Prompt4LJP: prompt learning for legal judgment prediction

Prompt4LJP: Prompt Learning for Legal Judgement Prediction

Qiongyan Huang[1], Yuhan Xia[4], Yunfei Long[4], Hui Fang[2,3*], Ruiwei Liang[1], Yin Guan[2,3†], Ge Xu[2,3†]

[1]College of Mechanical and Electrical Engineering, Fujian Agriculture and Forestry University, Fuzhou, 350100, Fujian, China.
[2]School of Computer and Big Data, Minjiang University, Fuzhou, 350108, Fujian, China.
[3]Fujian Mental Health Human-Computer Interaction Technology Research Center, Minjiang University, Fuzhou, 350108, Fujian, China.
[4]School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, Essex, UK.

*Corresponding author(s). E-mail(s): fanghui_ce@mju.edu.cn;
Contributing authors: huangqy@fafu.edu.cn; yx23989@essex.ac.uk; yl20051@essex.ac.uk; liangrw@fafu.edu.cn; niynaug@mju.edu.cn; xuge@pku.edu.cn;
†These authors contributed equally to this work.

Abstract

The task of Legal Judgment Prediction (LJP) involves predicting court decisions based on the facts of the case, including identifying the applicable law article, the charge, and the term of penalty. While neural methods have made significant strides in this area, they often fail to fully harness the rich semantic potential of language models (LMs). *Prompt learning* is a novel paradigm in Natural Language Processing (NLP) that reformulates downstream tasks into cloze-style or prefix-style prediction challenges by utilizing specialized prompt templates. This paradigm shows significant potential across various NLP domains, including short text classification. However, the dynamic word lengths of LJP labels present a challenge to the general prompt templates designed for single-word [MASK] tokens commonly used in many NLP tasks. To address this gap, we introduce the *Prompt4LJP* framework, a new method based on the prompt learning paradigm for the complex LJP task. Our framework employs a dual-slot prompt template in conjunction with a correlation scoring mechanism to maximize the utility of LMs

without requiring additional resources or complex tokenization schemes. Specifically, the dual-slot template consists of two distinct slots: one dedicated to factual descriptions and the other to labels. This approach effectively tackles the challenge of dynamic word lengths in LJP labels, reformulating the LJP classification task as an evaluation of the applicability of each label. By incorporating a correlation scoring mechanism, we can identify the final result label. The experimental results show that our *Prompt4LJP* method, whether using discrete or continuous templates, outperforms baseline methods, particularly in charges and terms of penalty prediction. Compared to the best baseline model EPM, Prompt4LJP shows F1-score improvements of 2.25% and 4.76% (charge prediction and term of penalty prediction) with discrete templates, and 3.24% and 4.05% with the continuous template, demonstrating *prompt4LJP* ability to leverage pre-trained knowledge and adapt flexibly to specific tasks. The source code can be obtained from https://github.com/huangqiongyannn/Prompt4LJP

Keywords: natural language processing, legal judgement prediction, prompt learning, legal application, masked language model

1 Introduction

Legal Judgment Prediction (LJP) aims to predict court decisions based on case facts, encompassing tasks such as law article prediction, charge prediction, and term of penalty prediction. Specifically in Figure 1, the text highlighted in red signifies key details extracted from the factual description, while the content highlighted in blue denotes the relevant law article, charge, and term of penalty applicable to the fact description. Substantial advancements in LJP have been achieved using sophisticated neural networks and text representation models [1–3]. For example, Luo et al. [1] improved the integration of factual descriptions with law articles using attention mechanisms. However, these neural methodologies primarily focus on extracting in-domain information from LJP datasets, often neglecting the rich semantic and linguistic information available in language models (LMs). Current approaches typically treat LJP tasks as straightforward text classification problems [4], which limits their effectiveness in leveraging the comprehensive legal knowledge embedded in LMs.

