Hindawi
Complexity
Volume 2022, Article ID 9031900, 15 pages
https://doi.org/10.1155/2022/9031900

WILEY | Hindawi

Research Article

Stock Price Prediction Based on Natural Language Processing[1]

Xiaobin Tang,[1] Nuo Lei,[1] Manru Dong,[1] and Dan Ma[2]

[1]*School of Statistics, University of International Business and Economics, Beijing 100029, China*
[2]*School of Statistics, Southwestern University of Finance and Economics, Chengdu 610071, Sichuan, China*

Correspondence should be addressed to Dan Ma; 219020208012@smail.swufe.edu.cn

Received 1 November 2021; Revised 13 January 2022; Accepted 25 February 2022; Published 6 May 2022

Academic Editor: Atila Bueno

The keywords used in traditional stock price prediction are mainly based on literature and experience. This study designs a new text mining method for keywords augmentation based on natural language processing models including Bidirectional Encoder Representation from Transformers (BERT) and Neural Contextualized Representation for Chinese Language Understanding (NEZHA) natural language processing models. The BERT vectorization and the NEZHA keyword discrimination models extend the seed keywords from two dimensions of similarity and importance, respectively, thus constructing the keyword thesaurus for stock price prediction. Furthermore, the predictive ability of seed words and our generated words are compared by the LSTM model, taking the CSI 300 as an example. The result shows that, compared with seed keywords, the search indexes of extracted words have higher correlations with CSI 300 and can improve its forecasting performance. Therefore, the keywords augmentation model designed in this study is helpful to provide references for other variable expansion in financial time series forecasting.

1. Introduction

The stock market is a barometer of the macroeconomy, which reflects many investors' expectations on the market for future economic conditions. With China's financial market's continuous reform and gradual opening, the stock market plays an increasingly important role in the national economy. Since the stock market has important functions such as resource allocation, economic adjustment, and price discovery, and is closely related to CPI, interest rate, and other indicators, the stock market index has an important reference value for the government's macroeconomic policy and the central bank's monetary policy; therefore, it has always been the focus of academic and industrial research.

The research on stock market price prediction has a long history. Although Fama [1] has developed the efficient market hypothesis, indicating that under ideal conditions, information in the past has been fully reflected in the share price, thus stock price can only be affected by newly emerged information. But due to its harsh assumption, the theory is always challenged by other researchers. In the market, fundamental analysis, technical analysis, quantitative analysis and other methods still occupy a place in active investment. With the rise of behavioral finance, people gradually realize that irrational behavior in the market is widespread. For example, psychological characteristics such as the herd effect make a piece of news in the market likely to lead to drastic fluctuations in the stock market; therefore, it is possible to analyze network public opinion data by statistical methods and then predict the stock market price. With our proposed keyword augmentation strategy based on Bidirectional Encoder Representation from Transformers (BERT) and Neural Contextualized Representation for Chinese Language Understanding (NEZHA), financial institutions, for example, can acquire more timely time series by web search index and improve their risk management strategy to address evolving market fluctuation.

The structure of this study is as follows: Section 2 introduces the development of natural language processing and stock-related literature. Section 3 introduces the basic model and algorithm used in this study. Section 4 introduces the framework of stock prediction method designed in this study. Section 5 is the experimental research on the prediction of CSI 300 stock index through empirical research, and Section 6 gives the conclusion.

2. Related Work

The prediction of the stock price trend has always been studied by scholars. The existing research model of stock prediction is mainly reflected in two aspects. On the one hand, traditional econometric models are used, such as regression model and ARIMA under the framework of least squares, because of a series of constraints and nonlinear data that cannot be well dealt with, and the performance effect of the model is limited [2–4]. On the other hand, machine learning and deep learning models should be improved and used. The predictors are common features of stock data (open and volume, etc.) to establish a stable and high-precision prediction model [5–7]. In terms of data types of prediction targets, stock prediction can be divided into classified predictions based on the rise and fall of stocks [8–10] and regression predictions based on stock time series data [11–13]. The difference lies in whether the data types of prediction targets are discrete or continuous, and this study belongs to the latter type.

Scholars have made remarkable achievements in stock price prediction. Still, the common feature of existing literature is to improve prediction methods to improve prediction accuracy, and there are the following deficiencies in feature selection: (1) Although predictors are widely used, the selection of predictors mostly relies on literature and empirical intuition, and there is no relatively scientific measurement standard. Because the selection of keywords is affected by subjective factors to some extent, it is inevitable to miss important keywords due to the limited selection range. However, if the keyword index set as a predictive variable is selected improperly, it will affect the accuracy of stock price prediction to a great extent. (2). The former Natural Language Processing (NLP) vectorization technique is not sufficient in semantic recognition and understanding, which is easy to cause information loss, thus leading to the deterioration of the quality of the vocabulary expansion of predictive variables. For example, the average of word vector ignores the importance of word order and semantics, resulting in information loss. The vectorized Word2Vec model, which maps words to fixed vectors, cannot take context into account in terms of word association and lacks generalization representation ability.

NLP aims to understand and dig out the connotation of human text language by computer. It is an efficient way to analyze a large amount of network text data. From statistical language models to deep learning language models, the models' ability to represent natural language texts is constantly improving and even exceeds human representation in some areas. The statistical language model mainly extracts keywords based on word frequency and subject word distribution [14–17]. With the development of computer's computing power, the deep learning language model based on large-scale neural networks has been realized. Compared with the traditional statistical language model, it has a stronger text mining ability. The BERT model proposed by Google improves the static representation of the Word2vec algorithm [18], integrates the advantages of ELMo model and GPT model to distinguish polysemic words and parallel pretraining [19, 20], and conducts pretraining through an in-depth bi-directional transformer structure. Then the BERT model can realize the word representation integrating context semantics [21]. Based on the BERT model, the NEZHA model (Wei et al., 2019) [22] adopted Whole Word Masking (WWM) and other technologies to improve Chinese text features and achieved the SOTA effect in a number of Chinese natural language tasks. Existing literature shows that BERT shows strong semantic recognition ability from different perspectives in text classification, machine translation, q&A, and other tasks; therefore, this study adopts the BERT and NEZHA models to realize the seed keyword expansion task [23–25].

For predicting missing data, Kong et al. proposed a novel multitype health data privacy-aware prediction approach based on locality-sensitive hashing [26]. With the advent of the era of big data, the emergence of search engines provides more and more quantitative data for network public opinion analysis. Among them, the keyword web search index is widely used in the research of stock price prediction due to its features of intuitive data form, fast update speed, and strong timeliness. The current research mainly innovates on the forecasting method based on the web search index [27–30], which also provides ideas for the research of this study.

With the continuous improvement and development of deep learning technology in machine learning, LSTM can automatically search nonlinear features and complex patterns in data, and it shows excellent predictive performance in practical application research. For example, in the study of portfolio application, Fischer and Krauss (2018) compared with other prediction models, the portfolio constructed based on LSTM can obtain better investment performance [31]. Li Bin et al. (2019) constructed a stock return prediction model in fundamental quantitative investment by using cyclic neural network and long- and short-term memory networks and other technologies, and the results show that the LSTM model is significantly superior to the traditional linear algorithm in identifying the complex relationship between anomaly factor prediction and excess return [32]. Liu et al. showed that LSTM could capture the relationship between historical climate data, which has good practicability for predicting greenhouse climate [33]. Mehtab, Baek et al.'s research also shows that the deep learning LSTM model has outstanding performance in stock prediction [34, 35].

Based on the above analysis, the following research methods are presented in this study. First, based on the seed-word database summarized in the existing literature, crawler technology, and search engine are adopted to capture the web text related to the stock price as the text database, and a large number of keywords are obtained after word segmentation. Second, the BERT model is used to represent the word vectorization and calculate the word similarity to conduct preliminary screening, and then the potential predictive variable keywords are extended. Then, the NEZHA model with better performance under the Mindspore framework is selected to finetune the keyword data set and obtain the importance of words in combination with the context to screen out the predictive word variables and further expand the predictive

variable keywords with higher quality. Finally, this study uses a machine learning LSTM prediction model to empirically test the set of predicted variables obtained and compares and analyzes the prediction effect of the model before and after the expansion of the set of variables.

3. Model and Algorithm

3.1. JIEBA Word Segmentation Algorithm. JIEBA word segmentation algorithm is an efficient sentence segmentation algorithm for Chinese. Compared with English, there is no obvious separation mark between Chinese words; so word segmentation algorithms are particularly important in Chinese semantic analysis. The word segmentation principle of the JIEBA word segmentation algorithm mainly includes the following three parts [36].

3.1.1. Generate All Possible DAG in the Sentence Based on the Prefix Dictionary. The JIEBA algorithm uses the data structure of Trie to store more than 300,000 common Chinese words. The prefix tree saves a large number of words in a tree-like path, concatenating words starting from the root node. Compared with the traditional hash table, it has the advantages of high efficiency and fast speed in the task of searching Chinese words.

According to the above prefix dictionary, the JIEBA algorithm abstracts all possible segmentation of a Chinese sentence into a directed acyclic graph (DAG) and records the word frequency of the training sample in Trie to further determine the most likely segmentation combination.

3.1.2. Use DP to Find the Most Probable Path and Segmentation Based on Word Frequency. In all DAGs, dynamic programming (DP) can be used to find the maximum probability path based on the word frequency in the sample. Set Path = $(\text{node}_1, \text{node}_2, \ldots, \text{node}_n)$. The goal of our programming is

$$\max \sum_i \text{weight}(\text{node}_i). \tag{1}$$

Where node_i represents each node where we possibly separate the sentence. $\text{weight}(\text{node}_i)$ represents the probability, which is represented by the frequency of the word in the corpus, of the from another node to the present node. We link these nodes together to make sure we get the most possible segmentation of the sentence. Let the route with the greatest probability be $P_{\max}$. In practice, we finds the most possible path in reverse. For node_x, there are of nodes behind such as $\text{node}_i, \text{node}_j, \text{node}_n$. Assume that the maximum split routes to reach the previous node are within $P_{\max i}$, $P_{\max j}$, $P_{\max n}$, etc. We can get the state transition equation in DP:

$$P_{\max x} = \max\left(P_{\max i}, P_{\max j}, \ldots P_{\max n}\right) + \text{weight}(\text{node}_x). \tag{2}$$

By solving this DP problem, we can find the path with maximum probability.

3.1.3. Use HMM and Viterbi Algorithm to Infer Uncollected Words. Suppose there are four hidden states of BEMS for each Chinese character in a Chinese vocabulary, namely B-Beging, E-End, M-Middle, and S-Single. The JIEBA algorithm uses Hidden Markov Model (HMM) to infer the hidden state chain of unlisted words. The conversion probability of the hidden Markov chain at each position has been stored in the above prefix dictionary, and the target sentence has provided a visible state chain. Therefore, the Viterbi algorithm is used to solve the hidden state chain of uncollected words to achieve the purpose of word segmentation.

3.2. NEZHA. The original BERT model was developed by Google. Although it has achieved good training results in English and other texts, it is mainly pretrained for English texts and not optimized for Chinese texts; therefore, there is still a lot of room for improvement. Huawei Noah's Ark Laboratory has developed a model focusing on NEural contextualiZed representation for Chinese lAnguage understanding, which is referred to as NEZHA for short [22].

Compared with the original BERT model, the NEZHA model mainly improved the following four aspects: (1) Using functional relative positional encoding that is conducive to the model's understanding of the sequence relationship in the text. (2) In the pretrained MLM task, the WWM skill is used, combined with JIEBA word segmentation. If a Chinese character is covered, other Chinese characters that belong to the same word as the Chinese character in the sentence will also be covered. Although the improvement increases the difficulty of model pretraining, it helps the model to better understand the information on the word dimension of Chinese text. (3) Using the mixed-precision training method, the data are reduced from FP32-bit to FP16-bit in the gradient calculation process, thereby reducing the volume of model parameters and speeding up the training. (4) Use Layer-wise Adaptive Moments optimizer for Batching (LAMB) training optimizer to optimize model training, shorten training time, adaptively adjust the learning rate when the batch size is large, and maintain the accuracy of gradient update. Therefore, this article uses the BERT model to initially select the matched derived keywords and then employ the NEZHA model to extract keywords from the related stock price text captured on the network.

Since NEZHA is an improved model based on BERT, we first introduce the BERT model structure based on the research of the Devlin et al [21]. Bidirectional Encoder Representations from Transformer (BERT) is a bidirectional representation encoder based on Transformer. Compared with the traditional RNN-based natural language processing model, BERT has the following advantages: (1) Using the encoder from Transformer as the model's basic structure, parallel training can be carried out, thereby improving the overall training speed of the model. (2) Compared with other generative models that also use the Transformer structure for pretraining (such as OpenAI GPT), the BERT model uses bidirectional representation for pretraining to better understand the context information token-level tasks.

The BERT model broke the records of many text understanding tasks, which is inseparable from the structure of the BERT model. The NEZHA model and the BERT model have almost the same model structure, both using the encoder part of the Transformer structure to process the input text through the stacked multihead self-attention mechanism and the fully connected network. In the Transformer structure, the embedding feature of the input text is the vector sum of the three vectors, including token embedding, segment embedding, and positional embedding. The NEZHA and BERT models have the same performance in word embedding and segment embedding. However, in terms of position embedding coding, NEZHA encodes the absolute position of BERT and improves it to functional relative position coding, which is conducive to the model's understanding of the sequential relationship in the text.

The encoder part of Transformer contains six layers, and each layer includes two sublayers, namely Multi-Head Self-Attention and Feed-Forward Network (FFN). There is a residual connection mechanism and a layer normalization mechanism between each sublayer to prevent gradient dispersion and explosion.

The self-attention mechanism is the key of NEZHA and BERT models for mining text semantics. By calculating the attention score to weight the original embedding, the attention mechanism can allow the language model to learn the dependencies between texts from a distance. At the same time, a multihead attention mechanism is formed by stacking multiple attention modules. The model can extract relevant information from different representation subspaces at different positions. Aiming at the keyword expansion demand in the stock price prediction problem, this mechanism can effectively learn the deep semantics of the keywords in the original text except for the position information, and then extract high-quality keywords related to stock prediction. The specific principle of the attention mechanism is as follows: First, the model multiplies the original embedding matrix by the corresponding weight matrix to construct three feature matrices of query (Q), key (K), and value (V). Assuming that the embedding matrix of the original text is X, and the corresponding weights to be trained are W_Q, W_K, W_V, the calculation formula of the above matrix is

$$(Q, K, V) = \left(W_Q, W_K, W_V\right)X. \tag{3}$$

Then, the weights are calculated through the query matrix and the key matrix, and normalized with the softmax function, it is weighted with the value matrix V. The specific calculation steps are as follows: First, the matrix Q and the K matrix are multiplied by dot product to calculate the initial attention weight matrix QK^T. In order to prevent the gradient dispersion problem of the Softmax function caused by the excessive value, the initial weight is further scaled to obtain $(QK^T/\sqrt{d_k})$, and then the Softmax function is used to normalize the weight. Finally, the weighted calculation is performed on the value matrix. The overall calculation formula is as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{4}$$

The multihead attention mechanism stacked at the same time can extract text information from multiple subspaces in parallel, so the multiple attention results are spliced and then multiplied by the training matrix W^O. The overall calculation formula of the multihead attention mechanism is as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_h\right)W^o, \tag{5}$$

where the single attention mechanism $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$. *Concat* function represents the splicing of multiple attention heads. The dimensions of each parameter matrix to be trained are $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, $W_i^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

The next fully connected FFN will further refine the calculation results of the multihead self-attention mechanism layer. It contains two linear transformations and an intermediate ReLU activation function. The specific form is as follows:

$$FFN(x) = \max\left(0, xW_1 + b_1\right)W_2 + b_2. \tag{6}$$

The NEZHA and BERT models have added residual connection and normalization processing common in deep networks between the abovementioned multihead self-attention layer and feed-forward neural network layer. They can be used in multilayered to improve the performance of the network; therefore, the output of each sublayer is processed as follows:

$$\text{output} = \text{LayerNorm}(x + \text{Sublayer}(x)). \tag{7}$$

The dimension of the output result is d_{model}; therefore, the basic structure of NEZHA implemented in our experiment is constructed in this study (see Figure 1). In this structure, we especially modify andutilize segment embedding so that the model better distinguishes our input ofkeywords and sentences.

