


Research Article

Research on the Classification of Policy Instruments Based on BERT Model

Jiani Zhao ^{1,2} and Cheng Li¹

¹College of Air Transportation, Shanghai University of Engineering Science, Shanghai, China

²School of Information Management & Engineering, Shanghai University of Finance and Economics, Shanghai, China

Correspondence should be addressed to Jiani Zhao; zhaojiani@sues.edu.cn

Received 31 July 2022; Revised 4 September 2022; Accepted 30 September 2022; Published 14 October 2022

Academic Editor: Lele Qin

Copyright © 2022 Jiani Zhao and Cheng Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Currently, policy instruments are classified mainly by means of manual encoding and checking, which is highly subjective and inefficient, which greatly hinders the development of policy research. The research tries to apply the automatic classification algorithm based on BERT (Bidirectional Encoder Representation from Transformer) to the policy instruments to improve the efficiency and accuracy of policy instruments classification. An entrepreneurship policy instrument classification model was established on the basis of the pretraining language model to realize the automatic classification of entrepreneurship policy instruments. The research showed that through optimization and improvement of the model, the F1 value was 0.86 on the test set, indicating a good classification effect; through the comparative experiment, it was further proved that the classification effect of this model was far superior to other three commonly used text classification models. The model greatly improves the efficiency and objectivity of policy instrument classification and provides a new idea for investigating entrepreneurship policies and more generalized policy instruments.

1. Introduction

As a common way of governing a state's social public affairs, policy instruments serve as a vital bridge that connects policy targets and policy results. Regarding the interpretation of specific policies, the selection and application of policy instruments reflect the diversity of public policies [1]. When public policies are implemented, different policy instruments may lead to varying policy objects. Moreover, the standards for assessing a policy instrument's effect also influence the final policy goal [2]. Policy instruments influence the goals and status of implementing policies. Beyond that, the government conducts several combination designs of policy instruments according to the circumstances of each industry to reach high adaptability. On the whole, policies cannot do without support from policy instruments in the entire process, ranging from formulation to proper execution. Sufficient and detailed data on the classification of policy instruments will provide a solid

foundation for the follow-up research on policy instrument coordination analysis and policy evaluation and provide a basis for further expanding the depth of policy instrument research. Currently, there are no unified standards in academia for classifying policy instruments. Hence, public policy researchers have proposed distinctive classification standards based on the respective research problems, interests, and areas. Consequently, policy instruments come in a diverse range of categories [3]. However, few studies in the theory circle have focused on the way to classify policy instruments. Instead, manual encoding and rechecks are employed [4–6]. Not only is manual classification strongly subjective but also it considerably lowers the efficiency of researching policy instruments. Despite this, the automatic classification algorithm based on text analysis has not been used with research policy instruments.

Therefore, the research proposes a BERT-based deep learning model to study the policy instrument classification of entrepreneurial policy texts referring to predecessors'

research. In recent years, the mode of pretraining-parameter fine-tuning has been a new pattern in the natural language processing realm [7]. Generally speaking, most deep learning models must be trained by vast data [8]. Unfortunately, this study only collected 470 pieces of entrepreneurial policy text, which can be divided into about 10,000 units for policy instrument analysis. This volume is smaller than other corpora, and the number of manually marked policy instruments is less than 2,000. Such a small sample size is unfit for training deep learning models with many parameters [9]. Nevertheless, the pretraining language models represented by BERT can conduct pretraining in a super-large corpus and start fine-tuning in samples with few downstream tasks. Thus, it can achieve an outstanding effect without many samples [10]. Owing to this trait, the BERT model achieves a satisfactory effect even in sample tasks with a small size of text, which is very suitable for the research scenario of this paper.

2. Literature Review

Text classification is devoted to classifying diverse levels of text, including sentences, paragraphs, and documents, making it a critical research orientation for natural language processing [11]. Currently, there have been many successful cases of applying text analysis to the question answering system, conversation system, and analysis of public opinions [12]. Existing mainstream text analysis methods can be divided into two categories: methods based on traditional machine learning and those based on deep learning. The latter is attracting growing favor from researchers.

Feed-forward network (FFN) is one of the earliest deep neural networks. This model regards the text as one “bag-of-words,” namely, a set of words, without considering the dependence relationship between different words and the sequence of words [13]. Recurrent neural network (RNN) [14] regards the text as a group of word sequences, to extract the relationship between words, text structure, and other information. However, the original RNN fails to achieve the same effect as the FFN model in the actual application of text classification. It is mainly because the original RNN brings a vanishing gradient and gradient explosion in back-propagation, which results in a catastrophic impact on optimizing the model’s parameters. Subsequently, the researchers optimized the original RNN and proposed an LSTM model. The convolution neural network (CNN) [15] is mainly used to process images. Unlike the time sequence information between the input sequences extracted by RNN, CNN excels in extracting the spatial position of input features [16].

