Accurate Classification Framework for Government Data: A Prompt-tuning and LangChain Approach

Anonymous ACL submission

Abstract

Government Data (GD) is pivotal in driving social and economic development. Currently, existing GD classification schemes require adherence to specific classification standards and formats. However, precise classification of GD is challenging due to the lack of high-quality classification samples and the resource-intensive nature of the manual classification process, resulting in poor classification performance. To address this challenge, this paper integrates prompt-tuning with LangChain to propose a LangChain-based Classification Framework for Government Data (GD-LCF) to enable automated classification without requiring extensive training samples. In our scheme, we first leveraged web crawling techniques to establish a comprehensive dataset of Chinese government data. Then, we introduce a Classification Prompting Pattern (CPP) and combine with the LangChain technology to facilitate zero-shot prompt-tuning, enabling us to achieve remarkable classification precision with minimal training data. Finally, to further mitigate the reliance on manual efforts, we developed a Clustering Mapping (CLM) strategy to transform the predefined categories and the LangChain output into clustered embeddings, which are subsequently classified based on their proximity to the category centers. Experiments show that GD-LCF achieves a classification precision of 74.33% when training data is few or even zero, which is the highest precision compared to existing classification schemes.

1 Introduction

Government Data (GD), also known as "public sector information", encompasses various categories such as demographics, economic indicators, and health metrics. They play a crucial role in enhancing government decision-making and public service quality and fostering transparency and accountability in governmental activities. During the COVID-19 outbreak, real-time health data facilitated global governments in monitoring virus spread, efficiently distributing healthcare resources, and disseminating vital information to the public [1]. Effective categorization of GD is essential for data processing, and organizing data by category enhances the utilization efficiency of GD. However, one of the challenges faced by governments is the lack of labeled reference samples during data classification. Manual labeling of samples is both time-consuming and labor-intensive. Hence, there is a pressing need for an automated approach to streamline the process and enhance efficacy.

In academia and industry, current methodologies for efficient GD categorization typically involve rule-based methods and deep-learning methods. Rule-based methods utilize predefined data recognition and classification rules, such as keyword matching (Yamada and Nakatoh, 2018), pattern recognition (Feng et al., 2024), or identifying structural attributes (Crusoe and Clarinval, 2023). For instance, Song et al. (2019) introduced a technique using IF-IDF for information retrieval to match GD datasets with specific retrieval rules for classification purposes. Nonetheless, these approaches often demand domain experts to formulate rules, posing challenges in adapting to evolving data diversity and complexity. Given the remarkable capabilities of deep learning in handling vast datasets, especially the exceptional performance of pre-trained large models in downstream tasks, researchers are exploring automatic classification methods employing deep learning models (Pan and Chen, 2021; Huang et al., 2021; Liu, 2022). For instance, Krzysztof and Pajak (2022) demonstrated the viability of fine-tuning a multilingual classification model as a language corrector to enhance classification precision.

Despite the efficacy of the above techniques,

[1] https://www.sohu.com/a/384365092_650579

certain challenges persist. 1) **Limited training samples:** Current pre-training-based classification methods (e.g., Transformer, BERT, or GPT) necessitate a substantial amount of labeled data for training. However, the scarcity of high-quality labeled datasets in the GD domain restricts the performance of existing schemes. 2) **Resource-intensive challenge:** Current neural network-based feature extraction techniques predominantly rely on fine-tuning, resulting in time and resource consumption. 3) **Domain expertise requirement:** Existing pre-training classification methods often depend on answer engineering or specialized knowledge to pre-define the answer space for classification outcomes, which is inefficient.

To address the above challenges, we propose a LangChain based Classification Framework for Government Data (GD-LCF) to classify automatically. To the best of our knowledge, this is the first work using prompt-tuning with LangChain to address the GD Classification tasks. The main contributions of our approach are listed as follows.

1) To address the zero-shot classification problem, we developed the Classification Prompt Pattern (CPP). By leveraging prompt engineering, we transformed GD classification problem into a mask prediction task and utilize Large Language Models (LLMs) to predict masks, thus achieving effective classification results.

2) To reduce resource consumption, we employed the LangChain approach, which involves remotely invoking models such as GPT-3.5 and GPT-4 to accomplish the classification task without adjusting model parameters.

3) To further reduce manual consumption, we developed Clustering Mapping (CLM), which classifies by judging the distance between the clustered embeddings and the predefined category centers. The CLM has few restrictions on the range of potential answers, obviating the necessity for answer engineering or specialized knowledge.