The functional relative positional encoding adopted by the NEZHA model Wei et al. [22] mainly improves the calculation of the self-attention mechanism so that the attention score can take into account the relative positional relationship between the two tokens. Let the sequence of network text input for crawling stocks be $x = (x_1, x_2, \ldots, x_n)$, the output sequence value be $z = (z_1, z_2, \ldots, z_n)$, where $x_i \in \mathbb{R}^{d_x}, z_i \in \mathbb{R}^{d_z}, W^K, W^Q, W^V$ are defined as above. Then the output value is calculated as follows:

$$z_i = \sum_{j=1}^{n} \alpha_{ij}\left(x_j W^V + a_{ij}^V\right), \tag{8}$$

where α_{ij} is the attention score calculated first by scaling the dot product of query matrix Q and key matrix K between position i and position j, and then by the processing of $Softmax$:

$$\begin{aligned} \alpha_{ij} &= \frac{\exp e_{ij}}{\sum_k \exp e_{ik}}, \\ e_{ij} &= \frac{\left(x_i W^Q\right)\left(x_j W^K + a_{ij}^K\right)^T}{\sqrt{d_z}}. \end{aligned} \tag{9}$$

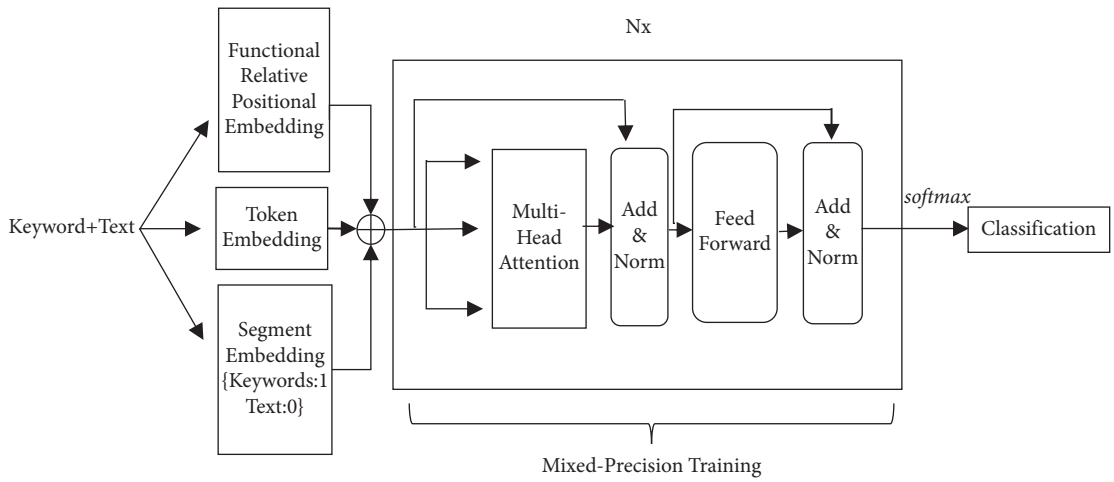

Figure 1: Basic structure of NEZHA model in our experiment. As the core of the model, it can deeply recognize and understand the semantic meaning of the text.

In formula (9), a_{ij}^{K} and a_{ij}^{V} represent the value of functional relative positional encoding. As for the case where the dimension of a_{ij} is $2k$ or $2k+1$, the calculation are as follows:

$$a_{ij}[2k] = \sin\left(\frac{(j-i)}{10000^{(2k/d_z)}}\right),$$
$$a_{ij}[2k+1] = \cos\left(\frac{(j-i)}{10000^{(2k/d_z)}}\right). \tag{10}$$

Under this positional coding rule, the trigonometric function will have different wavelengths in different dimensions, which would help the model learn the information contained in the relative position of the tokens in different dimensions, thus helping to improve the model's performance in downstream tasks.

3.3. LSTM. LSTM is short for Long Short-term Memory. It is mainly improved based on the original RNN in its hidden layer. By introducing Input Gate, Forget Gate, and Output Gate, LSTM can effectively solve the problem that the RNN network cannot capture the long-distance dependence in the long-distance sequence as discussed by Hochreiter and Schmidhuber [37]. This study uses NEZHA model to obtain the keyword and the LSTM model to predict the stock price sequence. LSTM can mine the dependence between the keywords' web search index and the stock price compared with traditional linear models.

The input gate, forget gate, and output gate play different roles in a cell of the LSTM model. Suppose the cell state value at the previous moment is C_{t-1}, the output result of the LSTM at the previous moment is h_{t-1}, and the network input value at the current moment is X_t. The forgetting gate is responsible for controlling the degree to which the state C_{t-1} of the previous period is retained, generating the forgetting threshold vector f_t, and the input gate is responsible for controlling the size of the current network input value X_t, and generating the input threshold vector i_t. The two works together generate the current cell state C_t. After that, the output gate is responsible for outputting the current LSTM output result h_t, with its output threshold vector o_t. Based on Hochreiter and Schmidhuber [37], the specific formulas are as follows:

$$f_t = \sigma\left(W_f \times [h_{t-1}, X_t] + b_f\right),$$
$$i_t = \sigma\left(W_i \times [h_{t-1}, X_t] + b_i\right), \tag{11}$$
$$o_t = \sigma\left(W_o \times [h_{t-1}, X_t] + b_o\right),$$

where W_f, W_i, W_o represent the weight matrix of the forget gate, input gate, and output gate, respectively. b_f, b_i, b_o are the bias matrix. $\sigma(\cdot)$ represents the *Sigmoi d* function.

In the process of calculating the current cell state value C_t, first calculate the intermediate variable $\widetilde{C}_t$ through the activation function $\tanh(\cdot)$ through the current input value X_t and the output value h_{t-1} of the LSTM at the previous moment, and the formula is

$$\widetilde{C}_t = \tanh\left(W_c \times [h_{t-1}, X_t] + b_c\right), \tag{12}$$

where W_c represents the weight matrix corresponding to the intermediate variable $\widetilde{C}$. b_c is the bias matrix, and $\tanh(\cdot)$ represents the tanh activation function. So the calculation formula of the cell state value C_t at time t is

$$C_t = f_t \circ C_{t-1} + i_t \circ \widetilde{C}_t, \tag{13}$$

where $\circ$ stands for dot multiplication.

Thus, the output value h_t of the cell is calculated according to the output gate to complete the calculation inside the cell:

$$h_t = o_t \circ \tanh\left(C_t\right). \tag{14}$$

In summary, the basic cell structure of the LSTM model is summarized in Figure 2.

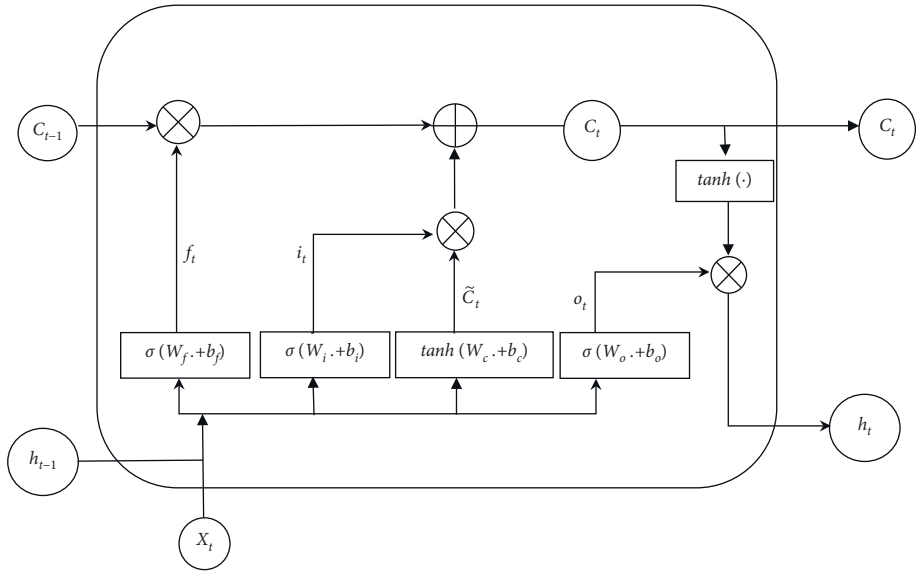

Figure 2: Basic structure of LSTM. In this context, C_t represents the cell state value at time (t); X_t represents the input value at time (t); H_t represents the cell output value at time (t) $\sigma(\cdot)$; and tanh $(\cdot)$ represent the activation function of updated information; (W) and (b) represent the weight matrix and bias of each gate control, respectively.