When humans read, they pay varying degrees of attention to each word in the text, and the attention mechanism was brought forth under the inspiration of this phenomenon. Transformer [17] is a self-attention network structure and has overcome the defect that RNN and its variants cannot be trained. Compared with CNN, this model can be used to directly calculate the relationship between two words separated by a long distance and is not limited by the length of the convolution core in CNN. Due to these merits,

transformer achieves an outstanding effect in actual application. Thus, it has emerged as the most mainstream network structure in natural language processing. As the natural language processing technology advances, the model of pretraining-parameter fine-tuning becomes a new pattern in this realm [7]. This pattern works in specific procedures. (1) the pretraining stage: some self-supervision learning tasks are designed in super-large corpora to pretrain the model. (2) fine-tuning stage: the parameters of the pre-trained model are fine-tuned in specific downstream tasks (such as text classification). The principal structure of ELMO [18] is a multilayered LSTM network, which adopts autoregression tasks in the pretraining stage to predict the next word based on a group of given words. BERT [19] is a pretraining language model proposed by Google AI Language in 2018 based on the Transformer network. It mainly takes on two tasks in the pretraining stage: Masked Language Model (MLM) and Next Sentence Prediction (NSP). The advent of BERT marks a milestone progress in the natural language processing field. Since it was launched, it has considerably outperformed previous models in several natural language processing tasks [10]. Currently, BERT is applied to text summary, machine translation, text similarity calculation, question answering system, text classification, and so on [20].

In recent years, text mining concerning policies and politics have been the new research direction in the natural language processing realm. D’Orazio et al. [21] judged whether a policy document involves military-related information through the Support Vector Machine model. Krebs [22] extracted information from a document to identify whether it may carry a potential nationalist intention. Chang and Masterson [23] proposed a deep network model based on LSTM to classify and process different categories of policy texts. Pujari and Goldwasser [24] proposed a BERT-based neural network model and integrated the author’s information to encode the policy text information from social media more accurately. Mukherjee et al. [25] employed Bart, an optimized version of BERT, to divide the text on social media, such as Twitter, into three categories: public health, policy, and others. Thus, it can be seen that the theoretical circle has not studied the text mining of policy texts extensively. Moreover, the application of natural language processing to policy instrument classification is more limited in the Chinese realm. Therefore, based on predecessors’ research, the paper proposed a deep learning model based on the pretraining language model to classify entrepreneurial policy texts into policy instruments.

3. Research Methodology

3.1. Model Settings

3.1.1. Model Structure. The BERT model structure is displayed in Figure 1, which mainly consists of N (a preset model hyperparameter, generally taken as 12) Transformer coding units. The core idea of BERT is the bidirectional attention mechanism introduced by Transformer encoder. BERT piles multiple Transformer encoders together, and

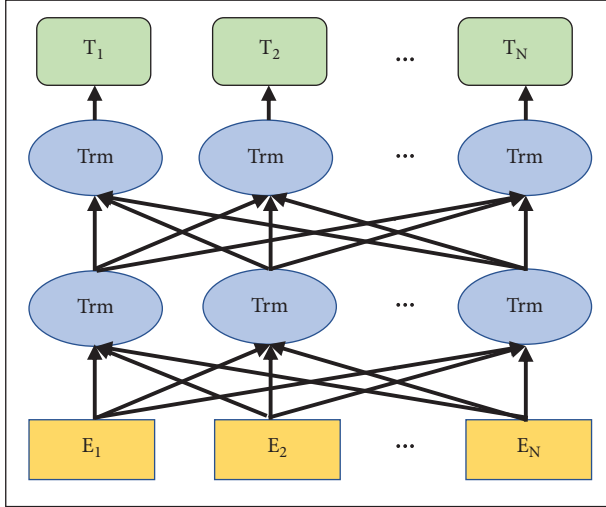


FIGURE 1: BERT model structure chart.

each encoder contains multihead attention mechanism and splices the coding of each multihead attention mechanism as the input of the next layer. Therefore, text features can be extracted better.

3.1.2. Pretraining Task. Two pretraining tasks are mainly proposed by BERT, namely, masked language model (MLM) and next sentence prediction (NSP). The core idea of MLM lies in randomly masking the tokens in a sentence at a certain probability so that they can be predicted by this model according to the context of the masked tokens. As for NSP, two sentences A and B are given, enabling the model to judge whether the sentence B is the next sentence of the sentence A in a concrete context. Positive and negative samples can be constructed by randomly sampling in large-scale corpora, to form a training set of NSP tasks.

3.1.3. Text Embedding. BERT embedding is mainly composed of three parts: token embedding, segment embedding, and position embedding, where token embedding resembles the word embedding in natural language processing. The tokens after text segmentation are mapped by BERT tokenizer to a vector one by one. Since the input of BERT may be two sentences, segment embedding means embedding different sentences. If the input is just one sentence, the segment embedding corresponding to each input token is E_A . If the input is two sentences A and B, the segment embedding corresponding to each token in the sentence A is E_A and that in the sentence B is E_B . E_A and E_B represent the segment embedding of sentences A and B, respectively. Given the lack of the position information of tokens in the sentences of transformer structure, the position of each word should be embedded, namely, each position is mapped to a vector that can be learned.

3.1.4. Transformer Encoder and Attention Mechanism. The core idea of transformer encoder lies in the introduction of the attention mechanism. For instance, when reading

entrepreneurship policy texts, people tend to pay more attention to words related to concrete policy instruments like “tax and rate reduction” and “guaranteed loan” while neglecting some auxiliary words or those without practical significance, such as “different regions” and “different departments.” Transformer encoder calculates the attention attracted by different words using the matrix multiplication method, to enable the parallel computing on GPU, which, to a great extent, accelerates the model training and computing speed. In addition, Transformer encoder is able to process the word dependence inside a random-length sequence, without needing to consider problems like gradient missing and explosion in RNN.