2 Related Work

2.1 GD Classification

Existing classification schemes in GD can be roughly divided into two categories: rule-based approaches and deep learning-based approaches. In the first scheme, Rule-based methods typically rely on predefined rules such as keyword matching, pattern recognition, and structural features for data recognition and classification. Along with the line, Yamada and Nakatoh (2018) utilized a multi-label classification technique to compare phrase and word frequency, assigning labels to GD based on training data. Feng et al. (2024) introduced an evolutionary game theory-based approach with long-term decision-making for enhanced GD disclosure and utilization. Crusoe and Clarinval (2023) proposed a taxonomy with 24 patterns for GD classification and used clustering to identify the optimal pattern.

Different from the rule-based approaches, deep learning-based approaches utilize neural networks to extract GD features for classification. Along with the line, Pan and Chen (2021) employed Bi-LSTM with attention mechanism to develop a superior GD classification algorithm compared to LSTM, logistic regression, and plain Bayes. Huang et al. (2021) formulated a multi-label cross-entropy loss function for a multi-classification system, indicating TextCNN and TextRCNN's stability on this loss function. Liu (2022) introduced a multi-label classification framework incorporating a Graph Convolutional Network with BERT and Memory Network, outperforming several other models for multi-label GD classification.

Although the above schemes are very effective in classification, rule-based methods necessitate domain experts for rule development, making them time-consuming and less adaptable to increasing data. In contrast, deep learning-based techniques need fewer experts for network design but demand substantial labeled data for training, presenting challenges in the GD domain.

2.2 LangChain

LangChain is a framework for building applications with LLMs, which provides a set of abstractions and tools for building applications with LLMs in a modular and extensible way. Existing research on LangChain mainly focuses on utilizing the ChatGPT interface to solve practical issues.

Singh et al. (2024) presented MindGuide, an innovative chatbot leveraging LangChain and ChatOpenAI to provide a comprehensive mental health assistant addressing the growing challenges of anxiety, depression, and suicidal thoughts in modern society. Soygazi and Oguz (2023) explored the potential of LLMs, particularly ChatGPT and LLMMathChain, in solving mathematical word problems, highlighting the need for further advancements to improve their performance in this domain. Nascimento et al. (2023) introduced a fam-

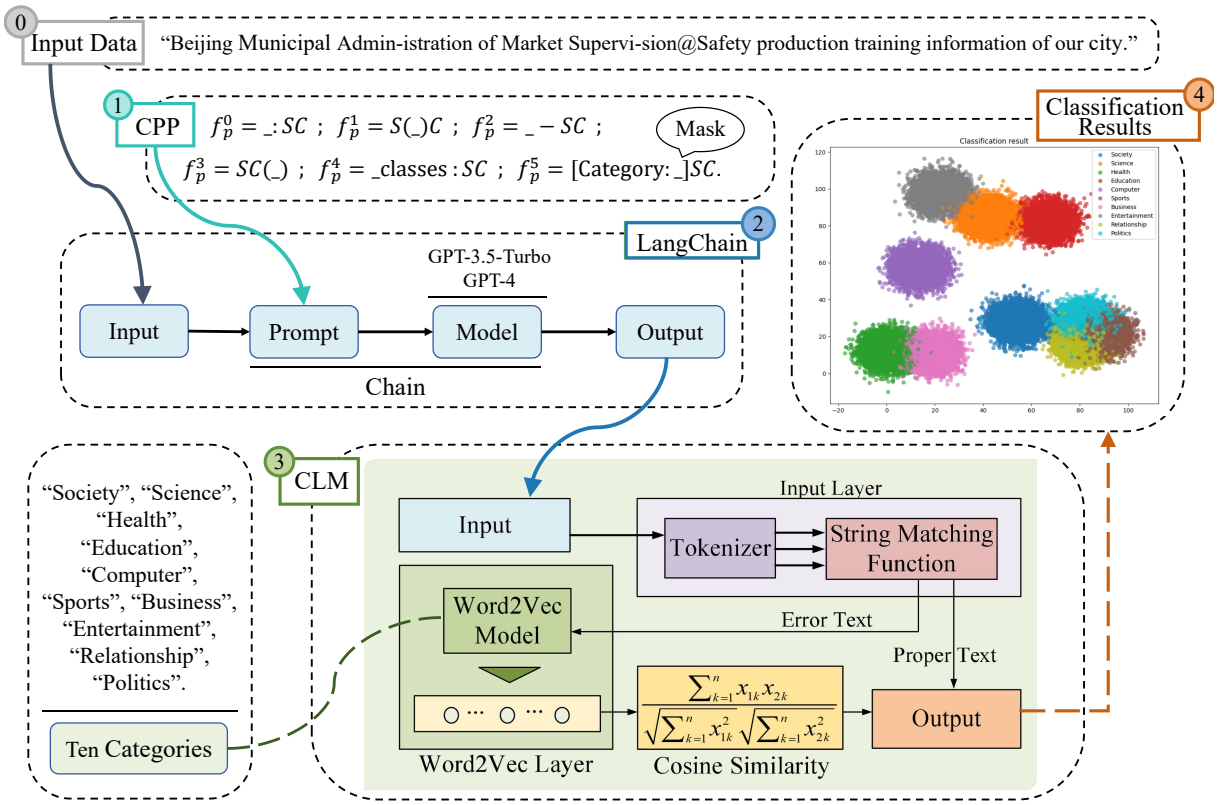

Figure 1: The LangChain based Classification Framework for Governmental Data (GD-LCF).

