Hindawi
Advances in Multimedia
Volume 2022, Article ID 6540696, 11 pages
https://doi.org/10.1155/2022/6540696

Research Article

Ptr4BERT: Automatic Semisupervised Chinese Government Message Text Classification Method Based on Transformer-Based Pointer Generator Network

Mingxin Li,[1] **Kaiqian Yin**,[2] **and Minghao Wang**[3]

[1]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[2]*College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao 266590, China*
[3]*College of Mechanical and Electronic Engineering, Shandong University of Science and Technology, Qingdao 266590, China*

Correspondence should be addressed to Mingxin Li; mingxinli@sdust.edu.cn and Kaiqian Yin; ykqccc@sdust.edu.cn

Received 25 May 2022; Revised 19 July 2022; Accepted 25 July 2022; Published 27 August 2022

Academic Editor: Mohammad R. Khosravi

With the development of Internet technology, government affairs can be handled online. More and more citizens are using online platforms to report to government departments, which is generating a lot of textual data. Among them, the basic but important problem is to automatically classify the different categories of messages, so that staff from different departments can process relevant information quickly. However, government messages have problems such as fast update rate, a large amount of information, long texts, and difficulty in capturing key points, which make supervised learning methods unsuitable for processing such texts. To address these problems, we propose a semisupervised text classification method based on a transformer-based pointer generator network named Ptr4BERT, which uses the pointer generator network with BERT(bidirectional encoder representation from transformers) embedding as a preprocessor for feature extraction. In this method, text classification can achieve very good results with a small set of labeled data, by extracting features exclusively from the message text. In order to verify the effect of our proposed model, we performed some experiments. Besides, we designed a crawler program and obtained two datasets from different websites, which are named HNMes and QDMes. Experimental results have shown that the proposed method outperforms the state-of-the-art methods significantly.

1. Introduction

A public opinion survey is a scientific survey method, which has the functions of reflecting public opinions, making references, and testing the effectiveness of policies. With the development of the Internet, people can express their problems on Weibo, WeChat, and other Internet platforms, which leads the number of messages rapidly increasing. It needs a lot of time for staff to classify the message. Therefore, establishing an automatic and high efficient text classification algorithm is the most basic and important work to improve the management level of government departments and the efficiency of problem-solving.

There are a lot of works using text classification algorithms, but these methods cannot play a good effect in the text classification of government messages. Because there are many differences between government messages and paper texts or news texts: (1) government messages are left by nonprofessional citizens. The content of messages is colloquial, and the text content is difficult to highlight the key points. Therefore, the traditional method based on professional terms is difficult to achieve good results. (2) The text content of messages is complex. Sometimes, there are words that are not in the same field. For example, the words in the text about commerce occasionally appear in the text about transportation, so we need to find the key content of the message. (3) The message length is different. Some messages are less than 50 words, but many are more than 100 words. The different length of the message is a great challenge to the accuracy of the classifier. For example, TextCNN can process short texts, but its classification accuracy of long messages declines sharply.

It needs us to make great efforts to improve the performance in the government message text classification scenario. Recently, some supervised methods and unsupervised methods have tried to solve the problem. However, there are still some limitations in the existing studies. On the one hand, supervised methods require a lot of labeled corpus to learn well, which is expensive and not suitable for large-scale corpus; on the other hand, although unsupervised methods do not need to label corpus manually, their performance is far from that of supervised methods at present.

To solve these problems, we propose a semisupervised text classification method based on BERT and PGN. We can automatically extract the features of long government messages using PGN, which can reduce the impact of noise data without the manual. Experimental results show that our method can achieve an amazing result only using 20% labeled data. Our method is significant in comparison with competitive supervised learning methods. In conclusion, the main contributions of this paper are as follows:

(1) we introduce a transformer-based pointer generator network for summary extraction of government messages. Through summary extraction, features of message text can be extracted automatically instead of traditional manual feature extraction methods.

(2) We propose a semisupervised method based on BERT for government message text classification.

(3) We use pointer generator networks to extract summaries from the original government messages and input them into our proposed semisupervised learning method. This method converts long texts into short texts and solves the problem of varying lengths of message texts.

(4) Experimental results generally require data from the real world. Instead of using public datasets, we designed a crawler program to crawl two groups of real datasets, named HNMes and QDMes. These experimental results prove our model exceeds the baseline, which is of great significance.

The remainder of this paper is organized as follows: the related studies are briefly introduced in Section 2. The proposed method is presented in Section 3. The extensive experimental results and discussions are given in Section 4. Finally, we summarize the paper and indicate the future work in Section 5.

2. Related Work

2.1. Pointer Generator Networks. In recent years, pointer generator network has been widely used in the natural language process field. Shobana and Murali [1] proposed a method for abstracting comments based on pointer generator network with an attention mechanism. It uses an abstract way to generate abstracts that more accurately summarize the abstracted comments. Kumar et al. [2] proposed a cognitive-based opinion summarizer called FP2GN. It selectively focuses on thematic and contextual cues with a multiheaded selfattention mechanism. Liuet al. [3] proposed a new sequence-to-sequence (Seq2Seq) framework. It connects transformers and pointer networks to jointly extract entities and relations, which helps the encoder to focus more accurately on syntactically relevant words. Hao et al. [4] proposed DeepDepict to solve the product information description problem. It uses a multi-pointer generator network to fuse data to generate information-rich and personalized product descriptions. Huang et al. [5] proposed a pointer generator network based on entity relations. They investigated the value of entity relations for improving the performance of the Seq2Seq abstract text summarization model. However, there are few methods to use the pointer generator network for feature extraction of text yet.

2.2. Text Classification Methods. Text classification has always been a key problem and challenge in natural language processing. In general, text classification methods are mainly supervised learning methods, unsupervised learning methods, and semisupervised learning methods.

Supervised methods usually use statistical methods to evaluate each category and use classification functions to select the category with the highest probability as the result. Representative methods include naïve Bayes [6, 7], logistic regression [8], decision tree [9], deep neural networks [10–12], and so on.

Unsupervised methods do not need us to label corpus manually; but using clustering methods, we can find similar messages in the same category. Haj-Yahia et al. [13] solved the problem of data marking by using expert knowledge and word embedding of unsupervised text classification. Yu and Ma [14] proposed a method based on K-nearest neighbors (KNN), which can be effectively applied to datasets with complex distribution without labels.