Multiheaded attention mechanism inputs the three following vectors: query vector $Q \in \mathbb{R}^{l \times d_q}$, key vector $K \in \mathbb{R}^{l \times d_k}$, and value vector $V \in \mathbb{R}^{l \times d_k}$, where l is the length of input sequence, d_q stands for the dimension of query vector, and d_k denotes the dimension of key vector and value vector. In practical applications, query vector is obtained by linearly transforming query statements (or its implicit expression), while key vector and value vector are acquired by linearly transforming key statement and value statements (or their implicit expressions).

In the multiheaded attention mechanism, each “head” corresponds to a scaled dot-product attention. Assume that there are totally h “heads” in the multiheaded attention mechanism, query, key, and value vectors are, respectively, input into h scaled dot-product attentions, and the h outputs are finally spliced as the final output of the multiheaded attention mechanism.

Scaled dot-product attention is specifically calculated through the following formula:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (1)$$

$$\text{softmax}(X) = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}},$$

where $\text{softmax}(X)$ is the activation function to realize the logistic regression of data; that is, the numerical vector is normalized to the probability distribution vector, and the sum of each probability is 1.

Therefore, the final output of multiheaded attention mechanism is $A = [A_1, A_2, \dots, A_h]$, where A_i is the output of the i (th) scaled dot-product attention.

3.2. Research Model and Process

3.2.1. Modeling. In this research, a PLM-multilayer perceptron (MLP) combined text classification model was proposed to classify the policy instruments in entrepreneurship policy texts. The open-source Chinese BERT model was firstly acquired, which was already pretrained in large-scale Chinese corpora, and a good deal of knowledge was learned. On this basis, the manually annotated entrepreneurship policy instrument samples were used to fine-adjust this BERT model.

The structure of the proposed model is exhibited in Figure 2, which mainly consists of three parts: PLM, max-pooling layer, and MLP, where PLM aims to extract a matrix representation fusing contextual information from policy instruments, max-pooling layer to perform dimension transformation, i.e., dimension reduction, of the characteristic matrix extracted from the PLM, and MLP to classify the textual characteristics extracted from the aforementioned network. In this research, entrepreneurship policy instruments were finally classified into 12 classes.

Since Chinese BERT model implements word segmentation by taking Chinese characters as units, the model input is a group of character sequences. In this research, the input of the BERT model was a character sequence with the length of 512, e.g., the analytical unit length of one policy instrument was 100 characters. As required by the BERT model, a special character “[CLS]” was added at the forefront of the sequence, and the length of this character sequence was 101 after word segmentation, being less than 512, so $512 - 101 = 411$ [pad]s were supplemented after this sequence until the final sentence length was 512. The vector corresponding to [pad] was 768-dimensional (768 is the dimension of the hidden layer), and the value of it was kept at 0. If there were 600 characters in the analytical units of one policy instrument, the first 511 characters in this sentence were extracted and [CLS] was added at the forefront to form a 512-character sequence as the input of the BERT model. After this group of character sequence after the word segmentation was acquired, each character was mapped to a 768-dimensional vector according to the mapping table in BERT tokenizer, namely, the glossary of token embedding, and this vector was integrated into the segment embedding and position embedding vectors corresponding to this character to form a final vector of this character, which was finally input into the BERT model.

The input sequence of PLM is set as X . The above process is expressed by the following formula:

$$H = \text{PLM}(X), \quad (2)$$

where $\text{PLM}(\ast)$ stands for PLM; $H \in \mathbb{R}^{l \times d_p}$ denotes the output of PLM; l is the sequence length of the policy instrument, which was 512 in this model, and the part less than 512 was complemented by [pad]; and d_p represents the state dimension at the hidden layer of PLM, which was set to $d_p = 768$ in this model.

After the output H of PLM was acquired, it was input into the max-pooling layer, to transform the matrix output by PLM into a vector as an eigenvector and finally input into MLP. The schematic diagram of max-pooling layer is displayed in Figure 3, i.e., the maximum value was taken from each row of the 768×512 matrix output by PLM to obtain a 768-dimensional vector, which was, namely, the output of max-pooling layer (see the following formula):

$$M = \text{MaxPool}(H), \quad (3)$$

where $\text{MaxPool}(\ast)$ represents the max-pooling layer and $\in \mathbb{R}^{d_p}$ stands for its output. In this model, $d_p = 768$, so M was a 768-dimensional vector.

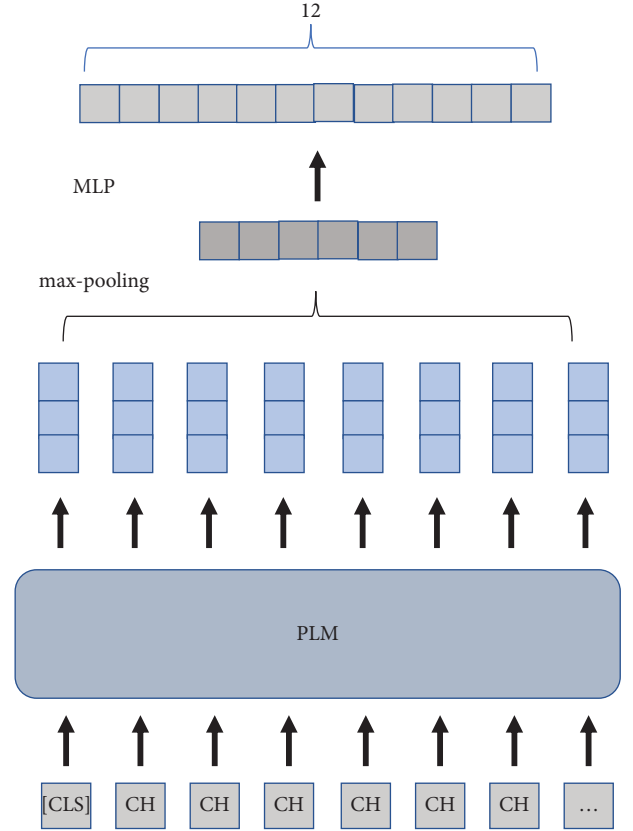


FIGURE 2: Structure chart of policy instrument classification model.

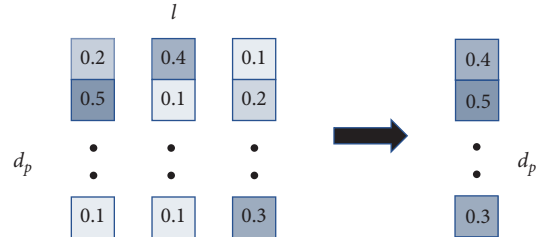


FIGURE 3: Schematic diagram of max-pooling layer.

In the end, the output M of max-pooling layer was input into MLP for classification, to obtain the final model output L , as follows:

$$L = W_2(\sigma(W_1M + b_1)) + b_2, \quad (4)$$

where W_* and b_* represent model parameters, $\sigma(\ast)$ is an activation function, $\sigma(x) = 1/(1 + e^{-x})$, L is the output vector of MLP, and the dimension of L is the number of policy instrument classes (12 in this research). Next, softmax processing was performed for L to facilitate its normalization. Therefore, the numerical value at the position i of the processed vector $\text{softmax}(L)$ was the probability for the input policy instrument analytical unit to belong to the class i , and the final classification result of this input policy instrument analytical unit obtained by the model was the class corresponding to the maximum probability in the vector.

3.2.2. Samples and Data Processing. In this research, a total of 178 entrepreneurship policy texts were chosen from 470 national-level entrepreneurship policies and then manually encoded. The 178 entrepreneurship policy texts included 7 documents issued by the State Council, 15 ones issued by the General Office of the State Council and 156 ones issued by other national ministries and commissions, covering representative entrepreneurship documents at each level of each department, with a time span of 2001–2020, so the whole research period was covered. Finally, 1,682 manually annotated analytical units, namely, policy documents, were acquired.

Whether the coding and classification of analysis units in sample policies are reasonable and reliable is directly related to the accuracy of the classification of policy instruments in subsequent studies. Therefore, it is necessary to conduct reliability analysis on the results of manual classification. In order to ensure the accuracy of coding, avoid researchers from making wrong judgments on policy texts due to subjective awareness and value preferences, and ensure a high reliability level, this study adopts the reliability formula of content analysis [26] to test the reliability of coding. The formula is as follows:

$$R = \frac{n \times k}{1 + (n - 1) \times k}, \quad (5)$$

where R is the reliability; n is the number of judges, $n = 2$ in this study, k is the average degree of mutual agreement (that is, the degree of mutual agreement between two judges), and its formula is

$$k = \frac{2M}{N_1 + N_2}, \quad (6)$$

where M is the number of policy instruments that the two judges agree on, N_1 is the number of policy instruments that the first judge judges, and N_2 is the number of policy instruments that the second judge judges.

In this study, the two judges classified 1682 texts with policy instruments at the same time and compared and analyzed the classification results. Among them, 1510 were consistent and 172 were inconsistent, so the average agreement degree $k = 0.8977$, and the reliability degree $R = 94.61\%$. It is generally believed that when the reliability is above 0.7, the previous research can be considered credible enough [27]. Therefore, the categories of entrepreneurship policy instruments classified in this study for the policy content analysis unit are credible. In order to further improve the accuracy of the classification of policy instruments, a secondary analysis was conducted for the 172 analysis units with inconsistent classification. The two judges discussed and negotiated, and expert consultation was adopted for the analysis units with large differences, and the classification of policy instruments was finally determined, as shown in Table 1.

3.2.3. Experimental Environment and Parameter Settings. In this research, the model was established using *Python3* programming language under the deep learning framework of *PyTorch* to complete the classification task of

entrepreneurship policy instruments. The experimental environment is depicted in Table 2.

The different values of hyperparameters were acquired through grid search within reasonable ranges, with partial results as shown in Figure 4, where the horizontal axis stands for the value of each hyperparameter and the longitudinal axis denotes the model effect under this value. Among the different values of each hyperparameter, the value corresponding to the best model effect was regarded as the value of this hyperparameter. Therefore, the model harvested the best effect under the batch size of 8 and learning rate of $2e-5$. The final hyperparameter values and the related training details of this model are as seen in Table 3.