ily of Natural Language Interfaces for Databases (NLIDBs) that leverage ChatGPT and LangChain to translate NLI questions into SQL queries or extract keywords for database searches, facilitating a few-shot learning process to create NLIDBs.

The above methods employing LangChain solely rely on conversing with the model to derive results, lacking further processing of the outputs generated by ChatGPT. Additionally, the conversational process necessitates providing substantial prior knowledge, which does not contribute to reducing the manual effort involved in classifying GD.

3 The Approach

3.1 Overview

As shown in Figure 1, we integrate prompt-tuning with LangChain to propose a classification framework (GD-LCF) for GD. Distinct from conventional pre-trained classification schemes used in the GD field, such as BERT and Transformer, GD-LCF introduces CPP, which employs prompt-tuning to convert the classification task into a mask prediction using a LLMs, achieving better classification results. Meanwhile, the LangChain method is utilized to address the classification task, requiring only the design of prompt content. This optimized approach improves classification precision while significantly reducing labor costs and directly lowering the number of necessary training samples. Moreover, GD-LCF incorporates CLM, which does not require answer engineering or specialized knowledge, further enhancing classification efficiency and reducing resource consumption.

The proposed GD-LCF is composed of sample processing, CPP, LangChain and CLM, and the pseudo-code for our approach is given in Algorithm 1. Among them, CPP, LangChain and CLM are the key technologies, which will be described in detail in the following chapters. The specific process is as follows: 0) Initially, we employ crawling techniques to obtain the input data from the open Chinese government website. 1) The CPP is designed based on the data features for prompt-tuning. 2) The prompt content, composed of several elements, is designed as a fixed template. Then, the input text and CPP are embedded within the template. 3) The template and the models provided by the LangChain interface constitute a chain, and invoking the chain can output the classification results. 4) Finally, the classification results and the predefined categories are transformed into vectors using a Word2Vec model, and the final classifica-

Algorithm 1: GD-LCF

Data: D_{raw}: Crawled Data, C_{predef}: Predefined Categories, P: Prompt Templates, M: LangChain Model
Result: $L_{\text{classified}}$: Final Classification Labels
Initialization: $D_{\text{crawl}}, D_{\text{clean}}, D_{\text{balance}}$
$D_{\text{crawl}} \leftarrow \text{Crawl}(D_{\text{raw}})$ $D_{\text{clean}} \leftarrow \text{Clean}(D_{\text{crawl}})$
$D_{\text{balance}} \leftarrow \text{Balance}(D_{\text{clean}})$
for $i = 1$ ***to*** $|D_{crawl}|$ **do**
  $s_{\text{clean}} \leftarrow \text{Clean}(D_{\text{crawl}}[i])$
  $s_{\text{std}} \leftarrow \text{Standardize}(s_{\text{clean}})$
  $s_{\text{balance}} \leftarrow \text{Balance}(s_{\text{std}})$
end
for $node \in PST.nodes$ **do**
  $token \leftarrow \text{GenerateUniqueToken}(node)$
  $tokens.append(token)$
end
for $j = 1$ ***to*** $|D_{balance}|$ **do**
  $T \leftarrow \text{CreatePrompt}(P, D_{\text{balance}}[j])$
  $x \leftarrow \text{InsertMask}(T, D_{\text{balance}}[j])$
end
for $k = 1$ ***to*** $|D_{balance}|$ **do**
  $y \leftarrow M(D_{\text{balance}}[k])$
end
for y ***in*** *ModelOutputs* **do**
  $\mathbf{v}_{\text{mask}} \leftarrow \text{Word2Vec}(y)$
  $\mathbf{v}_{\text{cat}} \leftarrow \text{Word2Vec}(C_{\text{predef}})$
  $L \leftarrow \text{CosineSim}(\mathbf{v}_{\text{mask}}, \mathbf{v}_{\text{cat}})$
end
for L ***in*** $L_{classified}$ **do**
  Evaluate(L)
end
return $L_{\text{classified}}$

tion is obtained by calculating the distance between the vectors. Details are as follows.