4. Methodology

Based on our existing seed keywords, this study first collects a large number of web texts related to stock prices through web crawlers. Second, we use JIEBA to segment the relevant texts of seed keywords, thus expanding the keyword vocabulary in terms of quantity and generating possible candidate words after removing the stop words. After that, we use the BERT model to vectorize the words and then calculate their similarity. By constructing (candidate keywords, text) pairs on the keyword data set, we apply the NEZHA model for transfer learning and further finetune it downstream, combining with the context to determine the importance of each word. Consequently, we successfully extract high-quality stock price prediction words. Finally, this study uses LSTM to predict the CSI 300 Index based on seed keywords and generated keywords, respectively. The details of our proposed algorithm are presented in Algorithm 1.

4.1. Pretraining of BERT and NEZHA. As a successful practice of transfer learning in NLP, the BERT and NEZHA models significantly reduce the difficulty of finetuning training by performing two unsupervised pretraining in a large amount of text, thereby achieving leading results in various downstream tasks. Unsupervisedpre-training methods including Masked LM (MLM)and Next Sentence Prediction (NSP) are of greatimportance in this stage [9]. The key to the keyword extraction task in this study is to infer the connection between the keyword and the sentence; therefore, it is necessary not only to dig out the meaning of the text at the word level but also to understand the logical relationship between sentences. Compared with the traditional unidirectional language model trained from left to right, the BERT and NEZHA models, as deep bidirectional network models, can predict the words covered in combination with the meaning of the context, thereby improving the model's sentence-level semantic information learning ability.

In the MLM task, 15% of the word chunks in each sentence sequence are randomly covered, marked as (MASK). The model adds a neural network at the end of the encoder as a classification layer and then uses the Softmax function to convert the output of the network into the predicted probability of each word in the vocabulary. After that, we select the word with the highest probability as the predicted result. Because in 15% of the word blocks that need to be randomly masked, the model only replaces it with (MASK) word block with 80% probability, with a random word with 10% probability, and in 10% situations, the model maintainsthe same word. This ensures that the pretraining can handle sentence without (MASK) chunks. Therefore, the probability of replacing it with a random word only accounts for 1.5% of the full text, which will not have a significant impact on the semantic understanding of the model. To be specific, the NEZHA model adopted the Whole Word Masking method here, which means the model masks not only the single Chinese character butalso other characters belonging to the same Chinese word. This skill helps themodel to better understand Chinese sentence in a more natural way and istherefore beneficial for our keywords extraction.

Compared with the MLM task, which mainly mines the token-level information inside the sentence, the NSP task focuses on understanding the logical connection of the sentence level, so it is very helpful for tasks that focus on text logic, such as question answering (QA) tasks and natural language inference (NLI). In the NSP task, the pretrained texts are sentence A and its next sentences B. Among them, sentence B has a 50% probability of matching the A sentence,

Input initial seed keywords from the literature
Stage 1: BERT word vector similarity selection
(1) Initialize empty similar words vocabulary S
(2) **For** each seed keyword w_i**do**
(3) Collect corresponding baidu baike text
(4) Construct keywords vocabulary V_i based on JIEBA segmentation
(5) Vectorize seed keywords w_i and potential keywords in vocabulary V_i based on BERTvec
(6) **For** each keyword v_j in potential keywords vocabulary V_i**do**
(7) Calculate cosine similarity score sim_{ij} between w_i and v_j
(8) **IF**$\text{sim}_{ij} >$ threshold then
(9) Add v_j to similar words vocabulary S
(10) **End for**
(11) **End for**
(12) Output similar words vocabulary S
Stage 2: NEZHA word importance selection
(13) Initialize empty similar & important vocabulary SI
(14) Collect data from CLUE data set in the form of (keywords, text)
(15) Randomly select words from text as pseudo-keywords at a ratio of 1 : 1
(16) Build finetune data set (Keyword/Pseudo-Keyword, text, label) as F
(17) Construct training set T and development set D from data set F
(18) Finetune BERT-TensorFlow, BERT-MindSpore, NEZHA-MindSpore in training set T
(19) Select the best performing model M (NEZHA-MindSpore) by precision on the development set D
(20) **For** each keyword v_j in similar words vocabulary S**do**
(21) Calculate context importance score I_j based on model M
(22) Add v_j and I_j to similar and important vocabulary SI
(23) **End for**
(24) Keep words with top 100 importance scores in vocabulary SI
Output similar and important vocabulary SI
Stage 3: LSTM stock index forecast
(25) **For** keyword w_k in SI do
(26) **For** lagging term t in 1 to 10 **do**
(27) Calculate lagged search index time series
(28) **End for**
(29) Use Pearson correlation coefficient to select the most related lagged term
(30) **End for**
(30) Train LSTM to forecast CSI300 stock index on the 2215-day train data set
(31) Calculate and compare model RMSE on the 243-day test data set
Output model RMSE

Algorithm 1: Experiment methodology.

which is marked as *IsNext*. In the other 50% cases, sentence B is randomly selected from the corpus and marked as *NotNext*. Since the MLM and NSP models are essentially classification tasks, the cross-entropy function is selected as the loss function; therefore, the overall loss function is obtained by adding and summing the above results. Overall, the training arrangement of textpairs, including sentences with a variety amount of lengths, enables us toprocess the logic connotation between two different pieces of texts, whichmakes it an ideal choice to select keywords from sentences.

Based on the abovementioned pretraining process, the BERT and NEZHA models have been pretrained on a large amount of corpus, thus significantly reducing the training cost of downstream tasks through this transfer learning method; therefore, this study uses the pretraining parameters from Google and Huawei. It enables the BERT model to vectorize the words and the NEZHA model to optimize the training parameters for downstream keyword discrimination.

4.2. BERT Word Vector Similarity Selection. Through a large amount of pretraining, BERT has stronger text representation capabilities as the number of network layers deepens. However, as the number of network layers increases, the output results of each layer of the network, especially the last layer, will be biased toward the pretrained objective function: the MLM task and the NSP task. Therefore, the network output of the penultimate layer is more objective and fairer and is suitable as a representative of word vectors. So in this study, we choose the penultimate network output of BERT as the word vector to represent the meaning of the word after average pooling.

The vectorization selection process uses the BERT model to vectorize the seed keywords and calculates the cosine value between the seed keywords and the candidate keywords to judge the similarity between the words and sort them by values. Then we set a certain threshold, perform preliminary screening of the candidate thesaurus according to the similarity, and keywordscorresponding to high

similarity values are retained (for detailed process, see Figure 3).

4.3. NEZHA Word Importance Selection. In this study, NEZHA model is employed based on existing keywords of stock price prediction, combined with the keyword corpus material in the CLUE data set to finetune the task of identifying keywords [17]. On the one hand, we start from the seed keywords of stock forecasts, collect the Baidu Encyclopedia text corresponding to each keyword, and use the JIEBA to segment and reorganize the encyclopadia text to construct a combination of (candidate keywords, text). Thus, the candidate set of keywords is expanded in breadth. On the other hand, this study integrates the number of news corpus in CLUE, constructs (keywords/pseudo-keywords, text, tags) data sets with the same steps, and performs finetuning training through the NEZHA model to construct keywords selection model. Finally, the finetuned model is used to screen potential keywords, thus filtering the keyword set in depth. The overall tuning process is as follows (Figure 4).

In the data set for English NLP model evaluation, the GLUE data set has been widely accepted and adopted. It has become a standard test data set for evaluating the effects of many NLP models. With the rapid development of the Chinese NLP field, CLUE, a Chinese data set benchmarking similar to GLUE, came into being. The CLUE data set is called the Chinese Language Understanding Evaluation benchmark, which is the first large-scale open-source data set for NLP model benchmark testing in Chinese [38]. To extract keywords for the task of stock price prediction, this study selects the news2016zh data set in CLUE as the training data for downstream finetuning training. The original data set includes (keywords, text) pairs. Using the JIEBA word segmentation tool, this study divides the text and randomly select pseudo keywords that are different from the original keywords of the text. During this process, the ratio of the original keywords to the pseudo keywords is maintained at 1 : 1. Thus, a data set of (keyword/pseudo-keyword, text, label) is constructed for subsequent BERT/NEZHA model training and verification of the classification effect.