3.2.4. Evaluation Indices. To facilitate the evaluation of the model effect, universal evaluation criteria were used, namely, accuracy, precision, recall, and F1 as evaluation indices. The concrete formulas for such evaluation indices of the samples belonging to class i are as follows:

$$\begin{aligned} \text{accuracy} &= \frac{\sum_{i=1}^N (TP_i + TN_i)}{\sum_{i=1}^N (TP_i + TN_i + FP_i + FN_i)}, \\ \text{precision}_i &= \frac{TP_i}{TP_i + FP_i}, \\ \text{recall}_i &= \frac{TP_i}{TP_i + FN_i}, \\ F1_i &= \frac{2 \times \text{precision}_i \times \text{recall}_i}{\text{precision}_i + \text{recall}_i}, \end{aligned} \quad (7)$$

where TP_i is the number of true values, TN_i is the number of true negative values, FP_i is the number of false positive values, and FN_i is the number of false negative values.

The above formulas represent the evaluation indices for the samples of class i . The evaluation indices of each class should be integrated to evaluate the comprehensive model performance through integration methods like macro, micro, and weighted average. In this research, Weighted Average was chosen to integrate the evaluation indices of each class, specifically as follows:

$$\begin{aligned} \text{precision} &= \sum_{i=1}^N (\text{precision}_i \times w_i), \\ \text{recall} &= \sum_{i=1}^N (\text{recall}_i \times w_i), \\ F1 &= \sum_{i=1}^N (F1_i \times w_i), \end{aligned} \quad (8)$$

where w_i is the proportion of samples belonging to class i in all the samples.

3.2.5. Experimental Process. To better evaluate the model effect in the training process, the sampled dataset was randomly segmented according to the proportion of 8 : 2,

TABLE 1: Manual annotation frequencies of policy instruments.

Class	A1	A2	A3	A4	B1	B2	B3	B4	C1	C2	C3	C4
Quantity	183	318	54	120	19	22	29	44	171	199	187	336

TABLE 2: Experimental environment.

Name	Configuration
CPU	Intel (R) Xeon (R) Silver 4114 CPU @ 2.20 GHz
Memory	32 GB
GPU	Tesla V100 16 GB
System	Ubuntu 20.04

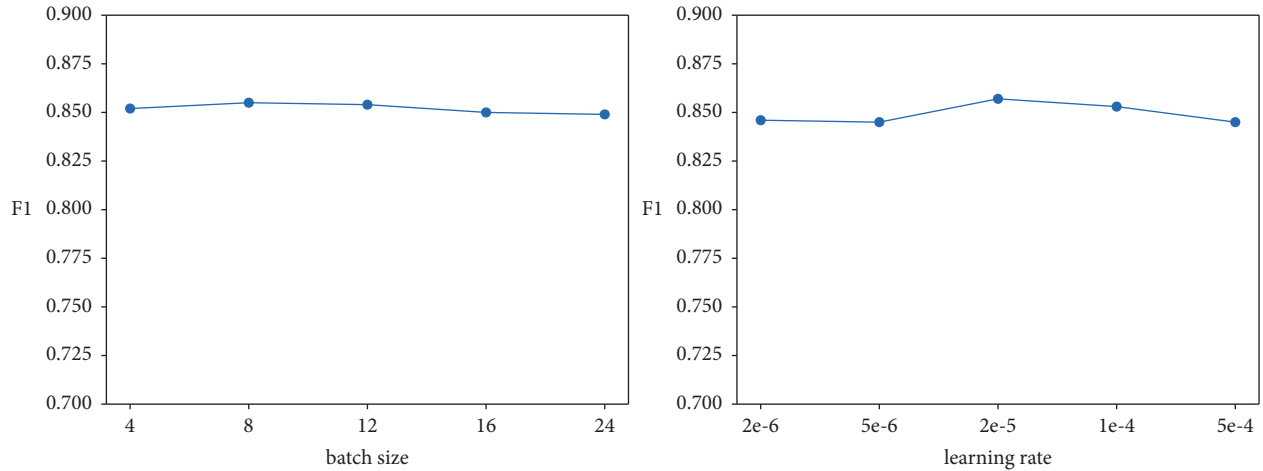


FIGURE 4: Analysis of different hyperparameter values.

TABLE 3: Hyperparameter settings.

Name	Value	Meaning
Optimizer name	Adam	Category of the optimizer used in model training
Batch size	8	Size of data trained by the model in each step
Epochs	10	Number of iterations on the training set during model training
Learning rate	2e-5	Learning rate in the optimizer
Dropout rate	0.2	Probability of shielding partial networks

and finally a training set (1346 units) and a test set (336 units) were acquired. The training set aimed to train the model and optimize the model parameters, and the test set aimed to evaluate the model effect.

It was not difficult to find by observing the manually annotated policy instrument samples that serious class unbalance existed among the samples, namely, the proportion of samples belonging to each class in all samples was unbalanced. Figure 5 shows the proportion of different categories in the sample. The class C4 with the largest number of samples accounted for 19.98% of all samples, while the class B1 with the least number of samples only accounted for 1.13%. On the whole, the proportion of class B policy instrument samples was evidently lower than that of the other classes.