Recent years have witnessed that semisupervised methods can achieve similar performance as supervised methods but require a very small number of labeled datasets. Thus, the semisupervised methods have received a great deal of attention due to their advantages. For example, Linmei et al. [15] used a heterogeneous graph attentional neural network for semisupervised learning, which achieved very good results in short text classification. Zhao et al. [16] improved the traditional naive Bayes classifier to achieve semisupervised text classification. Severin et al. [17] surpassed the random forest model by extracting keyword features.

For message text classification, Gustavo et al. [18] conducted text classification of microblog messages. Shafiq et al. [19] proposed a text classification method for WeChat messages. However, there are few methods for government message text classification. Our method improves feature extraction and makes it more suitable for government message text classification. What's more, since our features are derived from text, without introducing other information such as a knowledge graph, our method can be extended to different languages.

3. Proposed Method

3.1. The Framework. The framework is shown in Figure 1. We combined the feature extraction method and text classification to design an end-to-end model. For preprocessing, we introduce transformer embedding to pointer generator network instead of single Word2vec to extract features of government messages. The method can extract the main words of each message and it has high confidence in the text summarize task, which can automatically extract the text features. The details are proposed in Section 3.3.

Besides, we proposed a semisupervised learning text classification method based on Bidirectional Encoder Representation from Transformers (BERT). We can see that semisupervised learning can not only ensure the accuracy of text classification but also reduce the workload of manual annotation corpus. The details of the semisupervised method with analysis are presented in Section 3.5.

3.2. Data Process. To enable text data to be computed in Euclidean space and reduce the influences of useless features, we do the following works.

(1) Since Chinese words are not separated by spaces, we use a library called jieba (https://github.com/fxsjy/jieba) to perform word segmentation on the text and convert sentences into word sets.

(2) In natural language, there are a large number of words such as "a" and "is," which are called "Stop Words." These words are not only redundant but also cause problems such as reduced model accuracy and so on. Thus, we write a Python data processing program in order to remove stop words from the original text.

(3) We use Word2vec [20] to embed the original text so that we can calculate texts in Euclidean space.

(4) In model training, we used eight Intel Xeon E5 CPUs and two GeForce RTX 2080 Ti GPUs.

3.3. Transformer-Based Pointer Generator Network for Feature Extraction. Choosing a suitable method for text feature extraction is one of the important tasks in text classification work. However, the quality of text features is also the key to determining the performance of text classification models. In order to solve the problems in the text of government messages and improve the text quality, in this section, we propose a transformer-based pointer generator network model as shown in Figure 2. This method uses text summarization for text feature extraction. Experiments prove that this model is more concise and effective than existing competitive text feature extraction methods.

Unlike the text summarization task, our task does not require text generation for final distribution but rather converts the id into the corresponding embedding, which is used as the direct input to the classifier's input. Our method achieves end-to-end learning.

To improve the effect of feature extraction, we use the BERT embedding method [21]. Unlike Word2vec, the BERT embedding method fuses word embedding, segment embedding, and position embedding [22]. Thus, the input information has word information, position information, and sentence vector representation. This method can reduce the problem of periodicity of LSTM to some extent and can improve the effectiveness of feature extraction. The formula of BERT embedding is defined as follows:

$$\mathrm{PE}_{(\mathrm{pos},2i)} = \sin\left(\frac{\mathrm{pos}}{10000^{2i/d_{\mathrm{model}}}}\right), \tag{1}$$

$$\mathrm{PE}_{(\mathrm{pos},2i+1)} = \cos\left(\frac{\mathrm{pos}}{10000^{2i/d_{\mathrm{model}}}}\right), \tag{2}$$

$$\mathrm{FE}_{\mathrm{pos}} = \mathrm{PE}_{\mathrm{pos}} + \mathrm{WE}_{\mathrm{pos}} + \mathrm{SE}_{\mathrm{pos}}, \tag{3}$$

where pos is the position and i is the dimension. To ensure the dimension of word embedding is the same as the position embedding, d_{model} is the dimension of the word vector, which facilitates the fusion of vectors. In addition, we define the position embedding of pos position as $\mathrm{PE}_{\mathrm{pos}}$, the word embedding of pos position as $\mathrm{WE}_{\mathrm{pos}}$, and the segment embedding of pos position as $\mathrm{SE}_{\mathrm{pos}}$ The embedding after the fusion of the two is defined as $\mathrm{FE}_{\mathrm{pos}}$.

We use pointer generator network as a baseline [23] for feature extraction of message text. Following pointer generator network, the feature extraction of government message text follows the following formula:

$$\begin{aligned} e_i^t &= v^T \tanh\left(W_h h_i + W_s s_t + b_{\mathrm{attn}}\right), \\ a^t &= \mathrm{softmax}\left(e^t\right), \\ h_t &= \sum_i a_i^t h_i, \\ P_{\mathrm{vocab}} &= \mathrm{softmax}\left(V'\left(V\left[s_t, h_t\right] + b\right) + b'\right), \end{aligned} \tag{4}$$

where $W_h, W_s, b_{\mathrm{attn}}, v$ is the learnable parameter, a_t is the attention distribution, and h_t is the context vector of the source text. Besides, P_{vocab} is the feature of the context vector and V, V', and b' is a learnable parameter.

What's more, the generation probability $p_{\mathrm{gen}} \in [0, 1]$ at time step t can be computed as the following formula:

$$p_{\mathrm{gen}} = \sigma\left(w_h^T h_t + w_s^T s_t + w_x^T x_t + b_{\mathrm{ptr}}\right), \tag{5}$$

where w_h^T, w_s^T, w_x^T, and b_{ptr} are all the learning parameters of the model and σ is the sigmoid function,

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{6}$$

In addition, we fuse the relevant features of the vocabulary with the text content to form the new distribution, and the method is,

$$P(w) = p_{\mathrm{gen}} P_{\mathrm{vocab}}(w) + \left(1 - p_{\mathrm{gen}}\right) \sum_{i:\, w_i = w} a_i^t. \tag{7}$$