Prompt learning, a transformative approach that reformulates downstream tasks as cloze-style or prefix-style prediction challenges for LMs, including Masked Language Models (MLMs) using specialized prompt templates, has shown considerable promise across various Natural Language Processing (NLP) domains [5–11]. One of the primary advantages of prompt learning is that it enables models to better understand downstream tasks, thereby stimulating the recall of relevant knowledge embedded in LMs [12, 13]. However, current prompt learning templates are not well-suited for LJP tasks due to the nature of LJP labels. The dynamic word lengths of LJP labels present a challenge to the standard prompt templates designed for single-word [MASK] token commonly used in many NLP tasks. LJP labels often consist of complex, multi-word expressions, such as legal charges like “intentional injury” or legislative references like “Article 256,” which cannot be adequately captured by single [MASK] token. This

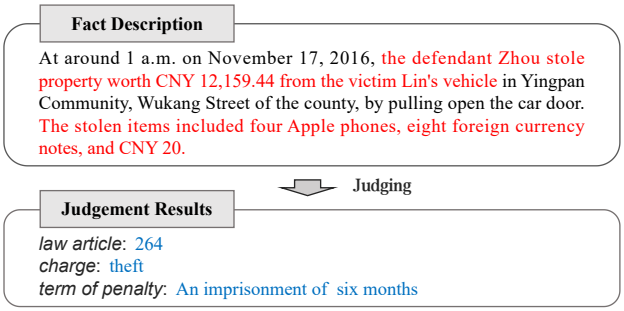

Fig. 1 An illustration of legal judgment prediction.

misalignment hinders the effective use of LMs' extensive pre-trained knowledge in LJP tasks.

To address this gap, we introduce the *Prompt4LJP* framework, a pioneering method that effectively utilizes LMs knowledge tailored for the complex and dynamic nature of LJP labels. Our contributions are centered on harnessing the extensive pre-trained knowledge of LMs and enhancing their ability to recall and apply relevant legal information through two main technical innovations:

First, we developed a Dual-Slot Prompt Template that directly incorporates the given fact description and a potential label into the prompt, respectively, transforming LJP tasks into masked language model challenges. This design engages the language model to apply its learned semantic knowledge and intuitively grasp the task through structured template guidance, accommodating the complex, multi-word labels typical in LJP.

Second, we introduced a novel Correlation Scores Ranking to assess candidate labels generated for each fact scenario. Although LJP tasks typically involve a single label per case, in real judicial applications, charges or law articles can be very similar. The ranking mechanism generates candidate labels that can assist in practical judicial decision-making by providing closely related alternatives and identifying the most accurate ground-truth label. This system significantly enhances the LM's capacity to leverage its extensive pre-trained legal knowledge, thereby boosting both accuracy and reliability in LJP tasks.

To evaluate our Prompt4LJP method, we conducted rigorous testing on the CAIL2018-small dataset [14], a recognized benchmark in the LJP field. The results are highly promising, showing that Prompt4LJP not only meets but often exceeds the performance of existing state-of-the-art (SOTA) neural models in predicting charges and terms of penalty. These outcomes highlight the efficacy of our tailored prompt template in adeptly managing multi-word labels and markedly improving the accuracy and reliability of LJP systems.

Our findings demonstrate that the Prompt4LJP framework effectively utilizes the rich, pre-trained knowledge embedded in LMs, optimizing it specifically for LJP tasks without the necessity for additional external datas. This framework stimulates LMs to recall pre-trained legal knowledge relevant to LJP tasks, significantly enhancing their

performance. Furthermore, this significant advancement in applying LMs knowledge directly addresses the unique demands of LJP, setting a new standard in the field.

2 Related Work

In this section, we review our related work. Section 2.1 covers existing methods for LJP and highlights their strengths and weaknesses. Section 2.2 discusses prompt learning in LJP and its limitations compared to our approach.

2.1 Legal Judgement Prediction

Traditional methods to LJP have primarily relied on rule-based systems or mathematical models [15–17]. While these methods have demonstrated notable accuracy, their applicability is often limited due to the labor-intensive process of feature engineering and the challenge of generalizing across diverse datasets. The intricate nature of manually crafting features restricts the scalability and adaptability of these approaches, especially in complex, real-world legal scenarios.

The advent of neural network techniques in NLP has revolutionized LJP research, shifting the focus from traditional methods to more dynamic, data-driven approaches. Neural networks provide the ability to automatically learn representations from textual data, which has spurred numerous studies aiming to enhance LJP accuracy and generalizability [1–3, 18–22]. This paradigm shift has not only improved prediction performance but also enabled the modeling of complex semantic relationships within legal texts.