Specific to this problem, Focal Loss was experimentally used to optimize the conventional Cross Entropy Loss of the model. This loss function could improve the loss weight of

the class with the least number of samples and reduce that of the class with the largest number of samples, to balance the losses of different classes. The concrete calculation formula for Focal Loss is as follows:

$$FL(p_t) = -(1 - p_t)^\gamma p_t, \quad (9)$$

where p_t is the model output of class t , i.e., the probability for a policy instrument to belong to the class t ; γ represents a hyperparameter preset by Focal Loss, and the results of grid search for γ are shown in Figure 6. Finally, it appeared that the model effect was the best under γ value of 2.

In the experiment, BERT-base-Chinese was firstly used as a pretrained model, and the final F1 value on the test set was 0.82. Next, the conventional Cross Entropy Loss of the model was optimized using Focal Loss to tackle the class unbalance of policy instrument samples, followed by a further experiment. Finally, the F1 value on the test set was

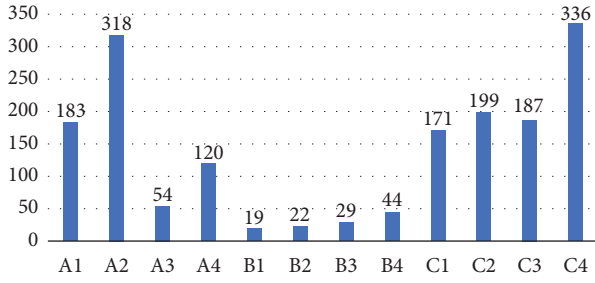


FIGURE 5: Schematic diagram of the proportion of training data.

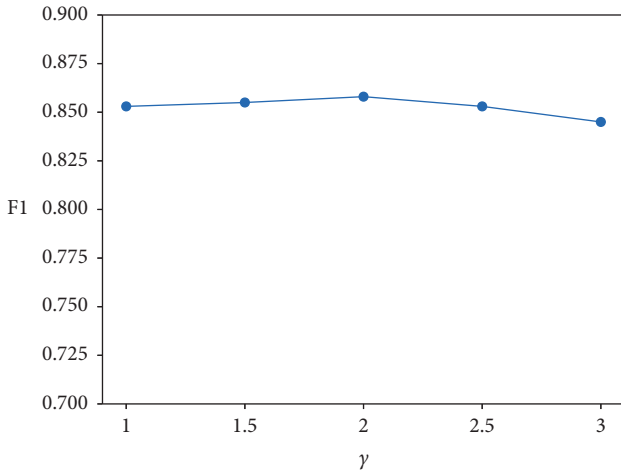


FIGURE 6: Value analysis of γ .

0.84, increasing by 2 percentages compared with that in the previous experiment.

Subsequently, different BERT modes were comparatively analyzed. Since the parameters in BERT pretraining process were further optimized by roberta model [28], a better effect than the original BERT model was achieved in multiple downstream tasks. Therefore, efforts were also made in this research to replace BERT-base-Chinese model with Chinese-RoBERTa-wwm-ext model to perform a similar experiment. Finally, the F1 value of Chinese-RoBERTa-wwm-ext model on the test set was 0.86, which was 2 percentages higher than the previous prediction effect.

Given this, Chinese-RoBERTa-wwm-ext was finally used as the PLM. The trend of the loss value during the training process of the proposed model is displayed in Figure 7, where the horizontal axis represents the training step and the longitudinal axis denotes the loss value under the current step number. Clearly, the model parameters tended to be converged after 2,000 steps, and the loss value was kept below 0.1 and no longer fluctuated.

The model effect on the test set upon the completion of each iterative step during the training process is exhibited in Figure 8, where the horizontal axis represents the number of iterations already completed in the training process and the longitudinal axis stands for the value of each evaluation index. Therefore, it could be seen that the proposed model gained the best effect after 10 iterations, under which all evaluation indices reached the highest values.

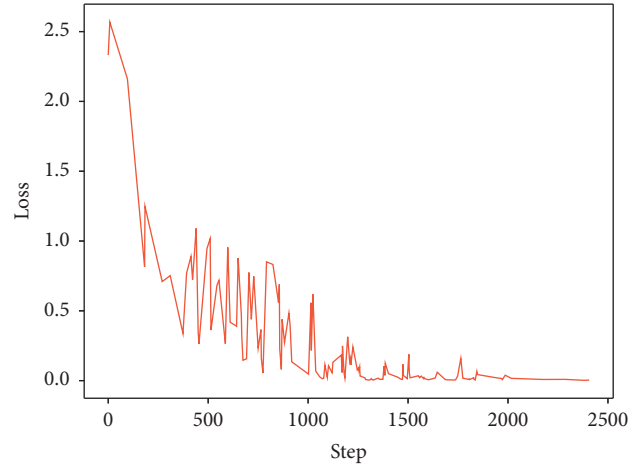


FIGURE 7: The trend of the loss value.

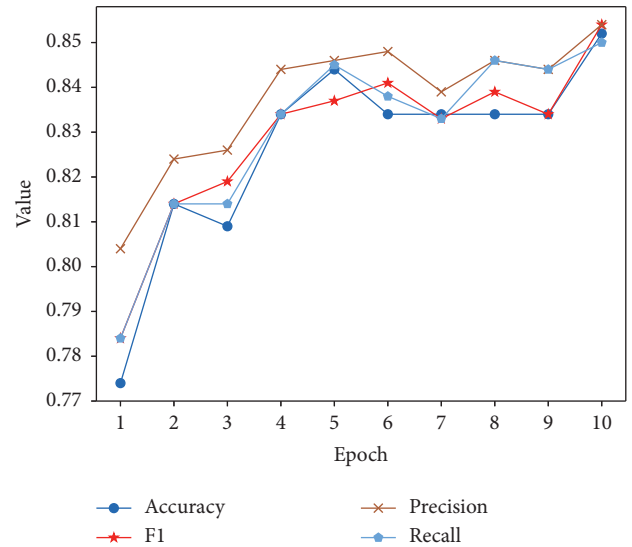


FIGURE 8: Test effect of the model.