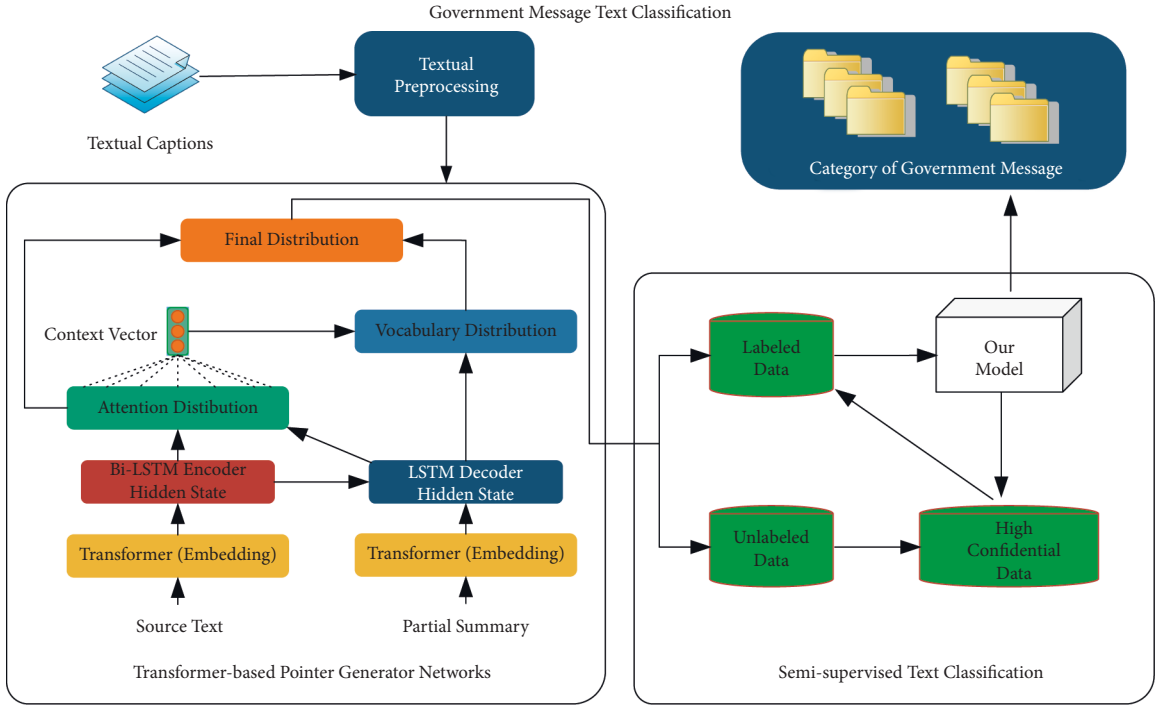

Figure 1: Framework for feature extraction and the semisupervised method for government message text classification.

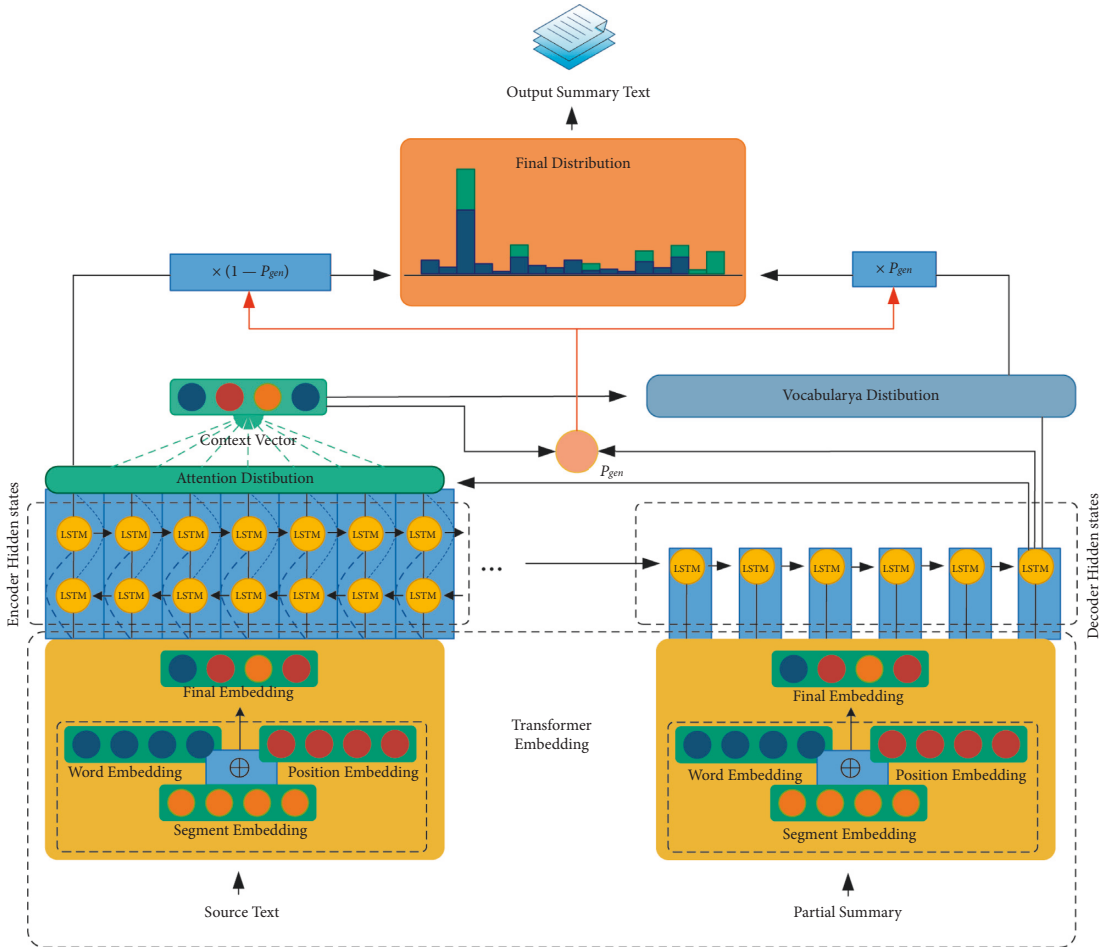

Figure 2: Architecture of the feature extraction model and transformer-based pointer generator network, for government message, preprocessing. We get the embedding of the final text and input it into the text classification method.

If the word is not in vocabulary distribution, $P_{\text{vocab}}(w)$ is zero. If the word does not appear in the source text, $\sum_{i:w_i=w} a_i^t$ is zero.

The BERT embedding pointer generator network can solve the problem of out-of-vocabulary, which has a very important role in text classification. It avoids the appearance of unknown words and extracts more important feature information. In addition, the problem of periodicity in LSTM can be reduced to some extent due to the addition of location information. Thus, more accurate long text features can be better obtained. Moreover, like PGN, the coverage mechanism avoids duplicate text, improves the quality of features, reduces the text length, and improves the efficiency of subsequent text classification.

3.4. BERT-Based Layer for Text Classification. In feature extraction, we use BERT as the method of embedding. In the text classification task, we process the vector output from PGN as input into the BERT model [21] in the same way as equation (3). After the BERT model, its output information is T vector. The framework is shown in Figure 3.

Traditional natural language processing methods use temporal methods to sequence processing and use attention mechanisms to determine which words we pay more attention to. The BERT model abandons the temporal method and uses a selfattention mechanism for words, which assigns different weights to different words without supervision and can better represent the characteristics between different words.

After the BERT model, the output T vector is processed. T vector will correspond to the output for each input. But since our task is text classification, we input the T vector into a fully connected layer and use the softmax equation (8) to predict the category.

$$p_i = \frac{\exp(y_i)}{\sum_{k=1}^{n} \exp(y_k)}, \tag{8}$$

where p_i is the probability of different categories and the highest probability is the category of input text.

3.5. Semisupervised Learning Method for Government Message Text Classification. In the semisupervised method, we first need to train the BERT-based layer with labeled data. We then predict the labels of the unlabeled dataset and add the high confidence data to the labeled dataset to train the BERT-based layer again. Repeat the process until all unlabeled data are labeled. Therefore, in this section, we will introduce how to select unlabeled training sets with high confidence in semisupervised learning.

After training the BERT-based layer with a labeled training set first, we then predict the unlabeled data using the model. Moreover, select a set of data with high confidence and iteratively add to the labeled training set until all training datasets have been labeled.

Since the datasets are government messages, they exist in the form of sentences. After preprocessing, we can obtain the vector representation of each sentence, and the cosine

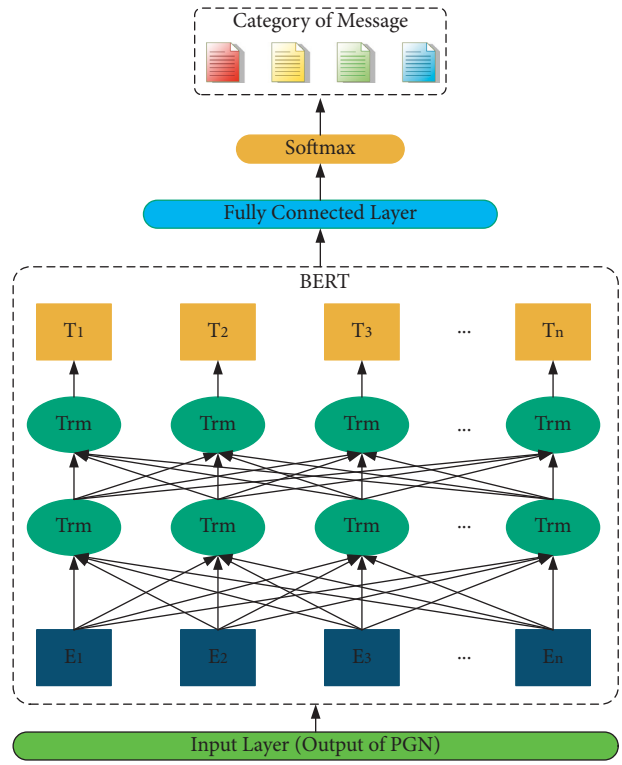

Figure 3: Text classification model based on BERT.

distance between sentence vectors is used as a method to determine the high confidence data. The calculation formula is as follows:

$$D(V_i, V_j) = 1 - \frac{V_i \cdot V_j}{\|V_i\| \|V_j\|} = 1 - \frac{\sum_{\alpha=1}^{m} V_i^{\alpha} \times V_j^{\alpha}}{\sqrt{\sum_{\alpha=1}^{m} (V_i)^2} \times \sqrt{\sum_{\alpha=1}^{m} (V_j)^2}}, \tag{9}$$

$$\text{s.t. } V_i \hat{\text{I}} \text{Training Set}, \; V_j \hat{\text{I}} \text{TestSet},$$

where V_i and V_j are vector representations of sentence i and sentence j respectively, m is the dimension of a vector, and V_i^{α} and V_j^{α} represent the components of V_i and V_j, respectively.

In the iterative training process, we will choose the unlabeled sample that is most similar to the labeled sample every time. These samples are considered to have high confidence. Algorithm 1 shows the details of the proposed semisupervised algorithm based on BERT. Where L is the labeled training set, U is the size of the unlabeled training set, and n is the number of unlabeled samples selected for calculation in each iteration.

4. Experiment

4.1. Datasets Description. We designed a crawler program and crawled two datasets from government affair websites in different cities, which we named HNMes and QDMes,

Input: labeled training set D_L, unlabeled training set D_U, test set D_T
Output: $Y = \{y_i\}$
(1) $index = \text{int}((|D_U| + |D_L| + |D_T|)/|D_L|) + 1$;
(2) $num = \text{int}(|D_U|/index)$;
(3) for $i = 1$: $index$ do
(4) Training the model with labeled data;
(5) Calculate the sentence vector;
(6) for $j = 1$: num do
(7) $x^* = \text{argmin}_{x \in D_U} CosineDistance(x, D_L)$;
(8) Predicting the label y^* of x^*;
(9) $D_L = D_L \cup \{(x^*, y^*)\}$.;
(10) end
(11) end
(12) BERTModel = BERTTrain(D_L);
(13) Y = Predict(BERTModel, D_T);
(14) Return Y;

Algorithm 1: Semisupervised algorithm based on BERT.

respectively. To keep the data consistent, each dataset has 7 categories. They are urban and rural construction, environmental protection, transportation, education, trade, medical care, and social security.

The information of datasets is shown in Table 1. In addition, all data are randomly shuffled before they are learned and validated. Five-cross-fold validation is adopted.

4.2. Performance Metrics. In our experiment, we adopt 3 metrics, precision, recall, and F1-measure, to measure the results. Their formal definitions are shown in equations (10)–(12):

$$\text{Precision} = \frac{N_{\text{correct}}}{N_{\text{output}}}, \tag{10}$$

$$\text{Recall} = \frac{N_{\text{correct}}}{N_{\text{manual}}}, \tag{11}$$

$$\text{F1} - \text{Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{12}$$

where N_{correct} is the number of correct concepts automatically extracted by the algorithm, N_{output} is the number of all concepts automatically extracted by the algorithm, and N_{manual} is the number of concepts that are manually labeled.