A prominent trend in recent research involves employing a multi-task learning (MTL) framework to model the subtasks of LJP. By addressing related tasks simultaneously, MTL captures dependencies among them, leading to more nuanced predictions. For instance, Zhong et al. [18] proposed a topological multi-task learning framework using a Directed Acyclic Graph (DAG) to model dependencies among LJP subtasks, capturing hierarchical relationships between law articles, charges, and terms of penalty, leading to improved prediction accuracy and coherence. Experiments on large-scale civil law datasets showed significant performance improvements over single-task and basic multi-task models. Meanwhile, Feng et al. [2] proposed an Event-based Prediction Model (EPM) with constraints, focusing on identifying key event information and leveraging cross-task consistency across LJP subtasks. Their approach achieves superior performance compared to existing SOTA models on benchmark LJP datasets. Similarly, Zhang et al. [3] proposed a supervised contrastive learning framework, using contrastive learning to address ambiguity in legal terminology by distinguishing fine-grained differences in law articles and charges. Their method enhances LJP accuracy and improves fact-label relationships, achieving superior performance on real-world datasets.

Despite their advantages, MTL-based methods introduce additional complexities. Effective implementation requires carefully balancing task weights and designing task-specific loss functions, which can complicate hyperparameter tuning and model optimization. Inspired by the success of prompt learning in various NLP domains [8, 11], which provides a lightweight yet powerful alternative to traditional deep

learning approaches by aligning model inputs with task-specific cues, we propose the Prompt4LJP framework. This framework aims to harness the potential of prompt learning to improve LJP, leveraging its ability to efficiently integrate contextual information and task-specific signals to enhance prediction performance.

2.2 Prompt Learning

Prompt learning, facilitated by pre-trained language models such as BERT [23] and GPT [24], has revolutionized various NLP tasks by framing them as cloze-style or prefix-style prediction challenges. Although successful in domains like news recommendation [8], implicit discourse relationship recognition [11], and text classification [5–7], its application in LJP faces unique challenges due to the complexity of legal terminology and the often complex, multi-word expressions that LJP labels entail.

Sun et al. [25] tackled the challenge of representing intricate legal charges by employing a fixed template with ten [MASK] tokens. This approach, while innovative, resulted in significant data sparsity from the numerous possible combinations of the [MASK] tokens. To mitigate this issue, they incorporated external knowledge bases to enrich contextual understanding, though their method depended on calculating the similarity between predicted outputs and actual legal terms, which can be problematic due to nuanced differences between similar terms.

Prompt4LJP diverges significantly by simplifying the integration of labels and facts. Our model uses a dual-slot prompt template that directly incorporates the given fact and a potential label into the prompt. This method efficiently evaluates correlation scores between facts and labels, converting the traditional multi-class classification challenge into a more straightforward binary prediction task. The Prompt4LJP framework guides LMs to leverage their extensive pre-trained knowledge more effectively, enhancing their ability to recall and apply relevant legal information for LJP tasks. This enhancement not only eliminates the need for external data sources and complex tokenization strategies but also increases the accuracy and applicability of prompt learning for LJP.

3 Our proposed method

In this section, we first give the essential definitions of LJP task. Then, we explain the details of our prompt templates, verbalizer and answer words. Finally, we present a comprehensive overview of our Prompt4LJP framework and detail the training strategy employed.

3.1 Task Definition

The three subtasks in LJP are denoted as t_a, t_c, and t_b, representing law article prediction, charge prediction, and term of penalty prediction, respectively. Each subtask is a multi-label classification task. To maintain consistency with previous studies [2, 3, 20], we consider only samples in which each subtask has a single label in the dataset. Given the factual description x of a legal case, the LJP task, denoted as $T = \{t_a, t_c, t_b\}$, aims to predict the result labels for the three subtasks. Formally,

y_i^t represents the i-th label of a subtask-specific label set Y^t, where $y_i^t \in Y^t$ and $i = 1, 2, ..., |Y^t|$ for $t \in T$. For instance, in the charge prediction subtask t_c, the label set $Y^c = \{Theft, Robbery, ..., Arson\}$ includes $y_1^c = Theft$, $y_2^c = Robbery$, and $y_{|Y^c|}^c = Arson$.

3.2 Dual-Slot Prompt Template for LJP

Within our research, each subtask—t_a, t_c, and t_b—is treated as an independent task, each utilizing a consistent prompt template format. Unlike conventional multi-class classification templates, which typically feature only one input slot for a single-word label or its expanded word from the LM's vocabulary, our templates include two slots: $< fact >$ and $< label >$. Specifically, $< fact >$ represents the fact statement x, while $< label >$ corresponds to the label y_i^t. Here, $< label >$ serves as a unified representation of $< crime >$, $< law >$, and $< term >$, representing charges, law articles, and terms of penalty, respectively. For instance, in the charges prediction task t_c, we construct a prompt template denoted as $f_{prompt}^c(< fact >, < crime >)$ as follows:

According to the following fact, whether the defendant is guilty of $< crime >$ *: [MASK].* $< fact >$.

Our method innovatively converts the multi-class classification task into a cloze-style mask-prediction task within the template. Prompt-guided MLMs aim to predict whether a given label y_i^t is a plausible result label for the factual description x. Additionally, we explore the impact of employing the continuous prompt on prediction performance for LJP. In our continuous template, we preserve the structure of discrete prompts but substitute discrete tokens on them with custom pseudo tokens. These pseudo tokens are arbitrarily chosen characters without intrinsic meaning, and their embeddings are initialized randomly. Here, we denote these pseudo tokens as $[P_{1:l_1}]$, $[Q_{1:l_2}]$, and $[M_{1:l_3}]$. Serving as learnable parameters within the MLMs, these pseudo tokens are strategically placed before $< label >$, [MASK], and $< fact >$, respectively, where l_1, l_2, and l_3 specify the number of pseudo tokens. For further clarity, Table 1 provides an overview of our designed prompt templates for the three subtasks.

Table 1 Prompt templates designed for the three subtasks in this paper, including discrete and continuous templates.

TYPE	TASK	TEMPLATE
Discrete	Law Articles	According to the following fact, whether the defendant violates Article $< law >$ of the Criminal Law: [MASK]. $< fact >$
	Charges	According to the following fact, whether the defendant is guilty of $< crime >$: [MASK]. $< fact >$
	Terms of Penalty	According to the following fact, whether it is reasonable to impose a $< term >$ punishment on the defendant: [MASK]. $< fact >$
Continuous	All	$[P_1]...[P_{l_1}] < label > [Q_1]...[Q_{l_2}]$ [MASK] $[M_1]...[M_{l_3}] < fact >$

3.3 Answer Words and Verbalizer

Based on our provided prompt template $f^t_{prompt}(< fact >,< label >)$, we simply select two opposite words from the vocabulary V of the MLMs as our answer words, specifically *yes* and *no*. These two words constitute our answer space V_a, where $V_a = \{no, yes\} \subset V$. In MLMs M, the probability of filling each word v from V_a into the [MASK] can be calculated as in Equation (1):

$$P(v \in V_a | x, y^t_i) = P_M(w | f^t_{prompt}(x, y^t_i)) \tag{1}$$

where w represents filling the $[MASK]$ with the answer word $v \in V_a$, i.e., $[MASK] = v$. As we don't directly use labels as answer words, the primary emphasis of our work doesn't focus on the construction of the verbalizer. Nonetheless, within the prompt learning paradigm, the verbalizer holds significance. Thus, we provide a simplified formulation for our work:

$$f_{verbalizer}(v) = \begin{cases} x \Rightarrow y^t_i & v = yes \\ x \not\Rightarrow y^t_i & v = no \end{cases} \tag{2}$$

In this formulation, $x \Rightarrow y^t_i$ indicates that y^t_i is a possible result label for x, while $x \not\Rightarrow y^t_i$ indicates the absence of y^t_i as a potential result label for x.

3.4 Our Prompt4LJP Framework

For each given factual description x_i, we aim to predict the ground-truth label $\hat{y}^t_i$ for the three subtasks—t_a, t_c, and t_b—respectively. Figure 2 illustrates our Prompt4LJP framework, which encompasses three steps for legal judgments: prompts construction, candidate labels prediction, and correlation scores ranking.

Step1:Prompts Construction. In subtask t, we generate $|Y^t|$ prompts for each given fact x_i. Formally, we denote $f^t_{prompt,i,j}(x_i, y^t_j)$ as the j-th prompt associated with the fact x_i, where x_i and y^t_j are inserted into the prompt template's $< fact >$ and $< label >$ slots, respectively. We represent these $|Y^t|$ prompts for x_i as a prompt set $D^t_i = \{f^t_{prompt,i,j}(x_i, y^t_j) \mid y^t_j \in Y^t\}$ in the subtask t. The prompt whose $< label >$ corresponds to the target label $\hat{y}^t_i$ is called the *positive prompt*, while the remaining $|Y^t| - 1$ prompts, which contain labels from Y^t excluding the label corresponding to $\hat{y}^t_i$, are termed *negative prompts*.

Step2:Candidate Labels Prediction. Denoted $s^t_{i,j}$ as the correlation score between x_i and y^t_j. The correlation score can be served as the confidence whether the label y^t_j is a result label for x_i. The higher correlation score, the greater probability that y^t_j is the ground-truth label for x_i. For the k-th prompt $f^t_{prompt,i,k}(x_i, y^t_k)$ of x_i, if the probability of the answer word $v = yes$ is greater than that of $v = no$, i.e., $P(v = yes | x_i, y^t_k) > P(v = no | x_i, y^t_k)$, we consider x_i and y^t_k to be correlated, with a correlation score of $s^t_{i,k} = P(v = yes | x_i, y^t_k)$. Then, we create a pair consisting of the corresponding label and the correlation score, denoted as $< y^t_k, s^t_{i,k} >$, and add it into the candidate label set C^t_i. When the same evaluation is applied to all prompts in the

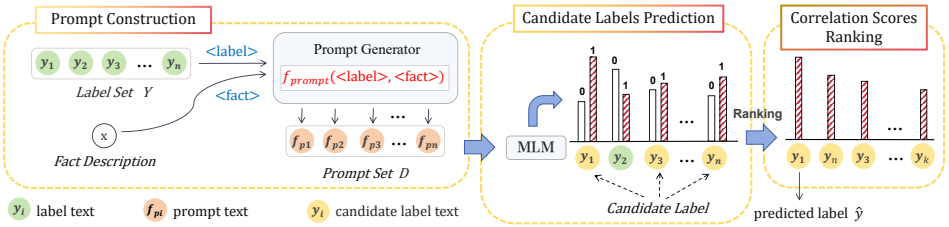

Fig. 2 The framework of Prompt4LJP. The three areas represent the three steps: Prompts Construction, Candidate Labels Prediction, and Correlation Scores Ranking.

prompt set D_i^t, we will obtain a candidate label set C_i^t for the fact x_i in the subtask t, where $0 \leq |C_i^t| \leq |Y^t|$.

Step3:Correlation Scores Ranking. We will sort the pair $< y_j^t, s_{i,j}^t >$ in the candidate label set C_i^t by the correlation score $s_{i,j}^t$ in order of largest to smallest. The label y_k^t corresponding to $< y_k^t, s_{i,k}^t >$ with the highest correlation score $s_{i,k}^t$ will be served as the final prediction label $\hat{y}_i^t$ for the fact x_i in the subtask t, which can be formalized as in Equation (3):

$$\hat{y}_i^t = g\left(\max\{s_{i,j}^t \mid \langle y_j^t, s_{i,j}^t \rangle \in C_i^t\}\right),\ t \in T \tag{3}$$

where g is a function finding the corresponding label y_k^t of the highest correlation score $s_{i,k}^t$ from the candidate label set C_i^t.

3.5 Training

We fine-tune the parameters of a MLM using the public CAIL2018-small [14] dataset and LAIC2021 dataset[1] with our custom prompt templates and answer space. For each subtask t, we use the cross-entropy loss function [26] to train the corresponding model:

$$L^t = \frac{1}{K}\sum_{k=1}^{K}[z_k^t log p_k^t + (1 - z_k^t) log(1 - p_k^t)] \tag{4}$$

where z_k^t and p_k^t are the gold label and predicted probability of the k-th training instance in the subtask t, respectively. We use the AdamW optimizer [27] with L2 regularization for model training.

4 Experiments

In this section, we introduce our experimental settings and conduct a series of experiments to evaluate the proposed Prompt4LJP framework.

[1] http://data.court.gov.cn/pages/laic2021.html

4.1 Experimental Settings

Dataset. We validated our method using the publicly available dataset from the Chinese AI and Law challenge [14]: CAIL2018-small. Each sample includes a factual description of a legal case, applicable law articles, charges, and terms of penalty. To align with state-of-the-art methods like CL4LJP [3], we used their preprocessing pipelines, filtering out samples with multiple labels. Statistical details of the datasets are shown in Table 2. Additionally, we tested the generalizability of our method with the newly released dataset of Legal Artificial Intelligence Challenge (LAIC) 2021[1], similar in content to CAIL2018-small. For LAIC2021, we used the data processing pipelines from ML-LJP [22]. Statistical details of the datasets are shown in Table 2.

Table 2 Statistics on datasets of CAIL2018-small and LAIC2021.

Dataset	CAIL2018-small	LAIC2021
#Training Set Cases	96,540	79,169
#Validation Set Cases	12,903	9,896
#Testing Set Cases	24,848	9,897
#Law Articles	101	70
#Charges	117	42
#Term of Penalty	11	9

Implementation Details. We utilize the pre-trained language model bert-base-chinese [23] provided by HuggingFace transformers [28]. For consistency across all subtasks of LJP, we employ the same training strategy. Specifically, the model is trained on 4 NVIDIA GeForce RTX 3090 GPUs simultaneously, with a batch size of 16 for each GPU. We use the AdamW optimizer [27] with a learning rate of 2e-5 and train the model for 8 epochs. The final epoch model is evaluated on the testing set. To assess the performance of our methods and baseline models, we employ four widely-used metrics for multi-class classification tasks: accuracy (Acc.), macro-precision (MP), macro-recall (MR), and macro-F1 (F1).

4.2 Baseline Methods

To fully demonstrate the effectiveness and superiority of our methodology in LJP tasks, emphasizing its capability to leverage knowledge acquired during the pre-training process, we conducted comparisons with three fundamental paradigms: SOTA neural networks, LLM-specific techniques such as prompt-based in-context learning (ICL), and parameter-efficient fine-tuning (PEFT).

Neural Methods. We selected seven SOTA neural models in Chinese LJP as our baseline models, and the full names of all models are shown in Table 3, including:

1. ***MLAC*** [1], which is an attention-based neural network method for charge prediction that incorporates the k most relevant law articles.
2. ***TOPJUDGE*** [18], which constructs a topological multi-task learning framework to capture dependencies among the three subtasks of LJP.

3. ***MPBFN-WCA*** [19], which utilizes dependencies among the three subtasks and integrates word collocation features of fact descriptions into the network via an attention mechanism to distinguish similar cases.
4. ***LADAN*** [20], which proposes a graph neural network to capture discriminative features between confusing law articles.
5. ***NeurJudge*** [21], which separates the factual description into several parts, each making a judgment for other subtasks.
6. ***EPM*** [2], which leverages key event information of legal cases to predict the result and utilizes consistency constraints between the three subtasks.
7. ***CL4LJP*** [3], which introduces a neural contrastive learning framework to capture the relationship between factual descriptions, similar law articles, and corresponding charges.
8. ***ML-LJP*** [22], which proposes a novel multi-law aware LJP method to enhance LJP by extracting label-specific features from facts and capturing high-order interactions among multiple law articles.

Table 3 The full names of SOTA neural models for LJP

Model	Full Name
MLAC [1]	-
TOPJUDGE [18]	A Topological Multi-Task Learning Framework for LJP
MPBFN-WCA [19]	A Multi-Perspective Bi-Feedback Network with the Word Collocation Attention Mechanism
LADAN [20]	A Law Article Distillation-based Attention Network
NeurJudge [21]	A Circumstance-Aware Neural Framework for LJP
EPM [2]	An Event-Based Prediction Model with Constraints
CL4LJP [3]	A Contrastive Learning Framework for the LJP Task
ML-LJP [22]	A Novel Multi-Law Aware LJP Framework

PEFT Method. Considering the limited computational resources, we chose the newly released Qwen1.5-1.8B-Chat with small parameters [29], which is tailored for Chinese, as our backbone for LoRA fine-tuning [30] on the CAIL2018-Small training set. The fine-tuned model is denoted as ***PEFT-Qwen1.5***. We utilized the LLama Factory [31] for LoRA fine-tuning on Qwen1.5-1.8B-Chat, leveraging 4 NVIDIA GeForce RTX 3090 GPUs. Specifically, we transformed the original CAIL2018-small dataset into the "instruction-input-output" JSON format tailored for the model, as depicted in Table 4. The model hyperparameters were fine-tuned based on the training set of CAIL2018-small, with detailed settings provided in Table 5.
ICL Method. Evaluation results from [4] indicate that the Qwen7B-Chat model [29] exhibits the best performance on legal tasks among Chinese-oriented LLMs. Therefore, we selected the latest known Qwen model, Qwen1.5-14B-Chat, as our backbone for zero-shot and few-shot experiments. For the given descriptions, we constructed suitable prompts to guide the model in providing the applicable law article, charge, and term of penalty. Due to the maximum input token limit of the model, we only conducted 0-shot, 2-shot, and 4-shot experiments. Experimental results show that the 4-shot

Table 4 The JSON format for fine-tuning, labeled as "instruction-input-output," assigns roles as follows: "system" describes the LJP task, "user" pertains to data input, and "assistant" denotes model output.

Role	Message
system	You will participate in a legal judgment prediction task. Given a set of case facts, you need to predict the applicable laws, charges, and possible terms of penalty. Please use your legal knowledge and logical reasoning to make predictions.
user	**Fact description**: On the evening of October 4, 2012, the defendant, Luo Moujia, ...
assistant	**Law**: Article 234 of the Criminal Law of the People's Republic of China **Charge**: Intentional Injury **Penalty terms**: Imprisonment for less than six months

setup yields the best results. Table 6 only presents the optimal results. We denote this method as ***Few-shot***. In alignment with the Qwen1.5-14B-Chat input format, we adhere to a structure comprising two message roles: "system" and "user." The "system" role is designated for task descriptions, while the "user" role is intended for text input. Additionally, we selected the sample with the most similar fact description to the given factual description as the input demonstrations. The specific prompt demonstration for Legal Judgment Prediction are shown in Table 7.

4.3 Main Experimental Results

The main results of the three subtasks on CAIL2018-small are summarized in Table 6. Our Prompt4LJP method, whether using discrete or continuous templates, outperforms baseline methods, particularly in charges and terms of penalty prediction. Compared to the best baseline model EPM, Prompt4LJP shows F1-score improvements of 2.25% and 4.76% (charges and penalties) with discrete templates, and 3.24% and 4.05% with the continuous template, demonstrating prompt-learning's ability to leverage pre-trained knowledge and adapt flexibly to specific tasks.

However, our performance in law article prediction is inferior to EPM, likely because the $< law >$ slot is filled with simple numbers (e.g., "256"), while the $< crime >$ and $< term >$ slots contain contextually rich phrases like "robbery"

Table 5 Hyper-parameter settings.

Hyper-parameter	Value
per-device-train-batch-size	4
gradient-accumulation-steps	2
learning-rate	5e-6
num-train-epochs	3.0
lr-scheduler-type	cosine
warmup-steps	0.1
bf16	true

Table 6 Experimental results on CAIL2018-small dataset. Results marked with ∗ represent those from models reported in [2], which share the same data preprocessing pipeline with our approach. All other results were obtained from our own experiments.

Tasks	Law Articles				Charges				Terms of Penalty			
Metrics	Acc.	MP	MR	F1	Acc.	MP	MR	F1	Acc.	MP	MR	F1
ICL Method												
Few-shot	64.01	59.81	53.65	51.94	65.06	69.70	60.52	59.72	08.13	17.84	13.37	07.68
PEFT Method												
PEFT-Qwen1.5	50.85	42.03	36.52	36.96	57.25	50.49	46.59	45.82	23.11	28.20	18.27	18.48
SOTA Neural Methods												
MLAC∗	73.02	69.27	66.14	64.23	74.73	72.65	69.56	68.36	36.45	34.50	29.95	29.64
TOPJUDGE∗	78.60	76.59	74.84	73.72	81.17	81.87	80.57	79.96	35.70	32.81	31.03	31.49
MPBFN∗	76.83	74.57	71.45	70.57	80.17	78.88	75.65	75.68	36.18	33.67	30.08	29.43
LADAN∗	78.70	74.95	75.61	73.83	82.86	81.69	80.40	80.05	36.14	31.85	29.67	29.28
NeuralJudge∗	79.02	75.69	75.23	74.87	81.22	77.51	78.17	77.99	36.84	34.80	32.22	32.48
EPM∗	**84.65**	**80.82**	77.55	**78.10**	84.10	84.55	80.22	81.43	36.69	35.60	32.70	32.99
CL4LJP	77.01	75.42	73.38	72.48	79.14	78.45	78.11	77.25	36.31	33.20	30.05	29.53
Prompt4LJP(ours)												
Discrete	79.24	78.28	77.27	76.25	84.66	85.19	84.12	83.68	**40.44**	39.40	**37.02**	**37.75**
Continuous	80.95	80.08	**78.42**	77.49	**86.01**	**85.82**	**84.68**	**84.67**	40.41	**40.92**	34.86	37.04

and "imprisonment for less than one year," providing more linguistic and semantic information to the model. Future research will focus on handling this issue.

Continuous templates generally outperform discrete ones in law articles and charges prediction, with accuracy and F1-score differences exceeding 1%, due to the flexibility of learnable pseudo tokens in continuous prompts. However, for penalty prediction, continuous templates slightly underperform discrete ones, with gaps averaging less than 0.4%, possibly because penalty prediction benefits from the stability of discrete templates.

Overall, both direct fine-tuning and prompting of LLMs perform poorly on LJP tasks, indicating a failure to fully utilize the encyclopedic linguistic evidence embedded in the pre-training process. In contrast, our Prompt4LJP framework, with its dual-slot prompt template, effectively harnesses legal knowledge within the pre-trained LMs, significantly improving LJP performance by better capturing domain-specific information and enhancing the model's capability in complex legal reasoning, ultimately surpassing current SOTA methods in accuracy and reliability.

Additionally, to validate the generalization capability of our method across different datasets, we conducted experiments on the LAIC2021 dataset. ML-LJP [22] represents the latest SOTA method on the LAIC2021 dataset. Consequently, we adopted its data processing pipelines and used ML-LJP as our primary baseline. Given that ML-LJP addresses law articles prediction as a multi-label classification task, whereas our study focuses on single-label scenarios, we limit our evaluation to the charges prediction and terms of penalty prediction. The specific experimental results are shown in Table 8.

We observed that for charges prediction, our method's performance is slightly inferior to ML-LJP. However, compared to other baseline models, our methods still demonstrates excellent performance. For instance, our method's F1-score surpasses that of the second-best baseline EPM, in both discrete and continuous templates. Additionally, in the discrete template, our method's accuracy also exceeds that of EPM. Further observation of terms of penalty prediction results reveals that our method exhibits superior performance. In the continuous template, the F1-score of our method is 1.29% higher than that of ML-LJP.

Overall, our method demonstrates superior performance on both the CAIL2018-small dataset and the newly released LAIC2021 dataset, particularly in the terms of penalty prediction subtask. By employing the dual-slot template and relevant scoring mechanism, our method effectively leverages existing legal knowledge in MLs and exhibits excellent generalization capabilities.

4.4 Hyperparameter Analysis

The number of pseudo tokens within continuous templates and the number of negative prompts for training represent our core hyperparameters. Experimental findings reveal that different parameter configurations exert distinct impacts across LJP subtasks.

4.4.1 The Influence of the Quantity of Pseudo Tokens.

From Table 1, we note that the three independent subtasks of LJP share the same continuous template pattern with $[P_{1:l_1}]$, $[Q_{1:l_2}]$, and $[M_{1:l_3}]$. To explore the impact of varying the number of pseudo tokens (i.e., l_1, l_2, l_3) on prediction performance, we

Table 7 Demonstration of prompt for LJP: Here, "$< fact >$" represents the factual statement of a legal case, while "[charge]", "[law]", and "[penalty]" denote the charge, law, and penalty term, respectively.

Role	**Message**
system	You will participate in a legal judgment prediction task. Given a description of the facts of a case, you need to predict the applicable law, the charge, and the possible term of penalty. Each charge, term of penalty and applicable law should be singular. Based on the provided case facts, use your legal knowledge and logical reasoning to make predictions. Output format requirements: output the predicted charge, law, and term of penalty separated by commas, without including any additional analysis or explanation. For example: theft,264,up to three years imprisonment.
user	**Example 1** input: $< fact1 >$ output: [charge 1],[law 1],[penalty 1] **Example 2** input: $< fact2 >$ output: [charge 2],[law 2],[penalty 2] **...** input: $< fact >$ output:

Table 8 Experimental results on LAIC2021 dataset. Text in **bold** denotes the best result, while underline indicates the second best result across the entire table. Results marked with represent those from ML-LJP, which share the same data preprocessing pipeline with our approach.

Tasks	Charges				Terms of Penalty			
Metrics	Acc.	MP	MR	F1	Acc.	MP	MR	F1
SOTA Neural Methods								
TOPJUDGE*	96.46	91.97	90.17	91.06	38.97	38.90	35.16	36.94
LADAN*	96.21	91.20	91.51	91.35	41.28	40.55	38.73	39.62
NeuralJudge*	94.21	88.43	84.73	86.54	37.12	38.16	34.07	36.00
EPM*	97.11	93.89	92.63	93.11	40.48	38.99	38.34	38.38
ML-LJP*	**97.54**	**95.56**	**93.73**	**94.64**	47.63	48.39	**46.68**	47.52
Prompt4LJP(ours)								
Discrete*	97.23	94.57	92.11	93.15	47.02	48.76	45.73	**48.81**
Continuous*	97.05	95.38	92.15	93.36	**47.75**	**50.26**	46.06	47.79

conduct three experiments for each subtask. We adopt a basic hyperparameter tuning strategy, setting l_1, l_2, l_3 to the same value, denoted as $l = l_1 = l_2 = l_3$. Given that in Chinese expressions, a complete word typically consists of two Chinese characters, we vary l in $\{0, 2, 6, 10, 16\}$, increasing in multiples of 2.

Figure 3 demonstrates varying optimal l values across subtasks according to F1-score, solely based on differences in label values and quantities within the same continuous template pattern. