dropped overall; among them, the price of carp has fallen sharply; in late August, the price has dropped from 25 yuan per kilogram to 18 yuan per kilogram, a 30% drop; the prices of crucian carp and silver carp also dropped slightly; according to the analysis of professionals, the main reasons for the decline in the price of freshwater fish are as follows: one is the excessively high prices of local freshwater fish in the early stage, and the other is that a large number of freshwater fish farmed in ponds have recently been put on the market; although the price of freshwater fish has dropped overall, it is still generally higher than the same period last year; as the Mid-Autumn Festival approaches, freshwater fish such as carp, grass carp, and silver carp may also experience price increases (source: Shouguang City, Ocean and Fishery Bureau, author: Nongbo Network)
Key words	Price, Freshwater fish, Shouguang City, Silver carp, Carp, Down, Whole, last year, The same period, Author, Pond, Price, Down, Moon Festival, Bureau of Fisheries, Source, Drop by, Approaching, Rise, Appear, analysis.
Key phrase	The price of freshwater fish
Summary	Despite the overall decline in freshwater fish prices, they are still generally higher than the same period last year; as the Mid-Autumn Festival approaches, the prices of carp, grass carp, silver carp, and other freshwater fish may also rise; Shouguang freshwater fish prices fell overall

threshold M, proceed to the next step; otherwise, iteratively calculate (6) and (7) according to the new cluster center.

(9) Judge the increment number, which is 0; the algorithm ends and outputs the topic number K and the clustering result. Otherwise, go back to step (1) and process the next incremental text.

In step (2) of the above algorithm, the similarity between the text S and each topic is calculated, and a valid text is selected from each topic as the representative of the topic. The calculation is shown in the following formula:

$$d = \max\left\{\sum_{\substack{i \neq j \\ j=1}}^{M_k} \operatorname{sim}\left(d_i, d_j\right)\right\}, \quad d_i \text{ and } d_j \in C_k. \tag{32}$$

Select the effective text of the text composition topic with the largest average similarity from each topic, C_k is the current topic set, and M_k is the number of current topic texts. At the same time, there will be a certain degree of similarity between topics. The similarity between topics can be detected for related topic drift. The similarity between topics is calculated by the following formula [26]:

$$\operatorname{sim}\left(C_i, C_j\right) = \max\left\{\operatorname{sim}\left(d_{C_i}, d_{C_j}\right)\right\}, \quad d_{C_i} \in C_i \text{ and } d_{C_j} \in C_j. \tag{33}$$

The similarity is calculated for the text sets in each two topics, and the maximum similarity is taken as the similarity between the topics.

3.10. Topic Tracking Algorithm. In this paper, K-nearest neighbor (abbreviated as KNN) is used, and some improvements are made on the original basis. KNN compares and selects the nearest K known topic texts related to the classified text according to the topics of the K texts. To determine the subject of the text to be classified, the algorithm steps are as follows:

(1) Calculate the similarity between the text to be tracked and the effective text of a known topic. See formula (4) for effective text selection and formula (3) for similarity calculation. Select the K topics with the highest similarity.

(2) Calculate the similarity between all the texts of K topics and the text to be tracked, select the K texts with the highest similarity, and calculate the average similarity of the K texts in the unit of topic.

(3) The maximum average similarity ≥ the threshold ρ, and it is determined that the text to be tracked belongs to this topic.

3.11. Sentiment Analysis Algorithm. This paper adopts the method based on sentiment dictionary to detect sentiment tendency. The construction of sentiment dictionary is a complicated and arduous task. This article adopts the combination of HowNet sentiment dictionary and NTU built by Taiwan University as the basic sentiment dictionary and adds the basic sentiment dictionary to users. The dictionary is used for word segmentation and necessary expansion and improvement. Using Word2Vec and basic emotional dictionary to build a dictionary, the construction process is shown in Figure 10 [27, 28].

The main steps of constructing an emotional dictionary are as follows:

(1) To retain emotionally inclined words in the corpus, here retain adjectives and adverbs as candidate emotional words.

(2) Construct the word vector of the corpus based on the word2vec word vector calculation tool, and obtain the 10 words closest to the candidate word.

(3) Judge whether all 10 similar words cannot be found in the basic emotional vocabulary, and none of them can jump to step (5); otherwise, proceed to step (4).