4.3. Baselines. We compare the proposed method with several baselines used to extract concepts. The following is a brief introduction to them.

Logistic regression is a classical classification method in statistical learning. Its regression formula is established for classification boundary, which is based on the existing data. It uses optimization methods to find the best fitting parameters. Fu et al. [24] used logistic regression to classify texts and achieved good results.

Naïve Bayes is a classification method based on probability and statistics, mainly using Bayes theorem and characteristic condition independence hypothesis. The model's algorithm logic is relatively simple and has very robust performance in text classification [25].

Table 1: Description of datasets.

Dataset	Language	Trainset	Testset	Category	Token
HNMes	Chinese	100,000	7,000	7	53,428,396
QDMes	Chinese	40,000	7,000	7	17,283,744

Support vector machine (SVM) [26] is a famous classification model. The main idea of this method is to find a hyperplane in space that is more suitable for dividing all data samples and to make the distance between all data in the sample set and this hyperplane the shortest. In addition, it can be converted into multiple classifiers using the softmax function.

TextCNN (Convolutional Neural Networks for Text Classification) is proposed by Chen [27]. Compared with the traditional method, it has a strong ability to extract shallow features of the text. Moreover, it can extract important features more efficiently, which occupies an important position in classification.

TextRNN (Recurrent Neural Networks for Text Classification) [28] is an application of RNN (recurrent neural networks) in natural language processing. It is good at capturing longer sequence information and can play an important role in text classification. Traditional RNN is easy to lose the gradient when processing long text, so we generally use long short-term memory (LSTM) for text classification.

TextRCNN (Recurrent Convolutional Neural Networks for Text Classification) is proposed by Lai et al. [29]. It combines the advantages of CNN and RNN and can handle not only long-text data but also short-text data.

BERT (Bidirectional Encoder Representation from Transformers) is proposed by Devlin et al. [21]. It uses a transformer as the baseline, and instead of using the traditional one-way language model for pretraining as

in the past, it uses the new masked language model so that it can generate deep bidirectional language representations.

ERNIE (Enhanced Representation through Knowledge Integration) is proposed by Sun et al. [30]. It is further optimized in the BERT model to model lexical structure, syntactic structure, and semantic information in the training data in a unified way, which greatly enhances the generic semantic representation.

4.4. Experimental Results and Analysis. In order to prove the effectiveness of our model, we divided the baselines into supervised learning methods and semisupervised learning methods. After comparison, our method is excellent.

In supervised learning methods, all training sets are labeled. However, in semisupervised learning methods, only 20% of the training sets are labeled, and data with high confidence should be selected to be added to the training set for iteration.

As shown in Table 2, the semisupervised learning method we proposed outperforms all competing baseline semisupervised models. For traditional supervised machine learning methods, the effect of our method is 22.3%-25.2% higher than logistic regression, 12.9%-15.4% higher than naive Bayes, and 6.8%-12.2% higher than SVM. And for a method of deep learning, our method also has a good performance. The performance of our method is 3.8%-4.6% higher than traditional TextRNN, 4.0%-5.3% higher than TextCNN, and 4.1%-4.3% higher than TextCNN. Besides, our method outperforms BERT and ERNIE by about 1%. The confusion matrix of the classification is shown in Figure 4.

It can be seen that the traditional machine learning method is not good compared with the deep learning method. Since traditional machine learning methods only consider statistical information but ignore the features of data and other content, it is easy to overfit. Although SVM is also a traditional machine learning method, it adds core and margin to reduce the problem of overfitting and improve the accuracy of text classification. Besides, deep learning methods can achieve a good index under supervised learning with a large amount of training data using neural networks. While the semisupervised learning method only uses a small amount of data, it cannot learn the features of data well. By extracting the feature using pointer generator network, we can achieve a surprising effect with a very small amount of data. Our method Ptr4BERT has the best performance compatibility and can greatly reduce the waste of human resources due to manual marking data and summarizing messages.

4.4.1. Ablation Experiments. To demonstrate that each component of our model has a positive effect on improving text classification accuracy, we designed three sets of experiments named Ptr4BERT-T, Seq4BERT, and semi-BERT. Experiments were conducted and analyzed by removing some components from the model. The experiment results are shown in Table 3.

Table 2: Performance comparison of different methods.

Method	HNMes			QDMes		
	P	*R*	F1	*P*	*R*	F1
Logistic regression	72.11	71.70	71.91	68.69	68.15	68.41
Naïve Bayes	81.42	79.46	80.43	77.95	78.86	78.40
SVM	84.67	82.19	83.41	86.54	84.39	85.45
TextRNN	90.02	90.06	90.04	89.53	88.82	89.17
TextCNN	90.32	89.27	89.79	88.07	88.81	88.93
TextRCNN	90.14	90.07	90.10	89.26	89.19	89.22
BERT	93.87	93.14	93.50	92.18	91.62	91.89
ERNIE	93.52	92.71	93.11	92.76	92.01	92.38
Ptr4BERT (ours)	94.39	94.38	94.39	93.36	93.39	93.37

(i) Ptr4BERT-T: this model removes the BERT embedding method and replaces it with origin Word2vec method. This variant is designed to investigate the effectiveness of the preprocessing method without BERT embedding.

(ii) Seq4BERT: this model replaces the pointer generator network with seq2seq, which do not need to calculate the distribution described in equations (3) and (5) This variant is designed to investigate the effectiveness of processing out-of-vocabulary words and the coverage mechanism.

(iii) Semi-BERT: this model removes the preprocessing method and only uses the semisupervised BERT method. This variant is designed to investigate the effectiveness of the automatic feature extraction method.

The results of the ablation experiments show that the accuracy of the model decreases regardless of the removal of any component. In addition, when the PGN is removed, the model can only analyze the long text of the message, which has more noise in the long text, reducing the accuracy of the model by about 2%. However, semi-BERT is slightly lower than the accuracy of the BERT model in Table 2. The reason for this is that although semisupervised learning requires less labeled data, the lack of labels in the iterations will have an impact on the accuracy of the model and cannot achieve the same high accuracy as supervised learning.

4.4.2. Efficiency of Semisupervised Learning Method. In order to study the performance of the semisupervised learning methods we proposed, we propose a supervised learning method, which has all the process methods defined in Section 3.3. It is a supervised learning version of Ptr4BERT. We use HNMes to compare the efficiency of both methods, and the result of different sizes of data is shown in Figure 5.