1) Sample Processing: The sample processing for this paper includes GD dataset establishment, sample cleansing, classification standards, and data balance. Details are provided in Section 4.2.

2) CPP: In the module of CPP, by appending specific patterns and masks to the input text, the input content is standardized, thereby guiding the LangChain module to better complete the classification task and achieve higher precision. Details are provided in Section 3.2.

3) LangChain: In the LangChain module, a fixed prompt template is first designed based on situation, task, answer format, and notes. Then, the CPP and input data are embedded into the prompt content. Subsequently, a chain is constructed using the model interface (GPT-3.5, GPT-4) in LangChain and the prompt template. Finally, the classification answer is obtained by continuously updating the prompt content and invoking the chain. Details are shown in Section 3.3.

4) CLM: In the module of CLM, an optimal Word2Vec model trained on the dataset is used to map all the classification answers into mask vectors while also mapping the predefined categories into category center vectors. Subsequently, the classification result is obtained by comparing the mask vector and the category center vector. The details are shown in Section 3.4.

3.2 CPP

To effectively utilize pre-training knowledge for downstream tasks, we introduce CPP to add context to the input. In addition, the classification results are obtained through mask prediction. We define CPP as follows:

Definition: Let M be a model provided by LLM, and T denote the token, which is the fundamental unit of the sample (such as a word or subword) that the model processes. Mask tokens are represented as $[_] \in T$, $x \in X$ is the sequence of tokens to be predicted, and $y \in Y$ corresponds to the ground truth label. The LLM employs a task-specific prompt function $f_p^n(_)$ to reformulate the input x into $\hat{x}$, inserting the mask token $[_]$ within it. Here, p is the pattern, and n is the index of the pattern. We fed these designed patterns into M and M is likely to pad $\hat{x}$ with a specific token at the masked location, thereby obtaining the detection token. We can formalize the prompt-based prediction as follows:

$$P(y|x) = \sum_{t \in T} P(y|t) P(t|f_p^n(x)) \tag{1}$$

where $P(t|f_p^n(x))$ is the probability of predicting token t given the prompted input $f_p^n(x)$, and $P(y|t)$ is the probability of label y given token t.

Prompt patterns: A prompt patterns consists of three parts: a string-type constant (defaultable), a mask and a GD text x. Each x contains a data source S and content C. For example, in patterns p_5 of formula (2), the sting-type constant is 'Category:'. We input the $patterns$ into LLMs, LLMs will output the type of GD text. In our work, we design 6 prompt patterns, as shown in formula (2).

$$\begin{aligned} &p_0 : f_p^0 = _ : SC && p_1 : f_p^1 = S(_)C \\ &p_2 : f_p^2 = _ - SC && p_3 : f_p^3 = SC(_) \\ &p_4 : f_p^4 = _classes : SC \\ &p_5 : f_p^5 = [Category : _]SC \end{aligned} \tag{2}$$

3.3 LangChain

LangChain is an open-source framework that simplifies the development of applications powered by

LLMs. It provides modular components and prebuilt chains to help developers harness the potential of LLMs for tasks like document analysis, chatbots, and more. The core capabilities of LangChain include context-aware applications that can leverage prompt instructions, examples, and existing content, as well as reasoning abilities that allow applications to make informed decisions based on the provided context. Overall, LangChain aims to make it easier for developers to build applications powered by the capabilities of LLMs.

In this paper, we utilize the LangChain interface to invoke GPT-3.5-Turbo and GPT-4 as LLMs to address the classification problem of open GD. The GD-LCF mainly consists of the Chat Model, Prompt Template, and Chain. The details of each component are as follows.

1) Chat Model: LangChain's Chat Model is a specialized language model designed for conversational interactions. Unlike typical language models that process a single input string and generate a single output, Chat Models handle a list of messages as input and produce a message as output. This conversational approach enabled by Chat Models facilitates more dynamic and interactive dialogues with the language model. It empowers the development of chatbot applications, customer support systems, and other multi-turn conversation-based use cases. By utilizing the ChatOpenAI component within LangChain's Chat Model, we enable the invocation of both GPT-3.5-Turbo and GPT 4.

2) Prompt Template: Prompt templates enable the structuring of inputs for LLMs. These methods offer a convenient approach to formatting user inputs and providing instructions to generate appropriate responses. Prompt templates help the LLMs comprehend the desired context and produce relevant outputs.

3) Chain: To better realize the effectiveness, the templates will be divided into situation, task, answer format, and note. The situation denotes the background of the classification data task. The task specifies the type of task the model is required to complete, the answer format emphasizes the desired output format, and the note emphasizes what the model should not output.

3.4 CLM

CLM is a module that accomplishes classification through the utilization of the Word2Vec model and vector computations, as shown in Figure 2, mainly consisting of the input layer, Word2Vec layer, Cosine similarity, and output layer. The details of each module are as follows.

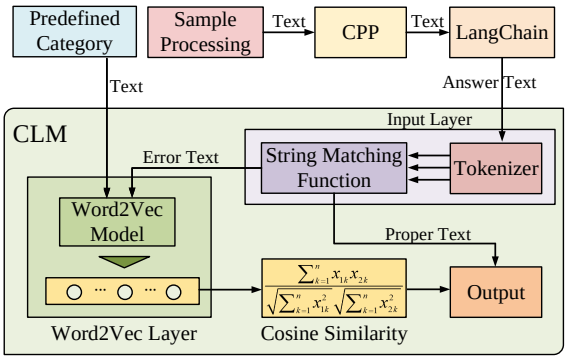

Figure 2: The structure of the CLM.

1) Input Layer: The main task of the input layer is to process the data output from LangChain, and filter out the invalid data. The input layer contains a tokenizer and a string matching function. The tokenizer will split the LangChain output, simply removing unnecessary content and extracting the key information. The string matching function can identify whether the mask predicted by the LangChain model belongs to one of the predefined categories. If the model's prediction result does not belong to the predefined categories, it will be filtered out as the input data for CLM.

2) Word2Vec Layer: After being processed by the input layer, the mask predicted by the LangChain model will be converted into a vector through a Word2Vec model. The Word2Vec model is based on a pre-trained model (Roberta) with inherent word vector conversion capabilities. It has been optimized by training on the dataset used in this work, enabling it to accomplish better the task of mapping words to vectors. Meanwhile, the predefined categories are also mapped to the same vector space through this Word2Vec model.

3) Cosine Similarly and Output Layer: This layer calculates the cosine similarity between the mask vector and each predefined category vector, and the most suitable predefined category is selected as the final classification result.

4 Experiment Evaluation

4.1 Experimental Environment

Experiments were conducted on a single GPU with 24GB RAM (NVIDIA GeForce RTX 4090). Training a single PET (Schick and Schütze, 2021) model with auxiliary language modeling for 300 steps on

Data Sources	Classification	Amount
Open Guangdong Platform	12	89,927
Fujian Province Public Information Open Platform	26	22,383
Beijing Municipal Public Data Open Platform	20	14,767
Sichuan Public Data Open Network	21	8,194
Tianjin Municipal Information Unified Open Platform	21	6,485
Shandong Public Data Open Network	20	4,655
Shanghai Municipal Public Data Open Platform	12	4,654
Zhejiang Government Service Network	22	1,146
Hainan Province Government Data Unified Open Platform	22	1,118
Ningxia Hui Autonomous Region Data Open Platform	24	803
Jiangsu Province Public Data Open Platform	20	772
Jiangxi Provincial Government Data Open Platform	12	386
totaling		155,290

Table 1: The list of data sources, their classification, and the amount of the dataset.

one GPU took around 30 minutes. Training the Word2Vec model for 5 epochs on one GPU took around 10 minutes. For training the Word2Vec model, we utilized 100,000 samples to pre-train the Roberta model and 5,000 samples for training. Concurrently, we opted for the Adam optimizer for prompt-tuning of our scheme, employing a skewed triangular scheme with an initial learning rate of 1e-5 and a maximum sequence length of 256. The batch size was set to 8.

4.2 The GD Dataset

To conduct the experiment effectively, the dataset used in this experiment and its processing are as follows. The sample processing is used to establish the GD dataset, sample cleansing, classification standards, and data balancing.

1) GD dataset establishment: In this paper, we create a dataset of Chinese GD. Firstly, to collect a large amount of GD, we investigate 12 provincial and autonomous government websites in China, including the data sources, categories, and quantities, as shown in Table 1. Secondly, crawler technology was used to obtain a large amount of GD, totaling 155,290.

2) Sample Cleaning: After creating the GD dataset, the next step is to ensure their quality and reliability. This process entails removing duplicate or irrelevant samples while rectifying any errors within the samples.

3) Classification Standards: To design a collection of categories with wide coverage and containing a relatively balanced amount of data, this paper summarizes the frequency of occurrence of each label obtained by the crawler. The experimental results proved that the less frequent categories have a large semantic overlap and need to merge. Therefore, not only are the labels with higher frequency selected, but some low-frequency labels are also semantically merged. After data organization and category classification, this paper divides the data categories into 10 general fields from the subject of the data, and the brief description, examples, and distribution of each category are shown in Table 2. This standard covers only the categories with the largest sample size and the widest range of fields.

4) Data Balance: To ensure the precision and reliability of the validation data, equalization was performed according to the categories to ensure the same amount for each category when training or testing.

4.3 Effentiveness Evaluation of our Scheme

Prompt: Unlike previous schemes that rely solely on data descriptions, our proposed prompt utilizes the chain of prompt processing mechanism to categorize input data efficiently. By leveraging 6 different CPP designs, the model dynamically adjusts its focus to optimize classification outcomes. The input prompt (e.g., "Beijing Municipal Administration...") is first processed by CPP's masking functions p_1 to p_6 , as shown in the CPP chain, which directly impacts the classification results.

Performance evaluation: To demonstrate the effectiveness of the proposed framework in the zero-shot learning scenario and to ensure data balance across categories, we randomly selected 500 samples from each predefined ten class, totaling 5,000 samples as the test set. Then, to ensure that the sample quality does not overly impact the model's performance, we divided the 5,000 samples into 5 groups, each containing 1,000 samples (100 per

Classification	Description	Proportion
People's livelihood services	Includes labor employment, consumption, shopping, legal services, etc.	0.378
Urban and rural construction	Includes urban planning, urban construction, industrial agriculture, etc.	0.125
Finance, taxation, and market	Includes tax policy, financial policy, fiscal budget, market supervision, etc.	0.124
Institutions and organizations	Include various types of organizational activities, such as government agencies and non-profit organizations	0.110
Public safety	Includes the maintenance of public order, the safety of public facilities, and the safety of public health	0.094
Science, education, culture, and tourism	Includes scientific and technological innovation, education and training, and cultural tourism	0.075
Resource environment	Includes the development and utilization of natural resources, environmental protection, and ecological restoration	0.046
Transportation	Includes various transportation information such as highway, railway, and water transportation	0.031
Credit Services	Includes the results of spot checks and inspections, warning lists, evaluation reports, etc.	0.013
Meteorology and geography	Include natural geographical factors such as climate, weather, and topography	0.004

Table 2: The list of classifications and their distribution within the dataset.

category). Therefore, for the 6 CPP designs we have created, each CPP requires 5 rounds of testing, and the tests are conducted on GPT-3.5-Turbo and GPT-4, resulting in a total of 60 experimental.

Method	p_0	p_1	p_2	p_3	p_4	p_5
PET(Schick and Schütze, 2021)	9.9	10.4	9.9	10.2	10.2	10.1
IPET(Schick and Schütze, 2021)	10.4	10.6	10.2	10.6	10.7	10.5
GPT-3.5-Turbo	58.5	56.5	57.1	54.7	59.0	56.1
GPT-4	68.4	68.9	68.7	67.6	66.7	67.1
GD-LCF(GPT-3.5)	59.0	60.0	60.9	60.1	**63.5**	56.2
GD-LCF(GPT-4)	74.2	**75.3**	74.2	73.3	74.6	74.4

Table 3: Comparing the effectiveness of GD-LCF with other prompt engineering schemes in zero-shot.

Zero-shot comparison: The performance of each CPP is benchmarked against advanced schemes such as PET, IPET, standard GPT-3.5-Turbo, and GPT-4 in zero-shot. The experimental results for all patterns represent the average of multiple training runs. As shown in Table 3, the precision of PET and IPET, as traditional prompt engineering schemes, performed poorly in all patterns: PET's precision ranged from 9.9% to 10.4%, while IPET's precision ranged from 10.4% to 10.5%. This shows that in the absence of training data, it is difficult for traditional prompt engineering schemes to effectively solve complex classification problems. Compared to PET and IPET, the existing new schemes GPT-3.5-Turbo and GPT-4 perform better, with precision ranges of 54.7%-59% and 68.4%-74.6% in p_0 to p_5, respectively. In contrast, our proposed GD-LCF is a significant improvement over GPT-3.5-Turbo and GPT-4, especially in GPT-4 (p_1), which achieves 75.3% precision, the highest precision of all schemes. This result fully shows that compared with other prompt engineering schemes, our GD-LCF framework significantly improves the classification precision, especially when combined with GPT-4, its performance is far superior to the traditional prompt engineering scheme, effectively proving the effectiveness and stability of GD-LCF in automatic classification of task weight.

Few-show comparison: In addition, we compared the effectiveness of GD-LCF training with traditional deep learning method (including methods based on traditional deep learning and BERT models) on small amounts of data (due to token length limitations), and the results are shown in Table 4. With the increase of training samples (TS), the effectiveness of each scheme is improved, among which, the Recall of GD-LCF shows a steady upward trend, and the F1 score is even increased from 0.61 to 0.73. In addition, under the premise that the TS are 0,10,100 respectively, the precision of the traditional model and BERT series is only about 10, and the precision of the Roberta series is between 14-21, both of which are far less than GD-LCF.

4.4 Ablation Study

To evaluate the effectiveness of GD-LCF in zero-show, we conducted an ablation study on the GD dataset, as shown in Table 5 for details. Each row in the table represents the selected configuration (indicated by a checkmark (✓)) and its precision in the ablation study. For example, in Table 5, row

	TS=0			TS=10			TS=100		
Method	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Johnson and Zhang (2017)	9.7	0.10	0.03	9.9	0.10	0.03	10.0	0.10	0.02
Joulin et al. (2016)	10.2	0.10	0.07	10.8	0.11	0.07	12.6	0.13	0.12
Zhang et al. (2019)	10.1	0.10	0.04	10.4	0.10	0.07	11.8	0.12	0.09
Liu et al. (2016)	9.8	0.10	0.02	9.8	0.10	0.05	10.8	0.11	0.08
Lai et al. (2015)	9.5	0.09	0.02	9.7	0.10	0.03	10.4	0.10	0.03
Zhou et al. (2016)	10.1	0.10	0.02	10.8	0.11	0.04	10.0	0.10	0.02
Vaswani (2017)	10.1	0.10	0.03	10.1	0.10	0.04	10.0	0.10	0.02
Devlin et al. (2019)	10.9	0.11	0.04	11.4	0.11	0.06	10.0	0.10	0.02
Tian et al. (2023)	12.6	0.11	0.06	14.4	0.14	0.11	10.5	0.10	0.03
Wang et al. (2022)	13.2	0.12	0.08	14.4	0.14	0.09	11.5	0.11	0.05
Belherazem and Tlemsani (2022)	10.6	0.10	0.02	11.2	0.11	0.06	10.7	0.11	0.04
Ma et al. (2021)	9.7	0.10	0.04	10.0	0.10	0.02	10.3	0.10	0.03
Sun et al. (2019)	9.1	0.08	0.05	9.5	0.10	0.08	12.2	0.12	0.08
Liu (2019)	14.5	0.15	0.14	16.2	0.17	0.16	15.7	0.18	0.16
Hambro et al. (2022)	16.3	0.16	0.15	19.3	0.19	0.16	20.7	0.21	0.24
GD-LCF	**75.3**	**0.65**	**0.61**	**76.5**	**0.75**	**0.72**	**78.7**	**0.77**	**0.73**

Table 4: Comparing the effectiveness of GD-LCF with traditional deep learning schemes. Where 'TS' is the training samples.

4 shows that under the combination of GPT-4 and our GD-LCF, the precision of data classification is 74.33, which achieves the optimal effect compared with other schemes, indicating that GD-LCF is significantly superior to existing schemes in terms of classification precision when the training data is little or even zero.

Number	GPT3.5	GPT4	GD-LCF	P
1	✓			59.0
2		✓		68.9
3	✓		✓	63.5
4		✓	✓	**75.3**

Table 5: The results of the ablation study in zero-shot. Where 'P' represents the precision of classification).

The above experiments show that GD-LCF framework shows great advantages and potential in handling zero-sample and small-sample classification tasks, and in GD classification scenarios, it can provide stronger classification capabilities than traditional language models at a lower cost of human resources.

5 Conclusion

This paper integrates prompt-tuning with LangChain to propose a classification framework (GD-LCF) for GD. This scheme tackles the challenges of the scarcity of high-quality labeled samples in the GD domain. Furthermore, we incorporated clustering mapping techniques to improve classification precision and efficiency. Experiments conducted on GD datasets demonstrate that our approach achieves high precision in classifying GD, surpassing existing GD classification schemes.

6 Limitations

Despite the significant progress made by the GD-LCF framework in the classification of government data, there are still some limitations. Firstly, the scheme is highly dependent on the design of the CPP. If the prompt content lacks diversity or is not sufficiently relevant to the context, it may lead to classification results that fail to capture the nuances of the target domain. Additionally, although there have been improvements in domain adaptability, the scheme's performance still varies across different domains, indicating that further optimization is needed when dealing with complex and highly diverse fields such as GD. Future research should focus on enhancing the prompt generation process and exploring more sophisticated methods to address the diverse characteristics of these domains. Furthermore, invoking the LangChain interface for classification tasks can be resource-intensive, which needs to be considered for efficiency in large-scale implementations. Nevertheless, these costs are generally regarded as acceptable in practical applications.

Acknowledgments

References

Assia Belherazem and Redouane Tlemsani. 2022. Boosting convolutional neural networks using a bidirectional fast gated recurrent unit for text categorization. *International Journal of Artificial Intelligence and Machine Learning (IJAIML)*, 12(1):1–20.