(4) Determine the emotional tendency according to the semantic similarity between the candidate emotional word and the commendatory emotional word

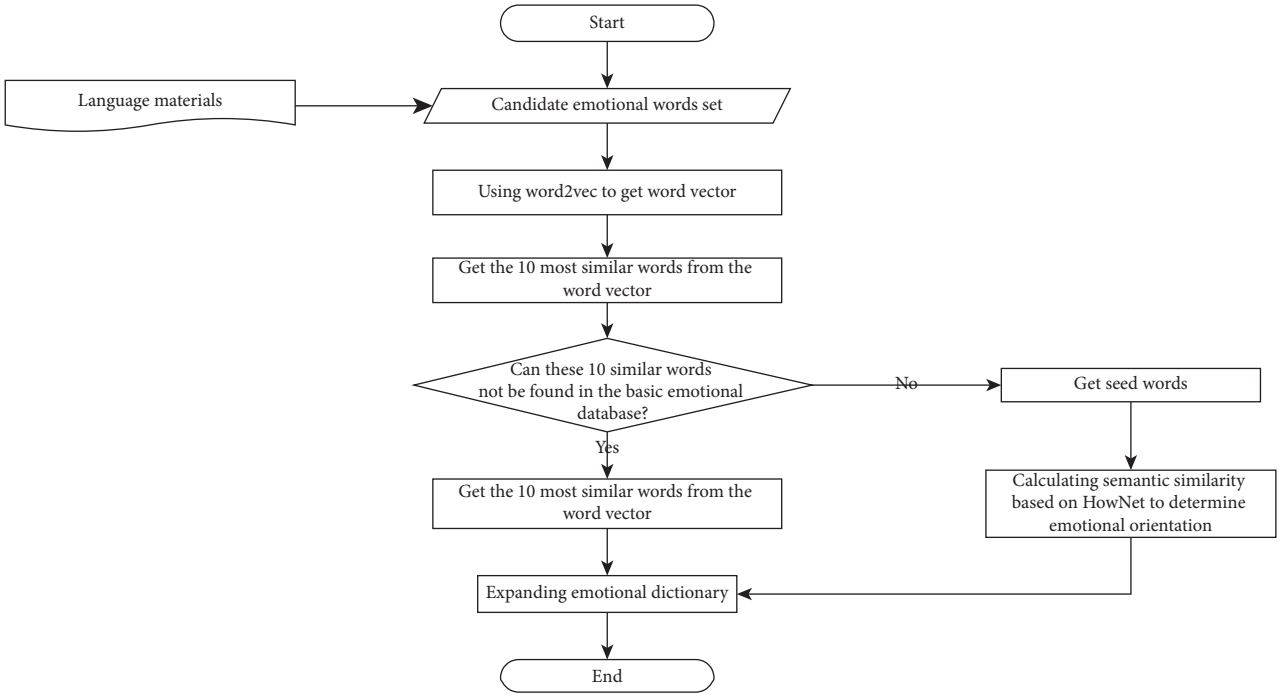

Figure 10: Emotion dictionary construction flow chart.

among the 10 similar words. The calculation is shown in the following formula:

$$O(\text{word}) = m\sum_{i=1}^{m}\text{sim}(\text{word}, P\text{word}_i) - n\sum_{j=1}^{n}\text{sim}(\text{word}, N\text{word}_j). \tag{34}$$

Among them, sim(word1, word2) uses the word vector of word1 and word2 to calculate the cosine value as the similarity, and the calculation formula is shown in formula (34), and *P*word represents the praise word, *N*word represents the derogatory word, and $O(\text{word}) > 0$ is the candidate word, and it is a commendatory word, and $O(\text{word}) < 0$ means the word is a derogatory word.

(5) Select 15% of the commendatory and derogatory words with obvious and strong emotional tendency as seed words from the basic emotional vocabulary, and use the HowNet tool to calculate the semantic similarity between the candidate words and the seed words to determine the emotional tendency:

$$O(\text{word}) = \frac{1}{m}\sum_{i=1}^{m}\text{sin}(\text{word}, P\text{word}_i) - \frac{1}{n}\sum_{j=1}^{n}\text{sim}(\text{word}, N\text{word}_j). \tag{35}$$

Among them, sim(word1, word2) is the semantic similarity calculated by the HowNet tool, and *P*word represents the praise word, *N*word represents the derogatory word, $O(\text{word}) > 0$ means the candidate word is the praise word, and $O(\text{word}) < 0$ means the word is a derogatory term.

Emotional words are assigned, positive emotional words have a score of 1, negative emotional words have a score of −1, neutral words are 0, degree adverbs are based on the score given in the emotional dictionary, and negative words are all set to −1. Sum up the sentiment weights of all words in the text; if the score obtained is greater than 0, it is a positive sentiment. If the score is less than 0, it is a negative emotion. If the score is 0, it is a neutral emotion.

With the increase of sentiment annotation data, the sentiment judgment of public opinion information is realized by constructing a text sentiment classification model. Sentiment classification is different from domain classification. The general feature extraction algorithm in domain text classification can play a very good classification effect, but it has its own independent characteristics in sentiment classification, and the general text feature extraction algorithm cannot play a good effect. The features that can be selected in sentiment classification include sentiment words, negative words, transition words, and degree adverbs. See Table 4 for specific descriptions, Table 5 for dictionaries, and Table 6 for negative dictionary, turning dictionary, and degree adverb dictionary.