We can see that the semisupervised learning method is significantly better than the supervised learning method. When only 20% of the data are labeled, the semisupervised learning method can achieve better performance than the supervised learning method. It means that semisupervised learning methods can achieve good performance with less labeled data.

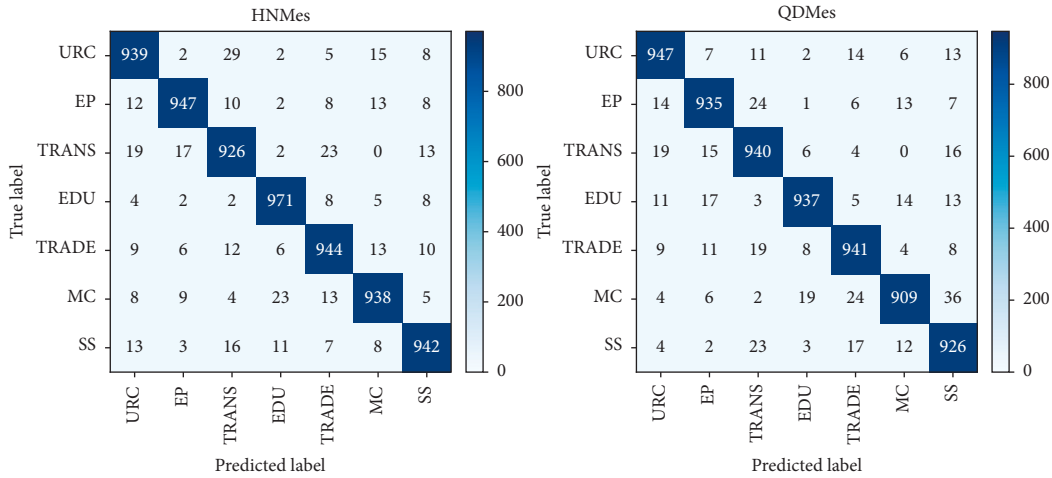

Figure 4: Confusion matrix after classification of HNMes and QDMes, where URC stands for urban and rural construction, EP stands for environmental protection, TRANS stands for transportation, EDU stands for education, TRADE stands for trade, MC stands for medical care, and SS stands for social security.

Table 3: Performance of ablation experiments.

Method	HNMes			QDMes		
	P	*R*	F1	*P*	*R*	F1
Ptr4BERT-T	94.08	94.01	94.05	92.72	92.51	92.62
Seq4BERT	93.20	93.08	93.15	92.13	92.40	92.26
Semi-BERT	92.82	92.73	92.77	91.88	91.79	91.83
Ptr4BERT (Ours)	94.39	94.38	94.39	93.36	93.39	93.37

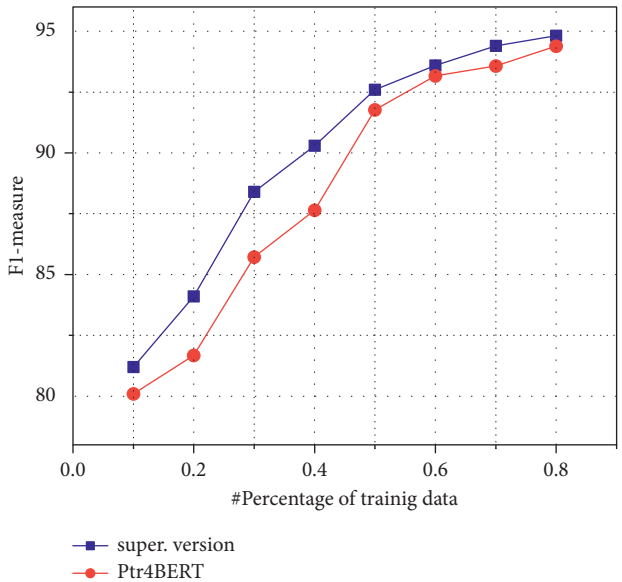

Figure 5: Performances of the proposed Ptr4BERT and its supervised version.

4.4.3. Influence of Embedding Dimension. Obviously, different dimensions of the vector have a certain influence on the result. We made a comparison in the five dimensions of [32, 64, 128, 256, and 512]. The influence of vector dimension on the classification is shown in Figure 6.

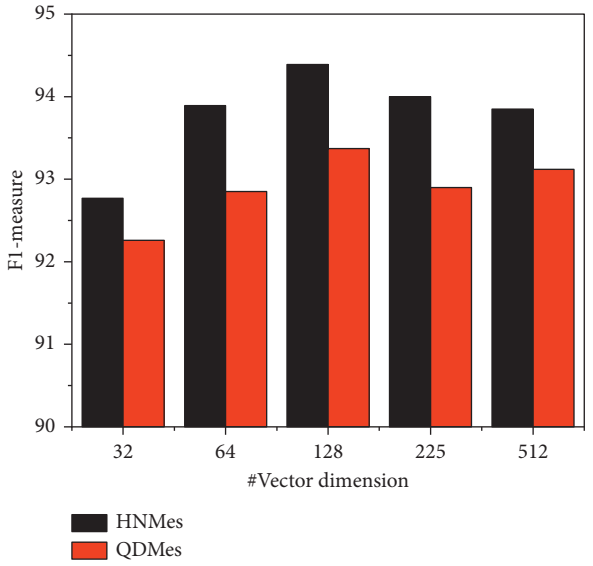

Figure 6: Performance of different vector dimensions.

It can be seen that the F1-measure of the two datasets can reach the highest when the word vector dimension is 128. The main reason is that when the dimension is too small, word features are lost too much. Besides, when the dimension is too large, the vector learns too many redundant features. Therefore, we set the dimension to 128, which can achieve the best result.

4.4.4. Influence of Iteration. It is important to determine the number of iterations. For example, *m* represents the number of unlabeled samples in each iteration. When the number of iterations is 2, *m* accounts for 50% of unlabeled data, and when the number of iterations is 4, *m* accounts for 25% of unlabeled data. Therefore, we evaluate the influence of the number of iterations on the results.