4. Results and Discussion

To further verify the model effectiveness and feasibility, the experimental results obtained by the proposed policy instrument classification model (PLM + MLP) were compared with those achieved by three models (TF-IDF + SVM, TextCNN, and LSTM) performing well in previous studies. The names of the comparison models and their corresponding concrete values are listed in Table 4.

The classification effects achieved by different text classification models are listed in Table 5, among which the last three belong to the proposed classification model (PLM + MLP). The column of pretrained models lists the concrete name of each PLM. Since PLM was not involved in TF-IDF + SVM, TextCNN, or LSTM, these items were left blank.

It could be seen from Table 5 that no matter based on which pretrained model, the PLM + MLP model reached a much better effect than the first three models mainly because the PLM could learn more universal knowledge from large-scale corpora, thus greatly enhancing the text representation

TABLE 4: Comparative experimental models.

Model name	Meaning
TF-IDF + SVM	Texts are quantified using TF-IDF and text vectors are input into SVM for classification
TextCNN	Word vectors are randomly initialized and input into TextCNN for classification
LSTM	Word vectors are randomly initialized and input into LSTM for classification
PLM + MLP	The PLM-based classification model proposed in this research

TABLE 5: Comparative experimental results.

Model	Pretrained model	Loss function	Accuracy	F1	Precision	Recall
TF-IDF + SVM	—	CE	0.75	0.74	0.75	0.75
TextCNN	—	CE	0.67	0.66	0.68	0.67
LSTM	—	CE	0.67	0.67	0.71	0.67
PLM + MLP	BERT-base-Chinese	CE	0.82	0.82	0.83	0.83
PLM + MLP	BERT-base-Chinese	Focal	0.83	0.84	0.85	0.83
PLM + MLP	Chinese-RoBERTa-wwm-ext	CE	0.85	0.86	0.86	0.85

ability. This model could achieve a good effect even in the case of a small downstream training data size. Meanwhile, the experimental results revealed that the Chinese-RoBERTa-wwm-ext-based model reached a better effect than the BERT-base-Chinese-based model. This was mainly because the parameters of this model were optimized better in the pretraining stage, and thus this model could learn the representation better than the original BERT model did and harvest a better effect in the downstream tasks of policy instrument classification.

In addition, it was found from Table 5 that the effect of TF-IDF + SVM model was better than that of LSTM and TextCNN mainly for the following reasons: the language is normative in policy texts, and the texts belonging to each class of policy instruments usually include some fixed terms. For example, terms like “talent” and “education” are contained in policy instruments belonging to class A1 (talent cultivation), and these text features can be extracted by TF-IDF very well. Because just a few training corpora were used in this research, TextCNN and LSTM models could hardly train a good word eigenvector on the contrary, which degraded their classification effect to a great extent.

5. Conclusion

In this research, an entrepreneurship policy instrument classification model (PLM + MLP) was established through the PLM-based text analysis technology to realize the automatic classification of entrepreneurship policy instruments. Finally, the F1 value on the test set reached 0.86, manifesting a satisfactory classification effect. Hence, this model is applicable to the research on national-level entrepreneurship policies and to the classification of local entrepreneurship policy instruments in China. If used, this type of automatic classification algorithm can greatly improve the classification efficiency and objectivity for policy instruments, thus rendering a new idea for studying entrepreneurship policies and more generalized policy instruments.

However, the accuracy of the classification model of policy instruments has not been further discussed in this

study. On the one hand, the policy text is divided into the units of analysis of policy instruments according to paragraphs, which is rather crude. Since the structure of policy texts issued by different publishing agencies is quite different, it is difficult to achieve accurate segmentation of policy instruments through a common rule for all policy texts. In this study, policy instruments are segmented according to newline characters, and this simple segmentation method will have a certain impact on the prediction results. On the other hand, in view of the category imbalance problem in entrepreneurial policy tools, this study does not further discuss the optimization of loss function. Therefore, the follow-up study will try to continue in large-scale policy text corpus BERT model training and optimization, to increase the BERT model related knowledge in the field of policy. In order to further improve the efficiency of policy text classifier, the division of policy instruments analysis unit and the imbalance of policy tool categories are optimized.

Data Availability

The data for this study are obtained upon request.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Social Science Fund of China (18BJL39) and Key Course Project of Shanghai Education Commission (s202108002).

References

- [1] N. D. William, *An Introduction to the Analysis of Public Policies*, China Remin University Press, Beijing, China, 2002.
- [2] J. Zhang, *A Study on Building Public Services in the New Countryside from the Domain of Market-Based Policy Instruments*, Guangxi Normal University, Guilin, China, 2011.

- [3] M. Howlett, "Government communication as a policy tool: a framework for analysis," *Canadian Political Science Review*, vol. 3, no. 2, pp. 23–37, 2009.
- [4] Y. A. Zhang, Z. Geng, and Y. N. Wang, "Regional scientific and technological innovation policy classification and policy instrument mining: a research based on the data of Zhong-guancun," *Science & Technology Progress and Policy*, vol. 32, no. 11, pp. 116–122, 2015.
- [5] M. Huang, G. L. Zhu, and J. Y. Hu, "On classification, selection and evaluation of innovation policy instruments," *Forum on Science and Technology in China*, vol. 1, pp. 26–30, 2016.
- [6] C. X. Zhou, M. J. Li, and H. Lin, "Research of regional strategic emerging industry policy analysis framework," *Studies in Science of Science*, vol. 35, no. 3, pp. 346–353, 2017.
- [7] K. S. Kalyan, A. Rajasekharan, and S. Sangeetha, "Ammus: A Survey of Transformer-Based Pretrained Models in Natural Language Processing," 2021, <https://arxiv.org/abs/2108.05542>.
- [8] Y. S. Kim, H. J. Song, and J. Han, "A deepfake-based deep learning algorithm for medical data manipulation detection," *Journal of System and Management Sciences*, vol. 12, no. 1, pp. 13–24, 2022.
- [9] C. M. Dumitrescu and I. Dumitrache, "Combining deep learning technologies with multi-level gabor features for facial recognition in biometric automated systems," *Studies in Informatics and Control*, vol. 28, no. 2, pp. 221–230, 2019.
- [10] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pretrained models for natural language processing: a survey," *Science China Technological Sciences*, vol. 63, no. 10, pp. 1872–1897, 2020.
- [11] K. Kowsari, K. Jafari Meimandi, and M. Heidarysafa, "Mendu , Barnes , Brown Text classification algorithms: a survey," *Information*, vol. 10, no. 4, p. 150, 2019.
- [12] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 273–292, 2019.
- [13] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé, "Deep unordered composition rivals syntactic methods for text classification," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 1681–1691, Beijing, China, July 2015.
- [14] T. Song, J. Sun, Y. Zhang, W. Peng, and J. Song, "An RNN model for generating sentences with a desired word at a desired position," *Tehnicki vjesnik-Technical Gazette*, vol. 27, no. 1, pp. 81–88, 2020.
- [15] J. W. Choi and E. R. Jeong, "Genicular artery embolization: beyond the placebo effect, and planning for the long road ahead," *Journal of Vascular and Interventional Radiology: Journal of Vascular and Interventional Radiology*, vol. 33, no. 1, pp. 11–13, 2022.
- [16] M. A. Kilvisharam, S. Poruran, and M. Y. Caffiyar, "A novel deep convolutional neural network architecture based on transfer learning for handwritten Urdu character recognition," *Tehnicki vjesnik-Technical Gazette*, vol. 27, no. 4, pp. 1160–1165, 2020.
- [17] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," 2017, <https://arxiv.org/abs/1706.03762>.
- [18] M. E. Peters, M. Neumann, M. Iyyer et al., "Deep contextualized word representations," 2018, <https://arxiv.org/abs/1802.05365>.
- [19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: pretraining of deep bidirectional transformers for language understanding," 2018, <https://arxiv.org/abs/1810.04805>.
- [20] S. Y. Lin, Y. C. Kung, and F. Y. Leu, "Predictive intelligence in harmful news identification by BERT-based ensemble learning model with text sentiment analysis, Information Processing & Management: Libraries and Information Retrieval Systems and Communication Networks," *An International Journal*, vol. 2, no. 59, 2022.
- [21] V. D’Orazio, S. T. Landis, G. Palmer, and P. Schrod, "Separating the wheat from the chaff: applications of automated document classification using support vector machines," *Political Analysis*, vol. 22, no. 2, pp. 224–242, 2014.
- [22] R. R. Krebs, "How dominant narratives rise and fall: military conflict, politics, and the cold war consensus," *International Organization*, vol. 69, no. 4, pp. 809–845, 2015.
- [23] C. Chang and M. Masterson, "Using word order in political text classification with long short-term memory models," *Political Analysis*, vol. 28, no. 3, pp. 395–411, 2020.
- [24] R. Pujari and D. Goldwasser, "Understanding politics via contextualized discourse processing," 2020, <https://arxiv.org/abs/2012.15784>.
- [25] A. Mukherjee, S. Bhattacharyya, K. Ray, B. Gupta, and A. K. Das, "A study of public sentiment and influence of politics in COVID-19 related tweets," *Computational Intelligence in Pattern Recognition*, pp. 655–665, Springer, Singapore, 2022.
- [26] X. Wang, B. Guo, and L. Su, "Analysis on the evolution of Shanghai’s science and technology policy based on content analysis," *Science and technology progress and countermeasures*, vol. 29, no. 23, pp. 104–107, 2012.
- [27] Y. Li, S. Wu, and F. Gao, "Changes, evaluation and trend of research funds management policy in China since the reform and opening up -- based on the content analysis of policy text," *Studies in science of science*, vol. 27, no. 10, pp. 1441–1447, 2009.
- [28] Y. Liu, M. Ott, N. Goyal et al., "Roberta: a robustly optimized bert pretraining approach," 2019, <https://arxiv.org/abs/1907.11692>.