



Research Article

Automatic analysis of alarm embedded with large language model in police robot

Zirui Liu, Haichun Sun^{*}, Deyu Yuan

School of Information Network Security, People's Public Security University of China, Beijing 100038, China

ARTICLE INFO

Article history:

Received 30 October 2024

Revised 29 January 2025

Accepted 3 February 2025

Available online 14 February 2025

Keywords:

Police robot

Large language models

Few-shot learning

ABSTRACT

Police robots are used to assist police officers in performing tasks in complex environments, so as to improve the efficiency of law enforcement, ensure the safety of police officers and maintain social stability. With the rapid development of science and technology, police robots are widely used in the field of public security, such as alarm reception, patrol, explosive disposal, reconnaissance and so on. However, police robots still have the problem of analysis deviation in the process of receiving the alarm, which leads to the low efficiency of police dispatch. This study aims to enhance the police alarm automatic analysis ability of the police robots to assist in the dispatch of police. In this paper, we propose a novel method (FSTC-LLM) for sample augmentation based on large language model and noise reduction. The experimental evaluations are carried out on the alarm data set and the THUC News data set. The results show that the proposed FSTC-LLM has excellent performance in few shot text augmentation tasks, and can assist police robots to complete the task of automatic analysis of alarm with high quality, which is of great significance to enhance public security.

© 2025 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

In recent years, with the vigorous development of police robot technology, various kinds of robots have emerged, such as the intelligence robot to automatically receive and analyze the alarm, the traffic robot to guide the traffic, and the explosive disposal robot [1] to find and dismantle explosives. In the process of automatically receiving and dispatching police, the intelligence robot needs to realize the human-computer interaction function, alarm analysis and task dispatch. However, due to the difficulty of obtaining a large number of high-quality data in the field of police to meet the training of the model, the trained model has weak generalization ability and the phenomenon of overfitting. As a result, the intelligence robot still has the problem of alarm analysis deviation in the process of automatically receiving and dispatching police. It greatly limits the application of police robots.

In order to improve the ability of automatic alarm analysis, police robots need to use limited sample data to train models to complete the task of accurate alarm classification. To solve the problem of insufficient training data, researchers have proposed various methods. Zhou et al. [2] proposed a data augmentation method, FlipDA, to improve the effect of small sample text classification tasks. Lei et al. [3] proposed the Task-Adaptive Reference Transformation (TART) network to enhance generalization

by transforming class prototypes into per-class fixed reference points in the task-adaptive metric spaces. At present, few-shot learning is mainly divided into three types: based on data augmentation, based on model fine-tuning and based on transfer learning [4]. The known methods have good performance in the field of few-shot learning, but there are still many shortcomings. Firstly, under the specific background of automatic alarm analysis, there are many kinds of alarm text data with sparse semantics, and it is difficult to extract data features. In addition, some data augmentation methods generate a lot of noise while generating new data, which pollutes the training data and leads to low training accuracy.

In order to solve the above problems, we propose a new method FSTC-LLM (Few Shot Text Classification Frame Assisted by Large Language Model) for automatic alarm analysis robot embedded with large language model (LLM), which designs various prompts according to the task of alarm analysis. Fine-tuning the LLM using LoRA generates the required enhanced sample data. In the training process, in order to suppress the interference of the LLM generation sample hallucination, a confidence learning module is designed to reduce the noise of the generated alarm sample data. The FSTC-LLM proposed in this paper only needs a small number of supervised samples to complete the task of automatic identification of alarm with high quality, which solves the problems of difficult acquisition of supervised samples and high cost of manual labeling. The main contributions of this paper are:

^{*} Corresponding author.

E-mail address: sunhaichun@ppsuc.edu.cn (H. Sun).

- (1) We propose a police robot technology of automatic alarm analysis embedded with large language model, which improves the effect of few shot text classification task and enhances the ability of automatic alarm analysis of police robot.
- (2) With the help of the LLMs, the enhanced alarm sample data set is generated. The effect comparison tests of several existing LLMs fine-tuning methods on the task of text classification are completed, and a scheme of instruction fine-tuning of LLMs based on feedback test is proposed.
- (3) In the training process, the idea of confidence learning is introduced to denoise the pseudo-label samples, which solves the problem of output hallucination of the LLMs.
- (4) Two experimental schemes are designed to verify the proposed alarm analysis robot technology. Verify the advancement of the model on the public data set THUC News and the effectiveness of the model on the alarm data set.

2. Related works

2.1. Policing text classification method

Alarm classification is an indispensable component of modern policing. As a crucial element of contemporary policing technologies, police robots can integrate deep learning algorithms with advanced robotic technologies, such as multimodal strain sensing system [5], legged odometry [6], and comprehensive locomotion control in humanoid robots [7]. These robots can not only enhance the efficiency of police operations, but also alleviate the work burden of police officers to some extent. The application of alarm classification in police robots is mainly to help police robots make reasonable judgments and decisions through fast and accurate classification of alarm data, so as to achieve efficient and accurate automatic police dispatch function.

Text classification refers to the process by which a computer maps a text containing information to one or several categories of topics. Early text classification techniques used traditional machine learning methods. With the continuous development of internet technology and the explosion of text data volume, deep learning methods have garnered wide attention from both academia and industry, including Convolutional Neural Networks (CNN) [8], Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) Networks [9]. Guo et al. [10] proposed a legal case classification model RnnTd based on LSTM and tensor decomposition layer. On the basis of TextCNN, Wang et al. [11] introduced an attention mechanism at the input layer and employed a word filtering algorithm to establish the ATextCNN model, specifically designed for multi-class alarm classification tasks. Zhou et al. [12] combined BiLSTM, BiGRU and a convolutional multi-head attention mechanism to identify fraudulent phone text. To further enhance the flexibility and generalization capabilities of the model, researchers proposed the concept of pre-trained language models. For example, Yuan et al. [13] introduced a BERT-RCNN hybrid approach to classify imbalanced Chinese traffic accident texts. After that, a series of LLMs, such as ChatGPT, Alpaca, and ChatGLM have gradually come into people's vision and been applied to police tasks.

2.2. Large language models method

LLMs are computational models capable of understanding and generating human language. These models are typically pre-trained on extensive unlabeled datasets to acquire vast amounts of linguistic knowledge. Subsequent techniques such as instruction fine-tuning, reward modeling, and reinforcement learning are employed to enhance their adaptability to downstream tasks.

LLMs have been widely applied in the field of human-computer interaction [14]. For example, Chung et al. [15] proposed a method for data generation by integrating LLMs with human intervention. Ding et al. [16] evaluated the performance, time, and cost-effectiveness of three different data annotation approaches based on GPT-3. Chen et al. [17] trained models with counterfactually augmented data to capture representations of causal structures within tasks. In the policing domain, Xing et al. [18] leveraged LLMs to extract key entities from police reports.

Although LLMs provide efficient methods for text classification tasks, they still face some difficulties: Firstly, the ability of the professional field is insufficient. Compared with traditional deep learning models, LLMs have stronger adaptability and generalization ability. However, because LLMs are trained on general large-scale data sets, they only perform well in the tasks of the general field, but in some specific professional fields, the prediction accuracy of the model is low due to the lack of relevant training corpus.

Secondly, LLMs are prone to generating hallucinations. With the continuous development of artificial intelligence, various LLMs have emerged, demonstrating exceptional performance in tests and exhibiting near-human semantic understanding capabilities, but they cannot avoid output results that deviate from the facts. Since LLMs are trained on large-scale general-purpose data, they have broad prior knowledge base many domains, which sometimes leads to the generation of seemingly correct but unfounded information. In text classification tasks, unlike traditional deep learning models, which finally output definitive class labels through a softmax layer, LLMs may generate content outside the predefined label set, thereby interfering with the text classification task.

2.3. Few-shot learning method

Few-shot learning is a method that trains the machine learning model with limited supervised information [19]. Its main goal is to learn a classifier for new categories using only a very small number of training samples.

With the development of artificial intelligence technology, deep learning models have been widely applied to NLP tasks, but training deep learning models relies heavily on large-scale supervised datasets. In some specific fields, it is difficult to obtain abundant high-quality supervised data, which restricts the deep learning models. Therefore, few-shot learning has gradually attracted the attention of researchers. Wang et al. [20] proposed a data augmentation method based on prompt, which trains a small-scale soft prompt in the pre-training language model to ensure the quality of the generated data. The data augmentation method solves the problems of weak generalization ability and overfitting of the model caused by insufficient sample sizes in the field of NLP. Northcutt et al. [21] following a data-centric approach, proposed the confident learning framework to find out the error samples by evaluating the joint distribution of the true label and the pseudo-label.

3. Prompt-based data augmentation few shot text classification method

In order to solve the problems such as the difficulty of obtaining labeled data in specific fields, the weak generalization ability of few shot text classification model, and the noise of data after using sample augmentation technology, this paper constructs different prompts for specific text classification tasks, uses different prompts to fine-tune the LLM, and obtains the optimal fine-tuned LLM under the optimal prompt. And, this paper constructs a framework of few shot text classification assisted

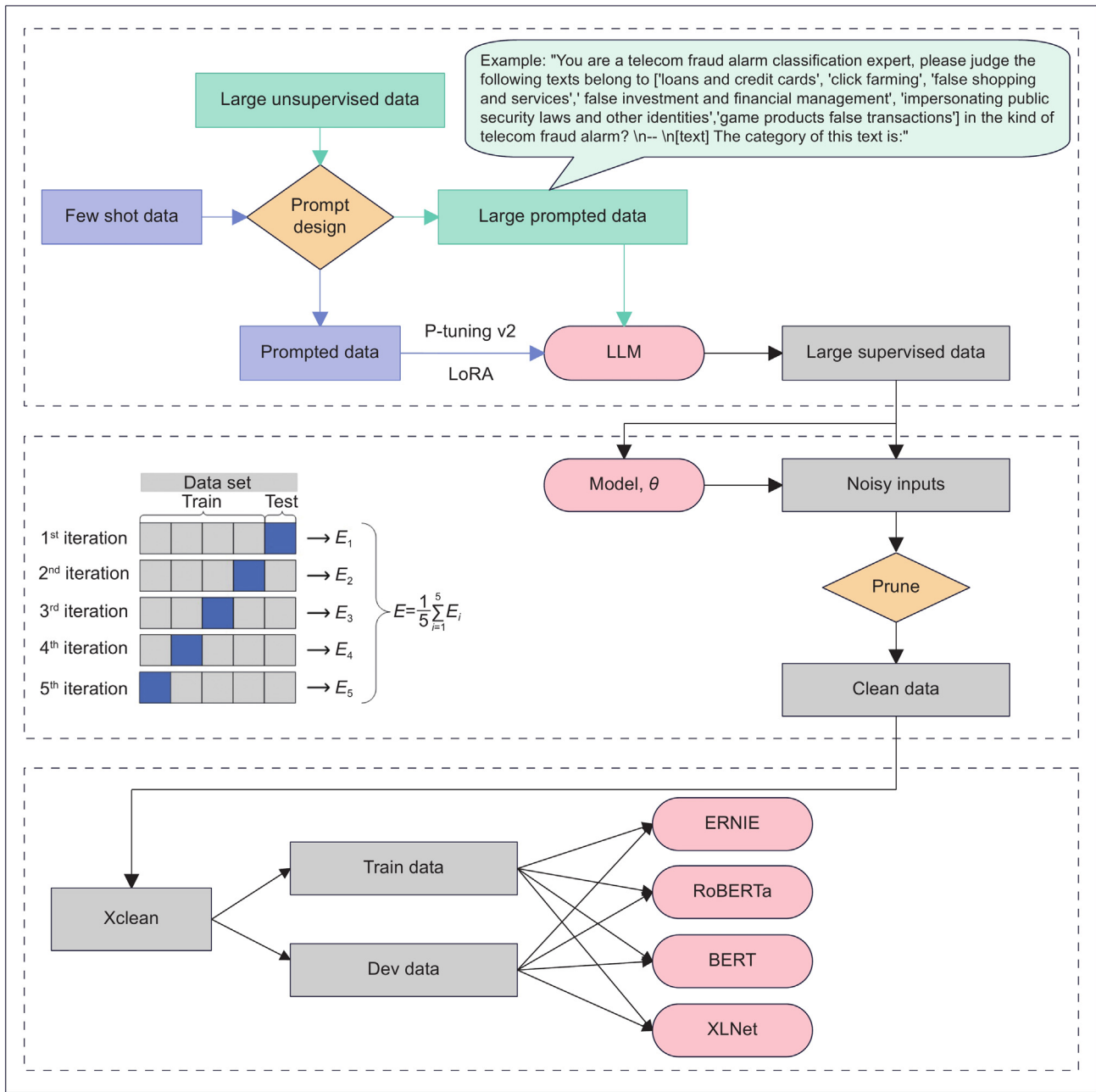


Fig. 1. Large Language Model Assisted Few Shot Text Classification Framework (FSTC-LLM). Since it is difficult to obtain labeled datasets in specialized fields, few shot labeled samples are used as inputs to fine-tune the LLM. Then, the best-performing fine-tuning model is used as a pseudo-label generator to label large-scale unlabeled data.

by large language model (FSTC-LLM), which combines the fine-tuning technology related to LLMs and uses the auxiliary means such as confidence learning to generate augmentation samples so as to achieve the task of few shot text classification. The framework is shown in Fig. 1.

As shown in Fig. 1, the FSTC-LLM framework proposed in this paper can be divided into three parts, namely, Augmented sample generation based on large language model, Confidence learning sample noise reduction and Deep learning training. Augmented sample generation based on large language model: Firstly, clean the labeled few shot data and unlabeled large-scale data. What is more, use different prompts to fine-tune the LLM on the few shot data, test the performance of different fine-tuning methods, and select the fine-tuning weight file of the LLM with the best performance. And finally, mount the trained model fine-tuning

weight file on the basic LLM and mark large-scale sample data to obtain a large-scale pseudo-label sample. Confidence learning noise reduction: The idea of confidence learning [21] is used to evaluate the joint distribution of real labels and pseudo-labels, and the Albert model is used to train the five-fold cross model to calculate the confidence of large-scale pseudo-label data, so as to screen the data scientifically. Deep learning training: Input the data with high confidence into the deep learning framework for training, and get the final text classification results, which verifies the advancement and effectiveness of the method proposed in this paper.

3.1. Augmented sample generation based on large language model

Fine-tuning techniques for LLMs can be divided into two types: full fine-tuning and parameter efficient fine-tuning. In

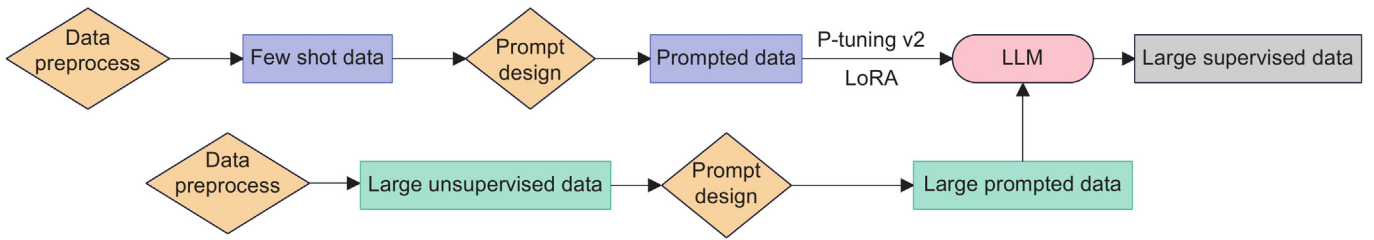


Fig. 2. Augmented Sample Generation Based on Large Language Model.

earlier studies, most researchers focused on full fine-tuning, but training costs increased markedly as model size and task complexity increased. Since then, parameter efficient fine-tuning has gradually become a hot topic in the research of LLMs. Fixing most of the parameters of the LLM and adjusting only a small number of parameters to train the LLM to adapt to downstream tasks not only greatly reduces the training cost, but also effectively improves the model performance. Next, we will detail two mainstream methods for parameter efficient fine-tuning.

3.1.1. P-Tuning v2

Prefix-Tuning [22] guides downstream task execution by freezing most of the model's parameters and training only a prefix portion. The fine-tuning method used in P-Tuning [23] maps discrete natural language into continuous word embeddings. However, it inserts these continuous prompts only into the first layer. Subsequent transformer layers rely entirely on the outputs of the previous layer for input. This design limits the effective propagation of continuous prompts to higher layers, constraining its performance on smaller-scale models.

P-Tuning v2 [24] adopts the idea of deep prompt tuning on the basis of the previous fine-tuning methods. It introduces prompts at every layer of the model and no longer relies on reparameterized encoders used in Prefix-Tuning and P-Tuning.

3.1.2. LoRA

Based on the idea of low-rank decomposition, Low-Rank Adaptation (LoRA) [25] decomposes a large parameter matrix into two smaller matrices during training to improve training efficiency. For pre-trained model parameters W_0 , the process of low-rank decomposition can be expressed as:

$$W_0 + \Delta W = W_0 + BA \quad (1)$$

Where W_0 is the initial parameter of the pretrained model, and ΔW is the parameter that need to be updated. Traditional full fine-tuning requires updating the entire parameter set W_0 . Using LoRA fine-tuning, only the matrices A and B, which compose ΔW , need to be updated.

3.1.3. Fine-tuning strategy

In this study, we select two general parameter efficient fine-tuning methods, P-Tuning v2 and LoRA. Compared with other parameter efficient fine-tuning methods, P-Tuning v2 has less interference to the model, keeps the original model weights unchanged during training, and performs well in the task of a small number of samples. LoRA has high parameter efficiency [26] and can not only observably reduce the number of training parameters but also provide finer-grained control over model weights. Based on these advantages, LoRA is more ideal for complex tasks. The Augmented sample generation based on large language model layer is shown in Fig. 2.

First, we preprocess the original text data and design the appropriate prompts according to the specific downstream task. These prompts aim to take full advantage of the semantic understanding capabilities of the LLM during fine-tuning, while

standardizing its outputs to reduce the probability of generating hallucinations.

Next, we feed the preprocessed data into the LLM, which is then trained using parameter efficient fine-tuning method to generate a model for the downstream tasks.

Finally, according to the experimental results, we select the fine-tuned model with the best performance as a large-scale pseudo-label generator. It annotates a large number of unsupervised samples to generate a high-quality pseudo-labeled dataset, further supporting the training of downstream tasks.

During the process of fine tuning the LLM, it may be difficult to control the fine tuning results because of the low interpretability of using parametric efficient fine tuning methods and the limited scale of fine tuning datasets. To address this issue, we combine fine-tuning techniques with prompt templates to ensure the quality of the generated text. For text classification tasks, we use the specially designed prompt templates, with their specific forms presented in Table 1 (Chinese prompt was used to train the model).

The Prompt 1 is a basic prompt to inform the LLM of the text classification task; the Prompt 2 is added with identity information to improve the understanding ability of the LLM; as the LLM is a generative language model, the next sentence output is added in the Prompt 3; the Prompt 4 allows the LLM to explain the reasons for classification to enhance interpretability; the Prompt 5 uses a thinking chain to allow the LLM to think step by step.

3.2. Confidence learning sample noise reduction

Due to the fine-tuned LLM is used to generate labels for large-scale unsupervised samples, there are a large number of noisy samples in the pseudo-label data set, and the effect of the model will be greatly limited if it is directly used as a training set to train the deep learning model. Therefore, with the help of the idea of confidence learning [21], this paper adds a confidence learning noise reduction layer to the few shot text classification framework assisted by a LLM to reduce the noise of large-scale pseudo-label samples and alleviate its impact. The confidence learning sample noise reduction framework is shown in Fig. 3.

As shown in Fig. 3, the confidence learning sample noise reduction framework is divided into two modules. Assuming that there is a noisy data set X , $[m]$ is the set of sample labels representing $\{1, 2, 3 \dots m\}$, for every sample $x \in X$, there is an unknown true label y^* and a possibly noisy original label \tilde{y} . The confidence learning screens out false samples through the joint distribution of y^* and \tilde{y} . The algorithm pseudo-code is as follows.

3.2.1. Estimating the joint distribution of noisy and true labels

First, a 5-fold cross-validation is performed on the pseudo-labeled dataset X . The N samples are evenly divided into 5 parts, each containing $\frac{N}{5}$. One of them is used as the test set, and the other four parts are used as the training set for fine-tuning the pre-trained Albert model. Then, we use the fine-tuned Albert

Table 1
Prompts.

Number	Prompt
Prompt 1	Please judge the following text belong to ['entertainment', 'game', 'science', 'society', 'finance', 'property', 'education', 'sports'] in the kind of news titles? \n - \n[text]
Prompt 2	You're an expert in classifying news titles, please judge the following text belong to ['entertainment', 'game', 'science', 'society', 'finance', 'property', 'education', 'sports'] in the kind of news titles? \n - \n[text]
Prompt 3	You're an expert in classifying news titles, please judge the following text belong to ['entertainment', 'game', 'science', 'society', 'finance', 'property', 'education', 'sports'] in the kind of news titles? \n - \n[text] \n - \n The category of this text is:
Prompt 4	You're an expert in classifying news titles, please judge the following text belong to ['entertainment', 'game', 'science', 'society', 'finance', 'property', 'education', 'sports'] in the kind of news titles? Please first explain the basis of the classification, and then give the result of the news title classification, only the result of the news title classification is output. \n - \n[text]
Prompt 5	You're an expert in classifying news titles, please judge the following text belong to ['entertainment', 'game', 'science', 'society', 'finance', 'property', 'education', 'sports'] in the kind of news titles? Please first summarize the content of the text, and then give the result of the news title classification, only the result of the news title classification is output. \n - \n[text]
Prompt 1	Please judge which of the following texts belong to ['loans and credit cards', 'click farming', 'false shopping and services', 'false investment and financial management', 'impersonating public security laws and other identities', 'game products false transactions'] in the kind of telecom fraud alarm? \n - \n[text]
Prompt 2	You are a telecom fraud alarm classification expert, please judge the following texts belong to ['loans and credit cards', 'click farming', 'false shopping and services', 'false investment and financial management', 'impersonating public security laws and other identities', 'game products false transactions'] in the kind of telecom fraud alarm? \n - \n[text]
Prompt 3	You are a telecom fraud alarm classification expert, please judge the following texts belong to ['loans and credit cards', 'click farming', 'false shopping and services', 'false investment and financial management', 'impersonating public security laws and other identities', 'game products false transactions'] in the kind of telecom fraud alarm? \n - \n[text] The category of this text is:
Prompt 4	You are a telecom fraud alarm classification expert, please judge the following texts belong to ['loans and credit cards', 'click farming', 'false shopping and services', 'false investment and financial management', 'impersonating public security laws and other identities', 'game products false transactions'] in the kind of telecom fraud alarm? Please first explain the basis of the classification, and then give the result of the telecom fraud alarm classification, only the result of telecom fraud alarm classification is output. \n - \n[text]
Prompt 5	You are a telecom fraud alarm classification expert, please judge the following texts belong to ['loans and credit cards', 'click farming', 'false shopping and services', 'false investment and financial management', 'impersonating public security laws and other identities', 'game products false transactions'] in the kind of telecom fraud alarm? Please first summarize the content of the text, and then give the result of the telecom fraud alarm classification, only the result of the news title classification is output. \n - \n[text]

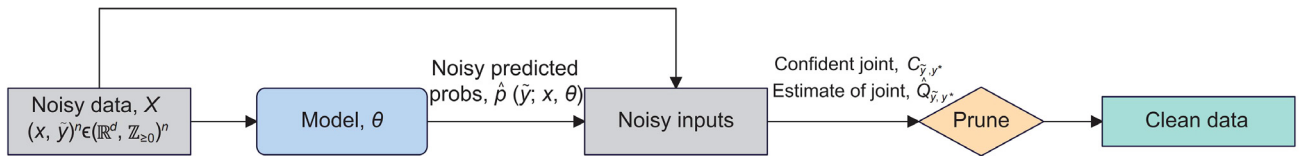


Fig. 3. Confidence Learning Sample Noise Reduction.

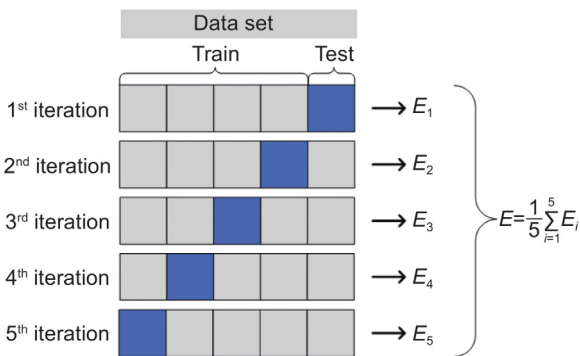


Fig. 4. Flow Chart of Five-fold Cross Validation.

model to predict the test set, generating the predicted probabilities $P[i][j]$ for all samples. The flow chart is shown in Fig. 4.

The samples are grouped according to their categories, and the set of samples with the possibly noisy original category j is denoted as $X_{\tilde{y}=j}$. For each category j , the confidence threshold t_j was calculated as the average probability of the samples in $X_{\tilde{y}=j}$

for category j . The calculation formula is as follows:

$$t_j = \frac{1}{|X_{\tilde{y}=j}|} \sum_{x \in X_{\tilde{y}=j}} \hat{p}(\tilde{y} = j; x, \theta) \quad (2)$$

Among, $|X_{\tilde{y}=j}|$ represents the number of samples in the set $X_{\tilde{y}=j}$, and $\hat{p}(\tilde{y} = j; x, \theta)$ is the predicted probability of the sample belonging to category j . For sample, we consider its true label y^* is determined as the category j corresponding to the maximum probability $P[i][j]$ and $P[i][j] > t[j]$. Then we compute the counting matrix $C_{\tilde{y}, y^*}[i][j]$. The calculation formula is as follows:

$$C_{\tilde{y}, y^*}[i][j] := |\hat{X}_{\tilde{y}=i, y^*=j}| \quad (3)$$

$$\hat{X}_{\tilde{y}=i, y^*=j} := \left\{ \begin{array}{l} x \in X_{\tilde{y}=i} : \hat{p}(\tilde{y} = j; x, \theta) \geq t_j, \\ j = \underset{l \in [m]: \hat{p}(\tilde{y}=l; x, \theta) \geq t_l}{\operatorname{argmax}} \hat{p}(\tilde{y} = l; x, \theta) \end{array} \right\} \quad (4)$$

Finally, we calibrate the counting matrix to make its sum of the counting matrix is the same as the total amount of data, and normalization is performed to obtain the joint distribution of the predicted labels and the given labels. The calculation formula is as follows:

$$\hat{Q}_{\tilde{y}, y^*=j} = \frac{C_{\tilde{y}=i, y^*=j}}{\sum_{j \in [m]} C_{\tilde{y}=i, y^*=j}} \cdot |X_{\tilde{y}=i}| \quad (5)$$

Table 2
Division of THUCNEWS dataset.

THUCNews dataset	train	augmentation	test
10shot	80	1600	16 000
20shot	160	1600	16 000

3.2.2. Cleaning data

First, for the non-diagonal elements of the counting matrix C , $n \cdot \hat{Q}_{\hat{y}=i, y^*=j}$ samples are selected to be sorted by the maximum margin for filtering to obtain the list of error label samples $L_{X_{error1}}$. Next, for each category, select $n \cdot \sum_{j \in [m]; j \neq i} (\hat{Q}_{\hat{y}=i, y^*=j}[i])$ samples to filter according to the lowest probability to obtain a list of error label samples $L_{X_{error2}}$. Finally, take the intersection of $L_{X_{error1}}$ and $L_{X_{error2}}$, then remove the samples with error labels in the original data set X . As a result, we can obtain a clean data set $L_{X_{clean}}$.

Algorithm 1 Confidence Learning Sample Noise Reduction

Input: Dataset X , Set of sample labels $[m]$

Output: Clean dataset X_{clean} and corresponding label set $[M_{clean}]$

- 1: Initialize the clean dataset X_{clean} as an empty set and the clean label set $[M_{clean}]$ as an empty set
- 2: Get the original label $y_{original}$ for each sample $x \in X$
- 3: The 5-fold cross-validation is performed on the dataset X to obtain the probability P for each sample under the class
- 4: For each possible label $y \in [m]$, the confidence threshold t and the count matrix C are calculated on the joint distribution Q
- 5: For the non-diagonal units of the counting matrix C , $n \cdot \hat{Q}_{\hat{y}=i, y^*=j}$ samples are selected for filtering, and sorted according to the maximum margin to obtain the list of error label samples $L_{X_{error1}}$. For each category, $n \cdot \sum_{j \in [m]; j \neq i} (\hat{Q}_{\hat{y}=i, y^*=j}[i])$ samples are selected for filtering and sorting according to the lowest probability to obtain the list of error label samples $L_{X_{error2}}$, and the intersection $L_{X_{error}}$ of $L_{X_{error1}}$ and $L_{X_{error2}}$ is taken
- 6: **for** each sample $x \in X$ **do**
- 7: **if** $x \notin L_{X_{error}}$ **then**
- 8: Add sample x to X_{clean}
- 9: Add label $y_{original}$ to $[M_{clean}]$
- 10: **end if**
- 11: **end for**
- 12: **return** the clean dataset X_{clean} and the corresponding label set $[M_{clean}]$

4. Experimental verification and analysis

In order to verify the effectiveness of the proposed FSTC-LLM framework in text classification tasks, experiments are carried out on the public data set and the police data set respectively. Firstly, prompts are designed for specific text classification tasks, and fine-tuning experiments are carried out on ChatGLM3-6B using different prompts and different fine-tuning methods. We select the prompt with the optimal performance and the LLM to perform large-scale sample data marking to obtain an enhanced sample data set. Then the idea of confidence learning is used to denoise large-scale labeled samples, and the denoising effect is compared. Finally, the denoised data is sent to the deep learning model for training.

4.1. Experimental dataset

We select two datasets for experimentation: the publicly available Chinese news text classification dataset THUCNews, and

Table 3
Division of Alarm dataset.

Alarm dataset	train	augmentation	test
20shot	120	1200	120
30shot	180	1200	120

Table 4
The symbol definition and calculation method.

Symbol	Name	Meaning
TP	True positive	The truth is a positive sample and the prediction is a positive sample
FP	False positive	The truth is a negative sample and the prediction is a positive sample
TN	True negative	The truth is a negative sample and the prediction is a negative sample
FN	False negative	The truth is a positive sample and the prediction is a negative sample

a specialized police alarm dataset. Next, we elaborate on the characteristics of the datasets.

The THUCNews dataset is a widely used Chinese news text classification dataset containing about 740,000 news articles across 14 categories. Each news includes both a title and body content. We select title data from eight categories: Entertainment, Game, Science, Society, Finance, Property, Education, and Sports as the experimental subjects. The average length of the resulting subset samples is about 20 words.

To evaluate the model's performance in few-shot learning scenarios, experiments were conducted using 10shot and 20shot strategies. Small samples were randomly selected from each category to fine-tune the LLM. The fine-tuned model was then used to generate labels for 1600 data samples. To enhance the quality of generated labels, confidence learning sample noise reduction layer was introduced to denoise the pseudo-labels. The clean data was subsequently split into training and validation sets in an 8:2 ratio for training the deep learning model. Additionally, 2000 samples from each category were selected, denoised using the confidence learning approach, and used as a test set to evaluate model performance. Details of the data partitioning are presented in Table 2.

The police alarm dataset consists of 6013 telecommunication fraud alarms classified into six categories: Loans and Credit Cards, Click Farming, False Shopping and Services, False Investment and Financial Management, Impersonation Public Security Laws and Other Identities, and Game Products False Transactions.

In the experiments, 20shot and 30shot sample datasets were selected from each category to fine-tune the LLM. The fine-tuned model was then used to generate labels for 1200 samples, which were subsequently denoised using a confidence learning sample noise reduction layer. The denoised data was divided into training and validation sets in an 8:2 ratio to train the deep learning model. Additionally, 120 samples were selected, denoised using the confidence learning approach, and used as a test set to evaluate model performance. Details of the data partitioning are presented in Table 3.

The THUCNews dataset includes 8 representative categories, covering diverse categories such as Entertainment, Science, and Finance, with the diversity of general fields. The police alarm dataset focuses on six high-frequency categories within the domain of telecommunication fraud, being more in line with the typical characteristics of the professional field. This selection fully demonstrates the text classification ability of the model in different fields. To ensure the robustness of model evaluation, the number of text samples in each category was kept balanced. The training, validation, and test sets undergo confidence learning sample noise reduction layer and are evenly distributed according to predefined ratios.

4.2. Experimental evaluation index

In this paper, the accuracy (ACC) and F1-measure in the traditional text classification task are used to evaluate the performance of the FSTC-LLM framework proposed in this paper. The symbol definition and calculation method are shown in Table 4.

4.2.1. Accuracy

Accuracy represents the proportion of correctly classified samples to the total number of samples, and its calculation formula is as follows:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

The higher the accuracy, the better the performance of the text classification model.

4.2.2. F1 measure

We also incorporate the F1 measure for assessing the model's performance, and its calculation formula is as follows:

$$P = \frac{TP}{TP + FP} \quad (7)$$

$$R = \frac{TP}{TP + FN} \quad (8)$$

$$F1 = \frac{2PR}{P + R} \quad (9)$$

The Precision (P) is used to represent the proportion of the number of true positive samples among all the predicted positive samples. The Recall (R) is used to represent the proportion of the actual number of true positive samples predicted out of all the actual number of positive samples. In general, the Precision and Recall cannot be increased or decreased at the same time, and a metric is needed to find a balance between them. Therefore, we consider both Precision and Recall of the model, and introduce F1 measure to comprehensively evaluate the performance of the model. The higher the F1 measure, the better the performance of the model.

4.3. Experimental parameters and environment

The experimental environment for this paper is Ubuntu 22.04 operating system, GPU processor is NVIDIA A100 graphics card, with 40 GB of memory, programming language is Python 3.10.13, and CUDA version is 12.2. The LLMs deep learning framework for the model uses Pytorch 2.1.2 and Transformers 4.36.2. In the fine-tuning experiment of the LLMs, the average time for fine-tuning the police alarm data set is 345.4 s, and the average time for fine-tuning the THUCNews data set is 161 s. In addition, to evaluate the scalability of the method, we also conducted experiments on the NVIDIA RTX 3090 GPU, verifying that it can complete the task of generating pseudo-labels and subsequent tasks. In practical applications, the alarm text classification model can be run on an Intel Core i9-13900H CPU, with an average inference latency of approximately 15 items per second.

4.4. Experimental results and analysis

4.4.1. Comparison of the enhancement effects of LLMs

In this paper, ChatGLM3-6B, the third generation LLM jointly released by Tsinghua University and Zhipu AI, is selected as the basic LLM. Compared with the second generation model, ChatGLM3-6B uses more diverse training data, more sufficient training steps and more reasonable training strategies, and its performance in many NLP tasks is better than that of the basic model below 10B in the same period. Compared with the

Table 5

Fine-tuning results for LLMs on the THUCNEWS dataset (ACC%).

Prompt	Pt v2(10s)	LoRA(10s)	Pt v2(20s)	LoRA(20s)
Prompt1	88.06	88.56	89.75	91.31
Prompt2	89.38	88.88	90.31	91.25
Prompt3	88.13	88.00	90.25	91.13
Prompt4	88.50	89.25	90.25	89.75
Prompt5	88.00	89.50	90.13	90.56

large model ChatGPT with 100 billion parameters of decoder architecture only, ChatGLM3-6B, which uses the idea of encoder-decoder architecture, has a significant advantage of lightweight operation with 600 million model parameters, and because of the confidentiality of data in police robots, the use of domestic open-source LLM is more conducive to data confidentiality.

In order to study the influence of fine-tuning on the LLM, based on ChatGLM3-6B, this paper uses THUCNews data of 10shot and 20shot samples and police alarm data of 20shot and 30shot samples to fine-tune the model by P-tuning v2 and LoRA, and evaluates the performance using ACC and F1 measure. The experimental results of THUCNews data are shown in Table 5.

On the THUCNews dataset, the fine-tuning effect of LoRA is better than that of P-tuning v2, the effect of Prompt 5 is the best under the 10shot sample size, the accuracy rate is 89.5%, and the effect of Prompt 1 is the best under the 20shot sample size, the accuracy rate is 91.31%. The quality of the fine tuning using the prompt is higher and the model output hallucination is reduced.

The experimental results of alarm data are shown in Table 6. On the alarm dataset, the fine-tuning performance of LoRA is also better than P-tuning v2. We found that using the Prompt 3 worked best on both of the two small sample sets. The accuracy rates were 85.42% with the 20shot and 86.5% with the 30shot. Therefore, we used the Prompt 3 to carry out subsequent pseudo-label generation for the alarm data set.

Although the prediction accuracy of LLMs can be significantly improved by combining parameter efficient fine-tuning with prompt-based template fine-tuning, the pseudo-labeled data generated through this process still contains incorrect labels. We define these erroneously generated samples as noise.

4.4.2. Comparison of the effects of confidence learning sample noise reduction

In order to further analyze the capability of the text classification framework assisted by large language model, the influence of the confidence learning sample noise reduction on the framework is discussed next. The confidence learning noise reduction is used to reduce the noise of the sample data that generates the label. The experimental results are shown in Table 7.

It can be seen from Table 7 that the confidence learning noise reduction can effectively clean a large number of noise samples, and the cleaned samples account for 4.3% to 5.4% of the total samples, so as to improve the accuracy of deep learning training data.

4.4.3. Comparative test

The data set denoised by confidence learning is defined as Xclean. For each set of data, this experiment uses different deep learning models to experiment on the enhanced THUC and alarm samples. The experimental results are shown in Tables 8 and 9.

It can be seen from Table 8 that, compared with the fine-tuned LLM, the general text classification ability of the few-shot text classification framework assisted by large language model is stronger, the accuracy of the best performance under the sample size of 10shot is 91.44%, and the accuracy of the best performance under the sample size of 20shot is 92.32%, which were 2.17% and 1.11% higher than that of the basic LLM, respectively.

Table 6
Fine-tuning results for LLMs on the alarm dataset (ACC%).

Prompt	Pt v2(20s)	LoRA(20s)	Pt v2(30s)	LoRA(30s)
Prompt1	82.83	83.92	84.33	84.08
Prompt2	83.58	84.33	85.08	84.92
Prompt3	85.17	85.42	86.25	86.50
Prompt4	83.17	84.25	84.00	85.08
Prompt5	82.83	84.50	83.33	85.00

Table 7
Noise reduction of augmentation samples.

	THUC	Alarm
Total number of enhanced samples	1600	1200
10/20 shot Enhanced Sample Cleanup	72	65
20/30 shot Enhanced Sample Cleanup	85	52

Table 8
Experimental results of few shot text classification framework assisted by large language model (THUCNEWS).

THUC (Xclean)	ACC% (10shot)	ACC% (20shot)	F1 (10shot)	F1 (20shot)
GLM Fine Tuning	89.50	91.31	0.8973	0.9144
FSTC-LLM-ERNIE	91.13	92.32	0.9112	0.9236
FSTC-LLM-RoBERTa	91.31	92.03	0.9125	0.9208
FSTC-LLM-BERT	91.44	91.66	0.9140	0.9170
FSTC-LLM-XLNet	91.04	91.64	0.9096	0.9163

Table 9
Experimental results of few shot text classification framework assisted by large language model (Alarm).

Alarm (Xclean)	ACC% (20shot)	ACC% (30shot)	F1 (20shot)	F1 (30shot)
GLM Fine Tuning	85.42	86.50	0.8556	0.8618
FSTC-LLM-ERNIE	87.50	89.06	0.8752	0.8839
FSTC-LLM-RoBERTa	88.14	88.98	0.8797	0.8870
FSTC-LLM-BERT	85.94	87.50	0.8607	0.8731
FSTC-LLM-XLNet	85.59	87.29	0.8556	0.8715

Table 10
Experimental results of deep learning models.

Model	ACC% (THUC)	F1 (THUC)	ACC% (Alarm)	F1 (Alarm)
FSTC-LLM	92.32	0.9236	89.06	0.8839
ERNIE	77.59	0.7600	85.59	0.5800
RoBERTa	89.84	0.8982	83.90	0.8319
BERT	88.06	0.5000	83.90	0.6500
BERT-DPCNN	87.38	0.8746	83.90	0.8367
TextCNN-Att	74.42	0.7430	81.36	0.8105
XLNet	84.80	0.8447	41.53	0.3543

Table 11
Experimental results of undenoised datasets and denoise datasets (THUCNews).

Model	Type	ACC% (10s)	ACC% (20s)	F1 (10s)	F1 (20s)
FSTC-LLM-ERNIE	Xclean	91.13	92.32	0.9112	0.9236
FSTC-LLM-ERNIE	Xdirty	90.69	67.42	0.9066	0.6440
FSTC-LLM-RoBERTa	Xclean	91.31	92.03	0.9125	0.9208
FSTC-LLM-RoBERTa	Xdirty	90.55	91.04	0.9051	0.9112
FSTC-LLM-BERT	Xclean	91.44	91.66	0.9140	0.9170
FSTC-LLM-BERT	Xdirty	90.81	91.13	0.9084	0.9116
FSTC-LLM-XLNet	Xclean	91.04	91.64	0.9096	0.9163
FSTC-LLM-XLNet	Xdirty	90.51	91.16	0.9046	0.9114

It can be seen from Table 9 that, compared with the fine-tuned LLM, the alarm text classification ability of the few-shot text classification framework assisted by large language model is stronger, the accuracy of the best performance under the 20shot sample size is 88.14%, and the accuracy of the best performance

Table 12
Experimental results of undenoised datasets and denoise datasets (Alarm).

Model	Type	ACC% (20s)	ACC% (30s)	F1 (20s)	F1 (30s)
FSTC-LLM-ERNIE	Xclean	87.50	89.06	0.8752	0.8839
FSTC-LLM-ERNIE	Xdirty	84.38	81.25	0.8462	0.8078
FSTC-LLM-RoBERTa	Xclean	88.14	88.98	0.8797	0.8870
FSTC-LLM-RoBERTa	Xdirty	83.90	81.36	0.8363	0.8099
FSTC-LLM-BERT	Xclean	85.94	87.50	0.8607	0.8731
FSTC-LLM-BERT	Xdirty	82.81	82.81	0.8309	0.8210
FSTC-LLM-XLNet	Xclean	85.59	87.29	0.8556	0.8715
FSTC-LLM-XLNet	Xdirty	82.20	83.90	0.8204	0.8330

under the 30shot sample size is 89.06%, which were 3.18% and 2.96% higher than that of the basic LLM, respectively.

In order to further verify the classification ability of the framework, this paper uses 30shot original samples to train the deep learning models, and the experimental results are shown in Table 10.

As shown in Table 10, the best accuracy of the deep learning model trained with 30shot is 89.84% and 85.59% respectively, but the classification effect is slightly inferior to that of the FSTC-LLM framework. The effectiveness of the proposed framework is verified.

4.4.4. Ablation experiment

We cancel the confidence learning sample noise reduction layer to explore the effect of it, and define the dataset with noise as Xdirty. We use each set of data to train different deep learning models. The experimental results are shown in Tables 11 and 12.

It can be seen from the Tables 11 and 12 that the training effect of Xclean is better than that of Xdirty under the four deep learning models, which verifies the role of the confidence learning sample noise reduction layer in this framework.

5. Conclusion

In this paper, the police robot is studied, and a novel method (FSTC-LLM) is proposed for sample augmentation based on LLM and noise reduction. The LLM is fine-tuned by few shot samples, and the fine-tuned LLM is used to enhance the samples, and then the enhanced samples are cleaned with confidence model, and finally sent to the deep learning model for training. In practical applications, the FSTC-LLM-ERNIE model with the highest accuracy rate is selected to be deployed in the police robot. The experimental results show that FSTC-LLM performs well in few shot training and has a good ability of police alarm classification, which can provide technical support for the police robot to complete the task of automatic analysis of alarm with high quality. In the future, we will extend our proposed method to more tasks, such as assisting police robots in voiceprint and image recognition.

CRedit authorship contribution statement

Zirui Liu: Writing – original draft, Validation, Supervision, Methodology, Investigation, Conceptualization. **Haichun Sun:** Writing – review & editing, Visualization, Funding acquisition. **Deyu Yuan:** Formal analysis, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded by the Basic research expenses of People's Public Security University of China (2024JKF02), and the Key Project of the Ministry of Public Security Technology Research Program (2024JSZ01).

References

- [1] D.S. Navare, Y.R. Kapde, S. Maurya, D. Pardeshi, P. William, Robotic bomb detection and disposal: Application using arduino, in: 2022 7th International Conference on Communication and Electronics Systems, ICCES, 2022, pp. 479–483, <http://dx.doi.org/10.1109/ICCES54183.2022.9836011>.
- [2] J. Zhou, Y. Zheng, J. Tang, L. Jian, Z. Yang, FlipDA: Effective and robust data augmentation for few-shot learning, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 8646–8665, <http://dx.doi.org/10.18653/v1/2022.acl-long.592>.
- [3] S. Lei, X. Zhang, J. He, F. Chen, C.-T. Lu, TART: Improved few-shot text classification using task-adaptive reference transformation, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 11014–11026, <http://dx.doi.org/10.18653/v1/2023.acl-long.617>.
- [4] K. Zhao, X. Jin, Y. Wang, Survey on few-shot learning, *J. Softw.* 32 (2) (2021) 349–369, <http://dx.doi.org/10.13328/j.cnki.jos.006138>.
- [5] Z. Mao, R. Kobayashi, H. Nabae, K. Suzumori, Multimodal strain sensing system for shape recognition of tensegrity structures by combining traditional regression and deep learning approaches, *IEEE Robot. Autom. Lett.* 9 (11) (2024) 10050–10056, <http://dx.doi.org/10.1109/LRA.2024.3469811>.
- [6] H. Ma, A. Song, J. Li, L. Ge, C. Fu, G. Zhang, Legged odometry based on fusion of leg kinematics and IMU information in a humanoid robot, *Biomim. Intell. Robot.* 5 (1) (2025) 100196, <http://dx.doi.org/10.1016/j.birob.2024.100196>.
- [7] S. Sun, C. Li, Z. Zhao, H. Huang, W. Xu, Leveraging large language models for comprehensive locomotion control in humanoid robots design, *Biomim. Intell. Robot.* 4 (4) (2024) 100187, <http://dx.doi.org/10.1016/j.birob.2024.100187>.
- [8] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE* 86 (11) (1998) 2278–2324, <http://dx.doi.org/10.1109/5.726791>.
- [9] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (8) (1997) 1735–1780, <http://dx.doi.org/10.1162/neco.1997.9.8.1735>.
- [10] X. Guo, H. Zhang, L. Ye, S. Li, RnnTd: An approach based on LSTM and tensor decomposition for classification of crimes in legal cases, in: 2019 IEEE Fourth International Conference on Data Science in Cyberspace, DSC, 2019, pp. 16–22, <http://dx.doi.org/10.1109/DSC.2019.00012>.
- [11] W. Wang, D. Feng, B. Li, J. Tian, Atextcnn model: a new multi-classification method for police situation, in: International Conference on Advanced Data Mining and Applications, 2020, pp. 135–147, http://dx.doi.org/10.1007/978-3-030-65390-3_11.
- [12] J. Zhou, H. Xu, Z. Zhang, J. Lu, W. Guo, Z. Li, Using recurrent neural network structure and multi-head attention with convolution for fraudulent phone text recognition, *Comput. Syst. Sci. Eng.* 46 (2) (2023) 2277–2297, <http://dx.doi.org/10.32604/csse.2023.036419>.
- [13] S. Yuan, Q. Wang, Imbalanced traffic accident text classification based on Bert-RCNN, *J. Phys.: Conf. Ser.* 2170 (1) (2022) 012003, <http://dx.doi.org/10.1088/1742-6596/2170/1/012003>.
- [14] C. Zhang, J. Chen, J. Li, Y. Peng, Z. Mao, Large language models for human-robot interaction: A review, *Biomim. Intell. Robot.* 3 (4) (2023) 100131, <http://dx.doi.org/10.1016/j.birob.2023.100131>.
- [15] J. Chung, E. Kamar, S. Amershi, Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 575–593, <http://dx.doi.org/10.18653/v1/2023.acl-long.34>.
- [16] B. Ding, C. Qin, L. Liu, Y.K. Chia, B. Li, S. Joty, L. Bing, Is GPT-3 a good data annotator? in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 11173–11195, <http://dx.doi.org/10.18653/v1/2023.acl-long.626>.
- [17] Z. Chen, Q. Gao, A. Bosselut, A. Sabharwal, K. Richardson, DISCO: Distilling counterfactuals with large language models, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 5514–5528, <http://dx.doi.org/10.18653/v1/2023.acl-long.302>.
- [18] X. Xing, P. Chen, Entity extraction of key elements in 110 police reports based on large language models, *Appl. Sci.* 14 (17) (2024) 7819, <http://dx.doi.org/10.3390/app14177819>.
- [19] B. Yu, C. Xingye, W. Jingxuan, Few-shot text classification method based on prompt learning, *J. Comput. Appl.* 43 (09) (2023) 2735–2740, <http://dx.doi.org/10.11772/j.issn.1001-9081.2022081295>.
- [20] Y. Wang, C. Xu, Q. Sun, H. Hu, C. Tao, X. Geng, D. Jiang, PromDA: Prompt-based data augmentation for low-resource NLU tasks, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 4242–4255, <http://dx.doi.org/10.18653/v1/2022.acl-long.292>.
- [21] C. Northcutt, L. Jiang, I. Chuang, Confident learning: Estimating uncertainty in dataset labels, *J. Artificial Intelligence Res.* 70 (2021) 1373–1411, <http://dx.doi.org/10.1613/jair.1.12125>.
- [22] X.L. Li, P. Liang, Prefix-tuning: Optimizing continuous prompts for generation, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 4582–4597, <http://dx.doi.org/10.18653/v1/2021.acl-long.353>.
- [23] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, J. Tang, GPT understands, too, *AI Open* 5 (2024) 208–215, <http://dx.doi.org/10.1016/j.aiopen.2023.08.012>.
- [24] X. Liu, K. Ji, Y. Fu, W.L. Tam, Z. Du, Z. Yang, J. Tang, P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021, pp. 61–68, <http://dx.doi.org/10.18653/v1/2022.acl-short.8>.
- [25] E.J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25–29, 2022, 2022, <http://dx.doi.org/10.48550/arXiv.2106.09685>.
- [26] V.B. Parthasarathy, A. Zafar, A. Khan, A. Shahid, The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities, 2024, <http://dx.doi.org/10.48550/arXiv.2408.13296>, arXiv preprint [arXiv:2408.13296](https://arxiv.org/abs/2408.13296).