As shown in Figure 7, when the number of iterations is 2, 4, and 6, respectively, the performance of our method gets better in both the datasets. But when the number of

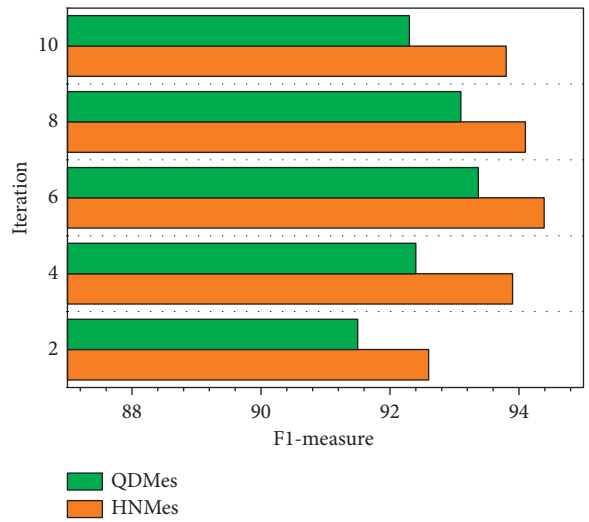

Figure 7: Influence of iterations on the experimental results.

Table 4: Description of English datasets.

Dataset	Language	Trainset	Testset	Category	Token
govMes1	English	20,000	3,000	6	1,036,902
govMes2	English	15,000	1,200	6	859,686

Table 5: Performance comparison of different methods in English dataset.

Method	govMes1			govMes2		
	P	*R*	F1	*P*	*R*	F1
SVM	81.26	80.21	80.73	79.25	78.32	78.78
TextRNN	86.11	87.23	86.66	86.12	85.36	85.74
TextCNN	85.66	85.27	85.46	84.07	85.72	84.89
BERT	89.26	88.59	88.92	87.21	88.04	87.62
Ptr4BERT (Ours)	90.83	89.57	90.20	90.12	89.86	89.99

iterations increases to 8 and 10, the performance of our method decreased.

The main reason is that when the number of iterations is small, each iteration will add more unlabeled data to the training set. Therefore, some unlabeled data with low similarity to the labeled dataset will be added to the training set, which will reduce the performance of the model. In addition, as the number of iterations increases, the data added to the training set decrease. So, fewer features are learned in each iteration. In short, we set the number of iterations of the semisupervised learning method to 6.

4.4.5. Effects in Different Languages. In fact, our method works well in other languages as well, not only Chinese but also English. To prove that our method can work well in different languages, we crawled two datasets of government messages from different states in the United States. Among them, the data are mainly divided into six categories: COVID-19, Climate, Racial Equity, Economy, Health Care, and Immigration, which we named govMes1 and govMes2. The specific information of the two groups of data is shown in Table 4.

Since we use transformer-based encoding, word vector training can be performed for both Chinese and English, and there is no major difference in the experimental process. We use SVM, TextRNN, TextCNN, BERT, and our method for comparison, respectively, and the comparison results are shown in Table 5.

Experiments show that the F1-score of our method has 10%-11% higher than traditional machine learning methods (SVM), 4%-6% higher than deep learning methods (TextRNN and TextCNN), and 1%-2% higher than language models with pretraining (BERT). It can be seen that our method maps short and long texts to short text space, which can better solve the problems such as long-text classification feature redundancy and improve the accuracy and robustness of the model.

4.4.6. Result Visualization. In order to display the results more clearly, we selected a small amount of data in the HNMes dataset and wrote a Python program to visualize the result. As shown in Figure 8, we show the labels of the original data, the data obtained after feature extraction, and

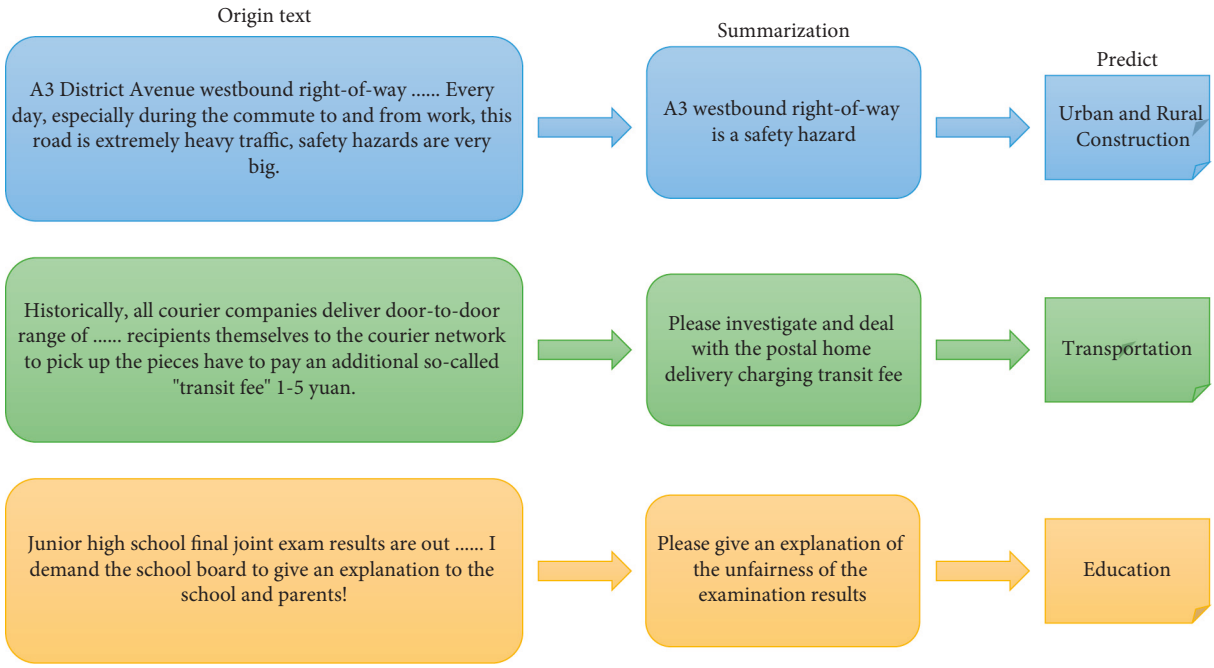

Figure 8: Visualization of results. The left column is the original text content, the middle column is the summarization of the original text, and the right column is the category predicted by our method.

the data obtained after prediction. Since our method is an end-to-end approach, the original data after feature extraction are obtained by adding a decoder layer to decode the vectors.

5. Conclusion and Future Work

This paper studies the text classification of government messages. We propose a semisupervised text classification method based on BERT and PGN. We can automatically extract the features of long government messages using PGN, which can reduce the impact of noise data without the manual. With semisupervised learning, we can achieve amazing results with very small datasets. In order to demonstrate the effectiveness of our method, we tested it in two real-world datasets. In both datasets, our results exceed baseline models. Experimental results have shown that the proposed method outperforms the state-of-the-art methods significantly.