For the input (keyword/pseudo-keyword, text) pair, the BERT/NEZHA model encodes it in the same way as in the pretraining to serve as the input vector of the encoder and calculates the output of the numerical vector at the position of (CLS), which contains the encoding representation of the entire sentence. The model attaches a fully connected classification layer to the back end of the encoder. Suppose the parameter matrix of the fully connected layer is W and the output vector at the (CLS) position is C, then the final prediction result Prob is

$$\text{Prob} = \text{Softmax}\left(CW^T\right). \tag{15}$$

Therefore, the cross-entropy loss function is calculated and back-propagated so that all the parameters to be trained in all models are updated end-to-end.

This study builds a model based on the above structure and uses BERT and NEZHA models for training under the Tensorflow framework and the Mindspore framework, respectively. Specifically, it includes three types of models: Bert-Tensorflow, Bert-Mindspore, and NEZHA-Mindspore. The TensorFlow framework is developed and maintained by Google and is adopted by most deep learning models due to its excellent hardware compatibility and visualization ability. However, the static graph operation that Tensorflow has adopted for a long time is conducive to project deployment, but it brings great difficulties to the rapid debugging and iteration of the code. In contrast, the dynamic calculation graph used by frameworks such as Pytorch is very conducive to debugging, but it is difficult to further optimize the performance. The Mindspore framework developed by Huawei takes a different approach and adopts an automatic differentiation method based on source code conversion, which not only brings convenience to model construction but also obtains good performance through static compilation and optimization [39]. We thank MindSpore for the partial support of this work, which is a new deep learning computing framwork [40].

In terms of the hyperparameter selection of the model, most of the parameters in this article are consistent with the default situation. At the same time, to compare the classification effect of each model, the batch size and epoch on the training set, development set, and prediction set are set uniformly. Among them, the batch size of the training set is the largest batch that will not cause Our of Memory (OOM) error in the code test to accelerate model training. At the same time, the training period on the training set is set according to the recommendation of Devlin et al [21]. On the development set and prediction set, the batch size of the model is consistent with the default model with only one epoch. The selection of parameters is as Table 1.

On the training set, this study compares the classification result of different models on the development set under different frameworks so as to select the best model for classification application on the prediction set. The output results of the model on the prediction set are processed by Softmax and used as the words' score of context importance to further screen the words with predictive potential.

4.4. LSTM Stock Index Forecast. We use the LSTM model to empirically predict the stock price based on the web search index of generated word to test the interpretive and predictive ability of the generated words on the stock price. In time series forecasting, proper lag processing of the data helps to accurately describe the relationship between the explained variable and the explanatory variable, thereby improving the forecasting effect. Therefore, this article first performs a certain order of lag processing on the data, uses the Pearson correlation coefficient to screen, and selects the reliable predictor variables with strong correlation (see Figure 5).

For deep learning models such as LSTM, the selection of hyperparameters will greatly affect the model's predictive ability. The parameter setting of LSTM is referred to in the

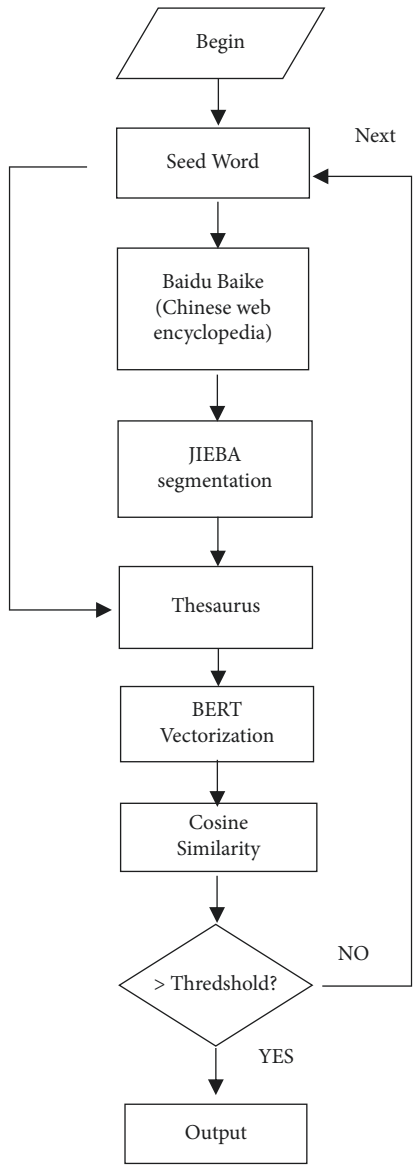

Figure 3: BERT vectorization selection process. We collected related Baike text from seed words and used BERT to perform word vectorization, which makes calculating cosine similarity possible. Finally, we select similar words above the threshold.

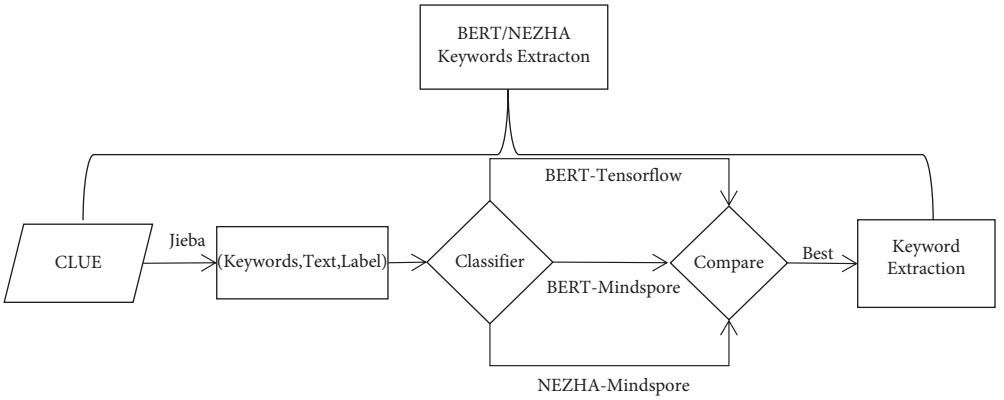

Figure 4: NEZHA importance selection process. We finetuned three different models and compared their performance to decide the best classifier and finally select the top 100 important words.

Table 1: Hyperparameters of NEZHA finetuning task.

Parameter	Train	Dev.	Test
Batch size	128	8	8
Epoch	4	1	1

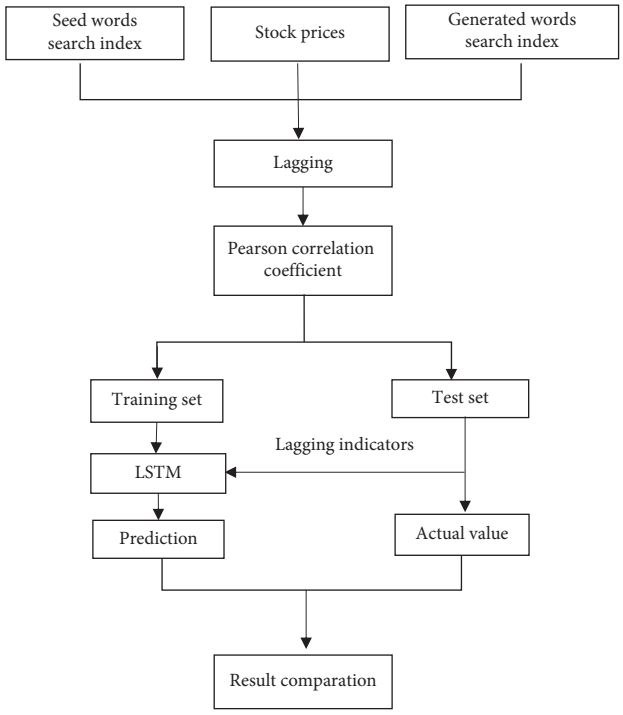

Figure 5: The LSTM Model prediction flow diagram. This figure reflects the comparison process between the predicted values and actual values of the two groups of models.

work of Tang et al. [41], in which the sliding window is set to 30 days, which means the stock price of the next trading day is predicted on the training set by learning the data of the past month. Thenumber of neuron nodes is set to 10, the total number of iterations is 500 epochs, and learning rate is 0.0006. Theoptimizer uses the Adam optimizer. The activation function of each gate is sigmoid,but the activation function of output gate adopts the tanh function, both ofwhich are the default settings of the LSTM.

5. Experiments

5.1. Experimental Data. The CSI 300 index is used as our forecast target. By referring to the existing literature and Baidu index recommendation, we select the seed keywords from the macro and micro aspects, respectively, in Table 2.

On this basis, this study uses the abovementioned vocabulary as search keywords, crawls relevant texts from Baidu Encyclopedia, and filters 19,609 long texts with a length of more than 50 words as corpus. JIEBA segmentation is performed on each text separately, and stop words are removed, thereby constructing a potential predictor variable vocabulary, with a total of 114k candidate words (under different contexts).

5.2. Similarity Selection. Based on the pretraining parameters and BERT vectorization, the potential predictor variable vocabulary related to the stock price is represented in the form of a vector through the multilayer stacked encoder mechanism.Thenthe words are screened from the perspective of similarity, and the semanticallyhighly related words are obtained. This study uses the cosine value between word vectors as a measure of words' similarity and calculates the cosine similarity for each seed keyword of stock price prediction and its corresponding candidate words. The threshold was set to 0.9, and 17,720 potential stock index prediction keywords and corresponding text context were obtained through preliminary screening. Some of the results are shown in Table 3.

5.3. Importance Selection. By calculating the similarity in the BERT vectorization model for preliminary screening, the model efficiently removes many words that have a low correlation with the seed vocabulary predicted by the stock index. Based on this, we introduce the NEZHA model to fuse the context of the candidate keywords and further filter the initial screened words through training of downstream finetune tasks, thereby carefully selecting the keywords according to their context importance.

In this stage, this study uses the news text data set in the CLUE data set. A corresponding number of pseudo keywords are randomly obtained from the text to keep the training sample balanced based on the manually labeled keywords. After that, we generate the standard data set as (text, keyword/pseudo keyword, tag (0 or 1)). In the stage of downstream finetuning, the input of the model is arranged as: [CLS] + text + [SEP] + keywords/pseudo keywords. During the training of the NEZHA model, the input is encoded by word embedding, segment embedding, and position embedding and then calculated by a multilayer encoder to generate the output vector in (CLS). Then we use the backend fully connected classification network structure and Softmax to predict the probability, representing the importance of the keyword in the text.

A total of 534,893 samples are screened in the training set, and a total of 19,609 samples are in the development set. This study trains the BERT-Tensorflow, BERT-Mindspore, and NEZHA-Mindpore models on the training set to compare the performance of the BERT model and NEZHA model in the Tensorflow framework and the Mindspore framework on the development set. Since the goal of this study is to extract keywords with high importance in the task of identifying keywords, the accuracy of the three models are compared, and the calculation formula is as follows:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{16}$$

Among them, TP stands for True Positive, that is, the sample itself is the correct keyword, and the model judges it to be the number of correct keywords; FP stands for False Positive, the sample itself is a pseudo-keyword, and the model judges it to be the number of correct keywords. The

Table 2: Macro and micro seed thesaurus.

Aspects	Seed keywords
Macro	Financial market, bank stocks, economy, inflation, market, stock market, stocks, stock index, stock price, stock market quotations, stock market, securities, securities market, equity, a shares, A-share market, Hong Kong stocks, deposit rates, finance, GDP, CPI, bull market, rate of return, fund company, inflation rate, market conditions, rise, tax, new stocks, daily limit, bar chart, plunge, dollar, bubble, currency, policy, futures, fund company, China news, economic data, bonus, financial network, information disclosure, stock index futures, futures trading, risk management, capital, stock code, asset management, wealth, financial securities, finance, financial news, securities investment funds, Chinese stocks, exchange rates, securities networks.
Micro	Account opening, stock trading, stock recommendation, blue-chip stocks, low-priced stocks, concept stocks, banned stocks, brokerage stocks, stock introduction, stock recommendation, simulated stock trading, bank loans, stock account opening, investment and wealth management, old stockholders, investors, asset management, bankruptcy, arbitrage, financing, insider, income, shorts, speculators, retail, the main force, loans, today's stock market, today's market, restricted stocks, allotments, dark horses

Table 3: Some results of BERT similarity screening.

Seed keywords	Candidate words	Cosine similarity	Results
Financial market	Money funds	0.9212	Keep
	Bond market	0.9489	Keep
	Long-term loan	0.8497	Remove
Policy	Organization	0.9071	Keep
	System	0.9099	Keep
	Country	0.8873	Remove
Lifted stocks	Outstanding shares	0.9216	Keep
	Underweight	0.8900	Remove
	Large-cap stocks	0.8782	Remove
Bank loan	Credit	0.9213	Keep
	Working capital	0.8897	Remove
	Discount rate	0.8540	Remove

performance of the three models in the development set is shown in Table 4.

The performance of three models above verified the performanceof our experiment design. Among them, NEZHA, based on the Mindspore framework, has achieved the best performance in the development set in the keyword discrimination task. This study uses the word importance probability calculated by the NEZHA-Mindspore model as the basis for ranking. Some results of the NEZHA model are shown in Table 5.

This study ranks the abovementioned word importance, selects the top 100 generated words as candidate stock price predictors. Then we use web crawlers to obtain the corresponding Baidu search index. The time interval is set from January 1, 2011, to February 29, 2021. Some of these words were removed due to a small search volume. After deduplication, a total of 61 effective generated words and 87 effective seed words are obtained. The details are in Table 6.

5.4. Predict CSI 300 Index with LSTM. The CSI 300 Index covers the stocks of the Shanghai and Shenzhen exchange in the selection of constituent stocks, and the industry composition is consistent with the market industry distribution ratio; therefore, we choose CSI 300 Index as the object of the empirical test.

Because web search data are affected by public opinion in all aspects, some search data may have a lot of noise, which may affect the prediction ability of LSTM when predicting the CSI 300 index; therefore, this study first uses the Pearson correlation coefficient analysis method to analyze the correlation. Words with rather lower coefficients are removed with an absolute value threshold of 0.6. What is more, the lag order is set to 10. This study selects the lag term with the highest absolute value of the correlation coefficient within the 10-order lag terms of each keyword as the predictor variable. We finally determine the predictive variables by performing the above operations on the seed words and generated words, as shown in Table 7.

The predicted time interval of the CSI 300 Index is set from January 1, 2011 to March 1, 2021. The holidays with no transaction data were filtered out, and a 10-day lagging was performed to obtain a total of 2458 days of valid data. This study uses the 2215-day Baidu search index data before February 29, 2020, as the training set, and the 243-day data from March 1, 2020, to March 1, 2021, as the test set to compare forecasting ability of the seed vocabulary and the generated vocabulary. Among them, the CSI 300 stock index data come from the Wind database, and the keyword data come from the Baidu search index. After LSTM trains the CSI 300 index on the trainingsets of seed word, generates word training sets, respectively, then predicts the test set. Wedid a lot of experiments and found that the RMSE of the generated keywords is lower than the RMSE of the seed keywords in most cases, which demonstrates thestability of our prediction model. Here, we presented one of our experiment result shown in Figures 6 and 7.

Table 4: Performance of BERT/NEZHA model finetuning tasks under different frameworks.

	BERT-TensorFlow (%)	BERT-mindspore (%)	NEZHA-mindspore (%)
Precision	89.86	89.63	90.06

Table 5: Some output results of the NEZHA model.

Micro words	Prob	Macro words	Prob
Allotment payment	0.9998	Stamp duty	0.9784
Bookmaker	0.9991	Spot	0.9398
Catch up	0.9967	Risk	0.9387
Account	0.9905	RMB	0.9184
Fixed assets	0.9873	Floating exchange rate	0.9011
Short position	0.9792	Bullish candlestick	0.8975
Blue-chip stocks	0.9792	Insurance	0.8947
Sell off	0.8598	Securities law	0.7970
Settlement	0.7770	Demand deposit	0.7842
Deposit and loan	0.7455	Taxation	0.7539

Table 6: Macro and micro seed keywords and generated keywords.

Aspects	Seed keyword & generative keyword
Macro	A-share, A-share market, CPI, GDP, K-line chart, rise, China news, Chinese stocks, information disclosure, fund companies, large caps, large cap market conditions, deposit interest rates, yields, policies, new shares, plunges, futures, futures trading, exchange rate, bubble, daily limit, Hong Kong stocks, bull market, taxation, dividends, economy, economic data, US dollar, stock price, stock market, stock market quotations, stock index, stock index futures, equity, stock, stock code, stock market, securities, stock market, securities investment funds, securities networks, wealth, finance, financial news, financial networks, currency, asset management, capital, inflation, inflation rates, finance, financial markets, financial securities, bank stocks, risk management, valuation, public policy, dividends, Growth Enterprises Market, land tax, compound interest, foreign exchange, foreign exchange quotation, dalian commodity exchange, taxation, turnover tax, shenzhen stock exchange, hot money, tax burden, econometrics, surplus, middle price, blowout, interest tax, crash, liquidation, price, nasdaq, stock market crash, delisting, short-selling[2]
Micro	Main force, today's market, today's stock market, low-priced stocks, insider, brokerage stocks, arbitrage, account opening, speculators, investment and financial management, investors, income, retail accounts, concept stocks, simulated stocks, stocks, bankruptcy, shorts, old stocks, stocks introduction, stock account opening, stock recommendation, blue-chip stocks, financing, lifting ban, loans, asset management, allotment, bank loans, restricted stocks, dark horses, Shanghai Composite Index, credit, bankruptcy, borrowing, borrowing (synonym in Chinese), short selling, rebound, hold-up, bookmaker, trading volume, cost, buy at the bottom, investment, foreign exchange, shorting, cancellation, bull stock, profit, consolidation, high-quality stocks, stock reform, lifting of the ban, account, purchase of foreign exchange, capital, repayment, stock selection, allotment payment, heaveweight stock, account cancellation, ex-dividend, ex-rights, annual interest rate, popular stocks, plummeting

Table 7: Predictive variables after correlation coefficient screening.

Data type	Words	Lag	Variable	Corr. Coef.
Seed keywords	CSI 300	1	y_{t-1}	0.9979
	Inflation rate	1	inflation_{t-1}	0.6903
	Chinese news	1	news_{t-1}	–0.6836
	Policy	10	policy_{t-10}	0.6456
	Dark horse	10	dark_horse_{t-10}	0.6238
	Stock quotes	1	quote_{t-1}	0.6130
Generated keywords	CSI 300	1	y_{t-1}	0.9979
	Compound interest	1	compound_{t-1}	0.7296
	Hot money	1	money_{t-1}	0.7096
	Dividend	1	dividend_{t-1}	0.6703
	Profit	1	profit_{t-1}	0.6513
	Annual interest	2	$\text{annual\_interest}_{t-2}$	0.6218

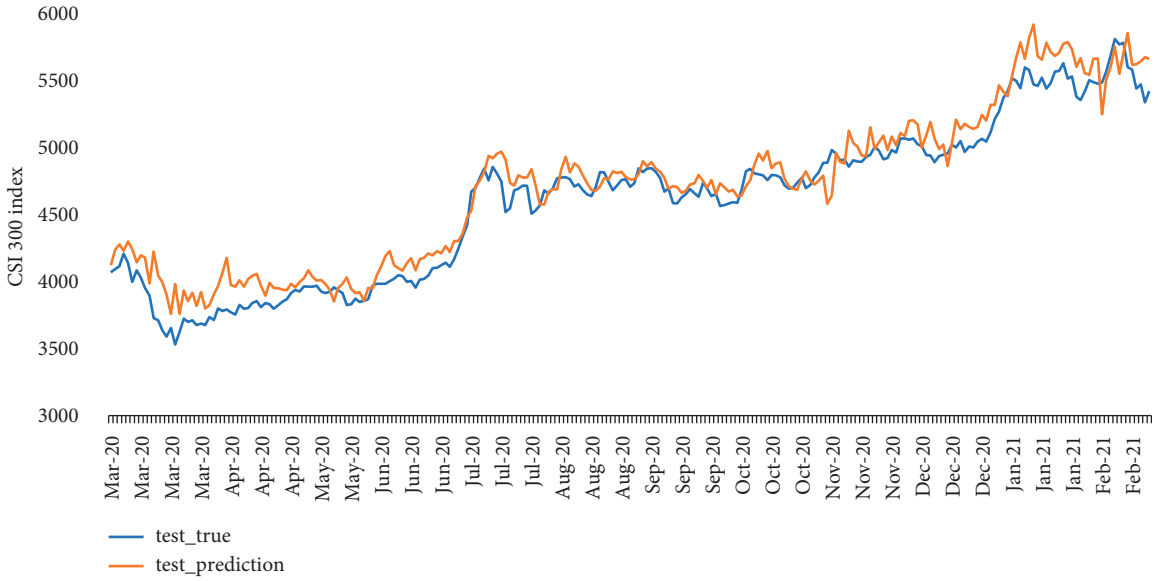

FIGURE 6: The trend chart of the relationship between LSTM model predicted values of seed vocabulary and CSI 300 index true values. The red and blue lines represent the true and predicted values, respectively ($n = 243$).

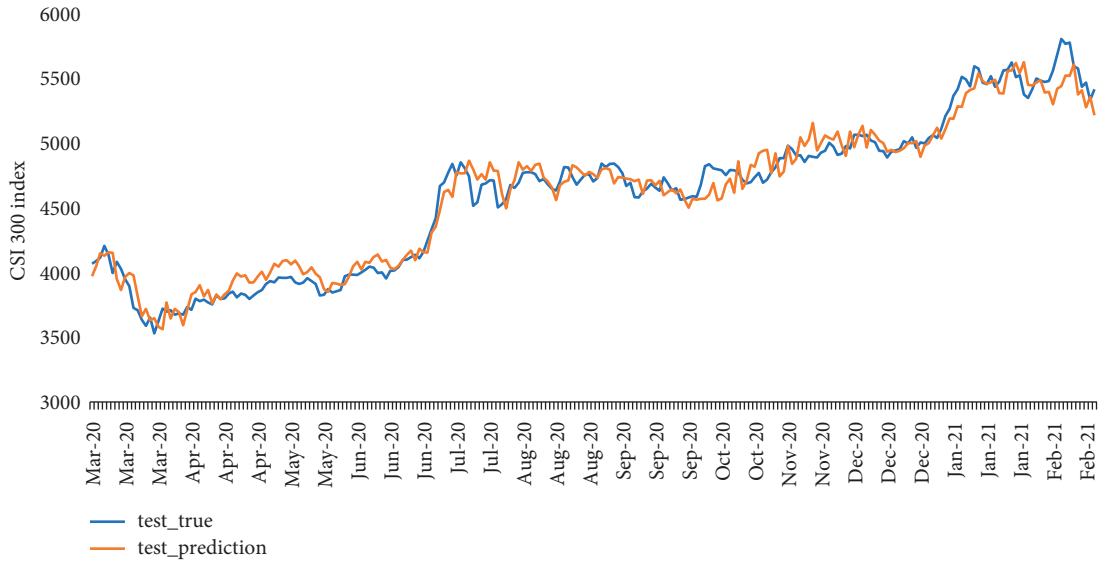

FIGURE 7: The trend chart of the relationship between LSTM model predicted values of Generated vocabulary and CSI 300 index true values. The red and blue lines represent the true and predicted values, respectively ($n = 243$).

Compared with the seed words of the CSI 300 Index, the same number of generated words obtained by the BERT word vector similarity filtering and NEZHA keyword selection have more stable and smooth prediction results for the CSI 300 Index. For our prediction task, this study uses the Root Mean Squared Error (RMSE) indicator as a measure of the model's predictive ability. The smaller the RMSEmeans the better the predictive effect. The calculation formula is

$$RMSE = \sqrt{\frac{1}{m}\sum_{k=1}^{m}(y_k - \hat{y}_k)^2}, \tag{17}$$

where y_k represents the true value, $\hat{y}_k$ represents the predicted value, and m represents the sample size of the test set. As the result shows, in this experiment, the RMSE is 154.1831 when the lagging term of the CSI 300 index itself and the seed keywords' searchindexes are used as the predictor variable. However, the RMSE is 110.6976 whenthe lagging term of the CSI 300 index itself and the generated keywords' searchindexes are used as the predictor variable. The decrease rate is 28.20%.

Our experimental results show that, compared with the original seed keywords, the NLP text mining technology designed in this study improves the prediction accuracy and accuracy of LSTM on the Shanghai and

Shenzhen 300 stock indexes by new generated keywords with better predicting stability and better forecasting ability.

6. Conclusion

Based on BERT and NEZHA models of artificial intelligence, we optimize the text mining technology for stock price index prediction and deeply expand the keywords of higher quality predictive variables. On this basis, we use the LSTM prediction model to empirically forecast the CSI 300 stock index. The empirical results show that, based on the text information mining method of BERT model similarity and NEZHA model importance, we can screen out high-quality prediction variables with higher correlation and stronger prediction ability from network texts, thus significantly improving the prediction effect of CSI 300 stock index.

The implications are as follows: First, the artificial intelligence text mining technology based on BERT and NEZHA frontier can be better applied to stock price prediction, which not only enriches the index system of stock price prediction but also helps regulators and investors to evaluate stock price trends and control stock price risks. Second, the text mining technology can realize the keyword expansion of stock price forecast, which can provide research ideas and references for the expansion of other macro index systems. In addition, this method has strong extensibility. Future research can consider more analysis angles based on similarity and importance to achieve more high-quality keyword extension, which is also worth exploring in the following research.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

The authors acknowledge the helpful comments from anonymous reviewer. Xiaobin Tang was supported by CAAI-Huawei MindSpore Open Fund (no. CAAIXSJLJJ-2021-045A) and the Outstanding Young Scholars Funding Program of UIBE [No. 21JQ09]. Dan Ma was supported by the National Social Science Foundation of China (grant number no. 21&ZD149).

References

[1] E. F. Fama, "Efficient capital markets: a review of theory and empirical work." *The Journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[2] M. Z. Asghar, F. Rahman, F. M. Kundi, and S. Ahmad, "Development of stock market trend prediction system using multiple regression," *Computational & Mathematical Organization Theory*, vol. 25, no. 3, pp. 271–301, 2019.

[3] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the ARIMA model," in *Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pp. 106–112, Cambridge, UK, March 2014.

[4] T. Vantuch and I. Zelinka, "Evolutionary based ARIMA models for stock price forecasting," *ISCS 2014: Interdisciplinary Symposium on Complex Systems*, Springer, Cham, Manhattan, NY, USA, 2015.

[5] S. Mootha, S. Sridhar, R. Seetharaman, and S. Chitrakala, "Stock Price Prediction Using Bi-directional LSTM Based Sequence to Sequence Modeling and Multitask Learning," in *Proceedings of the 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, USA, October 2020.

[6] S. Mehtab and J. Sen, "Stock Price Prediction Using Convolutional Neural Networks on a Multivariate Timeseries," 2020, https://EconPapers.repec.org/RePEc:arx:papers:2001.09769.

[7] L. Dos, S. Pinheiro, and M. Dras, "Stock market prediction with deep learning: a character-based neural language model for event-based trading," in *Proceedings of the Australasian Language Technology Association Workshop 2017*, pp. 6–15, Brisbane, Australia, December 2017.

[8] X. C. Xu and K. Tian, "A new method of stock index prediction based on sentiment analysis of financial text," *Journal of Quantitative and Technical Economics*, vol. 38, no. 12, pp. 124–145, 2021.

[9] R. Kaur, Y. K. Sharma, and D. P. Bhatt, "Measuring accuracy of stock price prediction using machine learning based classifiers," *IOP Conference Series: Materials Science and Engineering*, vol. 1099, no. 1, pp. 12–49, 2021.

[10] R. Gupta and M. Chen, "Sentiment Analysis for Stock Price Prediction," in *Proceedings of the 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, August 2020.

[11] S. Mehtab and J. Sen, "Stock Price Prediction Using CNN and LSTM-Based Deep Learning Models," 2020, https://arxiv.org/abs/2010.13891.

[12] A. Rui, "Big data business actual analysis: stock price prediction based on time series model,," *Modern Economics & Management Forum*, vol. 2, no. 2, pp. 63–71, 2021.

[13] S. Mehtab and J. Sen, "A Time Series Analysis-Based Stock Price Prediction Using Machine Learning and Deep Learning Models," 2020, https://arxiv.org/abs/2004.11697.

[14] G. Salton and C. T. Yu, "On the construction of effective vocabularies for information retrieval," *ACM Sigplan Notices*, vol. 10, no. 1, pp. 48–60, 1975.

[15] S. Deerwester and T. Landauer, G. Furnas, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[16] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[17] R. Mihalcea and P. Tarau, "Textrank: bringing order into text," in *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pp. 404–411, Stroudsburg, PA, USA, July 2004.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/abs/1301.3781.

[19] M. E. Peters, M. Neumann, M. Iyyer et al., "Deep contextualized word representations," 2018, https://arxiv.org/abs/1802.05365.
[20] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018, https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.
[21] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018, https://arxiv.org/abs/1810.04805.
[22] J. Wei, X. Ren, X. Li et al., "NEZHA: neural contextualized representation for Chinese language understanding," 2019, https://arxiv.org/abs/1909.00204.
[23] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune BERT for text classification?" *Lecture Notes in Computer Science*, Springer, Cham, vol. 11856, Manhattan, NY, USA, 2019.
[24] J. Zhu, Y. Xia, L. Wu et al., "Incorporating BERT into Neural Machine Translation," 2020, https://arxiv.org/abs/2002.06823#:~:text=The%20recently%20proposed%20BERT%20has,(NMT)%20lacks%20enough%20exploration.
[25] C. Qu, L. Yang, M. H. Qiu, B. Croft, Y. Zhang, and M. Iyyer, "BERT with History Answer Embedding for Conversational Question Answering," pp. 1133–1136, 2019, https://arxiv.org/abs/1905.05412.
[26] L. Kong, L. Wang, W. Gong, C. Yan, Y. Duan, and L. Qi, "LSH-aware multitype health data prediction with privacy preservation in edge environment," *World Wide Web*, 2021.
[27] T. Preis, H. S. Moat, and H. E. Stanley, "Quantifying trading behavior in financial markets using Google Trends," *Scientific Reports*, vol. 3, no. 1, pp. 1684–1686, 2013.
[28] B. Weng, M. A. Ahmed, and F. M. Megahed, "Stock market one-day ahead movement prediction using disparate data sources," *Expert Systems with Applications*, vol. 79, pp. 153–163, 2017.
[29] H. Hu, L. Tang, S. Zhang, and H. Wang, "Predicting the direction of stock markets using optimized neural networks with Google Trends," *Neurocomputing*, vol. 285, pp. 188–195, 2018.
[30] Y. Liu, G. Peng, L. Hu, J. Dong, and Q. Zhang, "Using Google Trends and Baidu index to Analyze the Impacts of Disaster Events on Company Stock Prices," *Industrial Management & Data Systems*, vol. 120, 2019.
[31] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.
[32] B. Li, X. Y. Shao, and Y. Y. Li, "Research on fundamental quantitative investment driven by machine learning," *China Industrial Economics*, vol. 8, pp. 61–79, 2019.
[33] Y. Liu, D. Li, S. Wan et al., "A long short-term memory-based model for greenhouse climate prediction," *International Journal of Intelligent Systems*, vol. 37, no. 1, pp. 135–151, 2022.
[34] J. Sen, S. Mehtab, and A. Dutta, "Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models," 2021, https://arxiv.org/abs/2009.10819.
[35] Y. Baek and H. Y. Kim, "ModAugNet: a new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module," *Expert Systems with Applications*, vol. 113, pp. 457–480, 2018.
[36] J. Y. Sun, "Jieba" Chinese Text Segmentation," 2012, https://github.com/fxsjy/jieba.
[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[38] L. Xu, X. Zhang, and Q. Dong, "CLUECorpus2020: a large-scale Chinese corpus for pre-trainingLanguage model," 2020, https://arxiv.org/abs/2003.01355.
[39] Y. Fan, "Research on the next-generation deep learning framework," *Big Data Research*, vol. 6, no. 4, pp. 69–80, 2020.
[40] Mindspore, 2020, https://www.mindspore.cn/.
[41] X. Tang, M. Dong, and R. Zhang, "Research on the prediction of consumer confidence index based on machine learning LSTM&US model," *Statistical Research*, vol. 37, no. 7, pp. 104–115, 2020.