Jonathan Crusoe and Antoine Clarinval. 2023. Classification of open government data solutions' help: A novel taxonomy and cluster analysis. In *Electronic Government - 22nd IFIP WG 8.5 International Conference, EGOV 2023, Budapest, Hungary, September 5-7, 2023, Proceedings*, volume 14130, pages 230–245.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Lijie Feng, Lehu Zhang, Jinfeng Wang, and Jian Feng. 2024. How to promote the participation of enterprises using open government data? evolutionary game analysis by applying dynamic measures. *Expert Syst. Appl.*, 238(Part F):122348.

Eric Hambro, Roberta Raileanu, Danielle Rothermel, Vegard Mella, Tim Rocktäschel, Heinrich Küttler, and Naila Murray. 2022. Dungeons and data: A large-scale nethack dataset. *Advances in Neural Information Processing Systems*, 35:24864–24878.

Weidong Huang, Chong Su, and Yuan Wang. 2021. An intelligent work order classification model for government service based on multi-label neural network. *Comput. Commun.*, 172:19–24.

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 562–570.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Pajak Krzysztof and Dominik Pajak. 2022. Multilingual fine-tuning for grammatical error correction. *Expert Syst. Appl.*, 200:116948.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2267–2273.

Bin Liu. 2022. GCN-BERT and memory network based multi-label classification for event text of the chinese government hotline. *IEEE Access*, 10:109267–109276.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Xu Ma, Jingda Guo, Andrew Sansom, Mara McGuire, Andrew Kalaani, Qi Chen, Sihai Tang, Qing Yang, and Song Fu. 2021. Spatial pyramid attention for deep convolutional neural networks. *IEEE Transactions on Multimedia*, 23:3048–3058.

Eduardo Nascimento, Grettel García, Wendy Victorio, Melissa Lemos, Yenier Izquierdo, Robinson Luiz Souza Garcia, Luiz André P. Paes Leme, and Marco A. Casanova. 2023. A family of natural language interfaces for databases based on chatgpt and langchain (short paper). In *Companion Proceedings of the 42nd International Conference on Conceptual Modeling: ER Forum, 7th SCME, Project Exhibitions, Posters and Demos, and Doctoral Consortium co-located with ER 2023, Lisbon, Portugal, November 06-09, 2023*, volume 3618.

Pei Pan and Yijin Chen. 2021. Automatic subject classification of public messages in e-government affairs. *Data Inf. Manag.*, 5(3):336–347.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 255–269.

Aditi Singh, Abul Ehtesham, Saifuddin Mahmud, and Jong-Hoon Kim. 2024. Revolutionizing mental health care through langchain: A journey with a large language model. In *14th IEEE Annual Computing and Communication Workshop and Conference, CCWC 2024, Las Vegas, NV, USA, January 8-10, 2024*, pages 73–78. IEEE.

Yajun Song, Zeyuan Li, Jie He, Zesong Li, Xin Fang, and Dagang Chen. 2019. Employing auto-annotated data for government document classification. In *ICIAI 2019: The 3rd International Conference on Innovation in Artificial Intelligence, Suzhou, China, March 15-18, 2019*, pages 121–125.

Fatih Soygazi and Damla Oguz. 2023. An analysis of large language models and langchain in mathematics education. In *Proceedings of the 2023 7th International Conference on Advances in Artificial Intelligence, ICAAI 2023, Istanbul, Turkiye, October 13-15, 2023*, pages 92–97. ACM.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: enhanced representation through knowledge integration. *CoRR*, abs/1904.09223.

Keyu Tian, Yi Jiang, Qishuai Diao, Chen Lin, Liwei Wang, and Zehuan Yuan. 2023. Designing bert for convolutional networks: Sparse and hierarchical masked modeling. *arXiv preprint arXiv:2301.03580*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Qian Wang, Hefei Ling, Baiyan Zhang, Ping Li, Zongyi Li, Yuxuan Shi, Chengxin Zhao, and Chuang Zhao. 2022. Bidirectional gated edge-labeling graph recurrent neural network for few-shot learning. *IEEE Transactions on Cognitive and Developmental Systems*, 15(2):855–864.

Yasuhiro Yamada and Tetsuya Nakatoh. 2018. Tag recommendation for open government data by multi-label classification and particular noun phrase extraction. In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2018, Volume 3: KMIS, Seville, Spain, September 18-20, 2018*, pages 81–89.

Yuebing Zhang, Zhifei Zhang, Duoqian Miao, and Jiaqi Wang. 2019. Three-way enhanced convolutional neural networks for sentence-level sentiment classification. *Information Sciences*, 477:55–64.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.