Aiming at the analysis of agricultural product network public opinion information and the large amount of information on the Internet, this paper proposes the design and implementation of a platform for agricultural public

Table 4: Emotion words in emotion dictionary (part).

Types	Emotional words
Positive emotion words	Love, be ashamed, tireless, gratified, praise, understand, support, and look forward to
Negative emotion words	Dissatisfaction, disappointment, waste, threat, evil, harm, fear, crisis, vulgarity, and nausea

Table 5: Classification characteristics of emotion.

Feature number	Feature content	Description
1	Positive and negative emotion words	Positive emotion words and negative emotion words
2	Negative Words	Words with negative meaning
3	Turning words	Words with turning meaning
4	Adverbs of degree	Adverbs describing degree
5	Part of speech	Part of speech of emotional words
6	Emotional punctuation	! and?

Table 6: Negative dictionary, turning dictionary, and degree adverb dictionary (part).

Dictionary type words	Dictionary type words
Negative dictionary: no, no, no, and five Transition dictionary: return, but, instead, but, and yet Degree adverb dictionary: 100%, extreme, absolute, extremely, very, especially, almost, slightly, extra, more, more, a little, a little, too, especially, and very	Negative dictionary: no, no, no, and five Transition dictionary: return, but, instead, but, and yet Degree adverb dictionary: 100%, extreme, absolute, extremely, very, especially, almost, slightly, extra, more, more, a little, a little, too, especially, and very

opinion data collection and monitoring system based on big data technology. The system can collect large-scale data, expand collection sites flexibly, perform preliminary natural language processing on the collected data in real time and import it into the database.It realizes the recognition and tracking of public opinion topics, realizes the early warning of public opinion information based on emotional polarity calculation and keyword monitoring, and visually displays the data.

4. Summary

(1) First, analyze the current status of online public opinion under the current development of the network environment, further analyze the current status of agricultural public opinion, and elaborate on the importance of effective monitoring of agricultural online public opinion and the relevant background conditions of online public opinion research at home and abroad

(2) Introduce related technologies such as Hadoop, Spark computing model, HBase database, Solr file retrieval service, and Scrapy-Redis distributed crawler in the big data ecological environment

(3) System demand analysis and nonfunctional demand analysis: design and explain the physical structure and technical structure of the system, hierarchically design functional modules, and design HBase and MySQL public opinion system databases

(4) Introduce the basic algorithms of text processing, text classification algorithms, and text clustering algorithms, and improve the algorithms in topic detection and tracking in public opinion analysis and sentiment analysis tasks

5. Conclusion

(1) Experimental comparison shows that word2vec has a stronger expression of text semantics and, at the same time, solves the high-dimensional sparse problem of tfidf and bow vectors, and bow text vectors perform poorly

(2) The research is concluded that the classification of agricultural product network public opinion information based on CNN-LSTM is superior to traditional SVM and Bayesian

(3) The solution in this paper can meet the user's requirements for the monitoring of network public opinion for agricultural products

There are several shortcomings in the research of this paper that need to be improved. For example, the website's anticrawling strategy and dynamic loading technology prevent the crawler to crawl information. The accuracy of algorithm analysis has been improved, but there are still errors, which can be further improved.

Data Availability

No data were used to support the findings of the study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Project of Science and Technology Department of Jilin Province (20190303035SF), Changchun Municipal Science and Technology Bureau

Project (20170101051JC), Project of Education Department of Jilin Province (JJKH20190923KJ), Changchun Science and Technology Plan Project Science and Technology Innovation "Double Tenth Project" Major Science and Technology Project (18SS018), and Science and Technology Development Program of Jilin Province (20190301024NY).

References

[1] China Internet Network Information Center, *44 Times China Internet Network Development State Statistic Report*, China Internet Network Information Center, Beijing, China, 2019.

[2] M. Lin, *Network Public Opinion: Research on Influencing Factors and Their Action Mechanism*, Zhejiang University, Zhejiang, China, 2013.

[3] X. Li and Y. Deng, "Analysis of national mentality on quality and safety of edible agricultural products-taking the public opinion event of "excessive strawberry residue causes cancer" in Beijing as an example," *Chinese Journal of Food and Nutrition*, vol. 21, no. 6, pp. 5–9, 2015.

[4] H. Huang, M. Huang, W. Zhang, S. Pospisil, and T. Wu, "Experimental investigation on rehabilitation of corroded RC columns with BSP and HPFL under combined loadings," *Journal of Structural Engineering*, vol. 146, no. 8, Article ID 04020157, 2020.

[5] Z. Xiong, N. Xiao, and F. Xu, "An equivalent exchange based data forwarding incentive scheme for socially aware networks," *Journal of Signal Processing Systems*, vol. 93, no. 1, pp. 1–15, 2021.

[6] J. Zhao, J. Liu, and J. Jiang, "Efficient deployment with geometric analysis for mmWave UAV communications," *IEEE Wireless Communications Letters*, vol. 9, no. 7, pp. 1115–1119, 2020.

[7] X. Li, Y. Qian, and Y. Deng, "Monitoring and analysis on internet public opinion of agro-products quality and safety in China, 2016," *Chinese Science Bulletin*, vol. 62, no. 11, pp. 1095–1102, 2017.

[8] B. Ling, "The first half of 2013 the quality and safety of agricultural products network public opinion survey profile," 2013, http://yuqing.people.com.cn/n/2013/0813/c364391-22540650.html.

[9] Y. Chen, W. Zheng, W. Li, and Y. Huang, "Large group Activity security risk assessment and risk early warning based on random forest algorithm," *Pattern Recognition Letters*, vol. 144, 2021.

[10] Z. Baozhong, X. Di, L. Yu, and L. Fusheng, "Multi-scale evapotranspiration of summer maize and the controlling meteorological factors in north China," *Agricultural and Forest Meteorology*, vol. 216, pp. 1–12, 2016.

[11] L. Guo, Y. Qi, Li Yan, Y. Lian, and X. Li, "Reflections on network public opinion monitoring of agricultural product quality and safety," *China Food and Nutrition*, vol. 18, no. 12, pp. 5–7, 2012.

[12] X. Xu and Y. Lai, "Risk monitoring and analysis of agricultural product quality safety network public opinion," *Journal of Fujian Administration University*, vol. 4, pp. 95–100, 2014.

[13] J. Hong and X. Ma, "On the collection, analysis and guidance of online public opinions," *Journal of Huazhong University of Science and Technology (Social Sciences Edition)*, vol. 6, pp. 104–107, 2007.

[14] Y. Kim, "Convolutional neural networks for sentence classification," 2014, https://arxiv.org/abs/1408.5882.link.

[15] Ji Dan and Y. Xie, "Historical review and reflection of Chinese online public opinion research -- based on the observation of CNKI and CSSCI highly cited papers," *Journal of Shanghai Jiao Tong University (Philosophy and Social Sciences Edition)*, vol. 20, no. 4, pp. 48–56, 2012.

[16] A Xinge, "Research summary of domestic and foreign public opinion," *Journal of Library Science*, vol. 33, no. 9, pp. 140–142, 2011.

[17] Y. Fu and X. Zheng, "Review and prospect of research on network public opinion at home and abroad," *Friends of the Editors*, vol. 12, pp. 56–58, 2013.

[18] T. White, *Hadoop's Authoritative Guide: Storage and Analysis of Big Data*, Tsinghua University Press, Beijing, China, 2017.

[19] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," in *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, pp. 29–43, New York, NY, USA, 2003.

[20] R. Lämmel, "Google's MapReduce programming model-Revisited," *Science of Computer Programming*, vol. 70, no. 1, pp. 1–30, 2008.

[21] C. Han, B. Zhang, H. Chen, Z. Wei, and Y. Liu, "Spatially distributed crop model based on remote sensing," *Agricultural Water Management*, vol. 218, pp. 165–173, 2019.

[22] W. Zhang, Y. Hu, J. Liu et al., "Progress of ethylene action mechanism and its application on plant type formation in crops," *Saudi Journal of Biological Sciences*, vol. 27, no. 6, pp. 1667–1673, 2020.

[23] H. Zengyong, *Design and Implementation of User Behavior Analysis System Based on Hadoop*, Beijing Jiaotong University, 2014.

[24] D. Ruoyao, "*Research on the discovery and tracking of internet topics based on blogs*," Doctoral dissertation, Beijing Jiaotong University, Beijing, China, 2011.

[25] Z. Tang, *Design and Implementation of Hadoop Based Recommendation System*, University of Electronic Science and Technology, Chengdu, China, 2013.

[26] J. Song, Q. Zhong, W. Wang, C. Su, Z. Tan, and Y. Liu, "FPDP: Flexible privacy-preserving data publishing scheme for smart agriculture," *IEEE Sensors Journal*, vol. 99, p. 1, 2020.

[27] Z. Lv, Y. Han, and A. K. Singh, "Trustworthiness in industrial IoT systems based on artificial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 99, p. 1, 2020.

[28] Z. Lv, L. Qiao, J. Li, and H. Song, "Deep learning enabled security issues in the internet of things," *IEEE Internet of Things Journal*, vol. 99, p. 1, 2020.