Government message text classification can improve the work efficiency of government staff and make staff better reply to the masse's opinions and suggestions. In the next step, we will use the knowledge graph to make message text classification more accurate. In addition, we will use hot word extraction, text clustering, named entity recognition, and other methods to analyze the popularity of messages to improve the degree of automation of the government affairs system.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Mingxin Li and Kaiqian Yin both contribute equally to this paper.

Acknowledgments

The research was supported by Chinese National College Students' Innovation and Entrepreneurship Training Program Project (No. 202110424026) and Shandong College Students' Innovation and Entrepreneurship Training Program Project (No. S202010424058).

References

[1] J. Shobana and M. Murali, "Abstractive review summarization based on improved attention mechanism with pointer generator network model," *Webology*, vol. 18, no. 1, pp. 77–91, 2021.

[2] A. Kumar, S. Seth, and S. Gupta, "Sentic Computing for Aspect-Based Opinion Summarization Using Multi-Head Attention with Feature Pooled Pointer Generator Network," *Cognitive Computation*, vol. 14, pp. 1–19, 2021.

[3] L. Liu, M. Wang, X. He, L. Qing, and J. Zhang, "Extracting relational facts based on hybrid Syntax-Guided transformer and pointer network," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 6, pp. 12167–12183, 2021.

[4] S. Hao, B. Guo, H. Wang et al., "DeepDepict: enabling information rich, personalized product description generation with the deep multiple pointer generator network," *ACM*

Transactions on Knowledge Discovery from Data, vol. 15, no. 5, pp. 1–16, 2021.

[5] T. Huang, G. Lu, Z. Li, J. Song, and L. Wu, "Entity relations based pointer-generator network for abstractive text summarization," *Advanced Data Mining and Applications*, Springer, pp. 219–236, Berlin, Germany.

[6] S. Ruan, B. Chen, K. Song, and H. Li, "Weighted nave Bayes text classification algorithm based on improved distance correlation coefficient," *Neural Computing & Applications*, vol. 34, pp. 1–10, 2021.

[7] S. H. Lu, D. A. Chiang, H. C. Keh, and H. H Huang, "Chinese text classification by the Naïve Bayes Classifier and the associative classifier with multiple confidence threshold values," *Knowledge-Based Systems*, vol. 23, no. 6, pp. 598–604, 2010.

[8] T. P. Jurka and Y. Tsuruoka, "Maxent: an R package for text classification using low-memory multinomial logistic regression," *ChemInform*, vol. 31, no. 35, pp. 595–601, 2012.

[9] S. Bahassine, A. Madani, and M. Kissi, "An Improved Chi-Sqaure Feature Selection for Arabic Text Classification Using Decision Tree," in *Proceedings of the 2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pp. 1–5, IEEE, Mohammedia, Morocco, October 2016.

[10] Z. Tan, J. Chen, Q. Kang, Z. Mengchu, and A. Abdullah, "Dynamic embedding projection-gated convolutional neural networks for text classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 99, pp. 1–10, 2021.

[11] A. Fesseha, S. Xiong, E. D. Emiru, M. Diallo, and A Dahou, "Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya," *Information*, vol. 12, no. 2, p. 52, 2021.

[12] C. Du and L. Huang, "Text classification research with attention-based recurrent neural networks," *International Journal of Computers, Communications & Control*, vol. 13, no. 1, p. 50, 2018.

[13] Z. Haj-Yahia, A. Sieg, and L. A. Deleris, "Towards unsupervised text classification leveraging experts and word embeddings," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 371–379, 2019.

[14] X. Yu and F. Ma, "An Unsupervised Text Classification Algorithm Based on K-Nearest Neighbor," *Journal of the China Society for Scientific and Technical Information*, vol. 27, no. 4, pp. 550–555, 2008.

[15] H. Linmei, T. Yang, C. Shi, and H. Ji, "Heterogeneous graph attention networks for semi-supervised short text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4821–4830, EMNLP-IJCNLP, Hong Kong, 2019.

[16] L. Zhao, M. Huang, Z. Yao, R. Su, Y. Jiang, and X. Zhu, "Semi-supervised Multinomial Naive Bayes for Text Classification by Leveraging Word-Level Statistical Constraint," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2017.

[17] K. Severin, S. Gokhale, and A. Dagnino, "Keyword-based semi-supervised text classification," in *Proceedings of the 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, pp. 417–422, Milwaukee, WI, USA, July 2019.

[18] L. Gustavo, S. Luís, T. Jorge, and O. Eugénio, "Tokenizing micro-blogging messages using a text classification approach," in *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data (AND '10)*, pp. 81–88, Association for Computing Machinery, New York, NY, USA, October 2010.

[19] M. Shafiq, X. Yu, A. A. Laghari, Y. Lu, .K. K. Nabin, and A. Foudil, "WeChat Text and Picture Messages Service Flow Traffic Classification Using Machine Learning Technique," in *Proceedings of the 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 58–62, IEEE, Sydney, NSW, Australia, December 2016.

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

[21] J. Devlin, M. W. Chang, K. Lee, and T. Kristina, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018, https://arxiv.org/abs/1810.04805.

[22] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[23] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with Pointer-Generator Networks," 2017, https://arxiv.org/abs/1704.04368.

[24] Q. Fu, X. Dai, S. Huang, and J. Chen, "Forgetting word segmentation in Chinese text classification with L1-regularized logistic regression," in *Proceedings of the Pacific-asia Conference on Knowledge Discovery and Data Mining*, pp. 245–255, Springer, Berlin, Heidelberg, 2013.

[25] S. Xu, "Bayesian Naïve Bayes classifiers to text classification," *Journal of Information Science*, vol. 44, no. 1, pp. 48–59, 2018.

[26] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[27] Y. Chen, "Convolutional neural network for sentence classification," University of Waterloo, 2015, https://arxiv.org/abs/1408.5882.

[28] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2873–2879, AAAI Press, California, U.S A, July 2016.

[29] S. Lai, L. Xu, and K. Liu, "Recurrent convolutional neural networks for text classification," in *Proceedings of the Twenty-ninth AAAI Conference on Artificial Intelligence*, AAAI Press, California, U.S A, January 2015.

[30] Y. Sun, S. Wang, Y. Li et al., "Ernie: enhanced representation through knowledge integration," 2019, https://arxiv.org/abs/1904.09223.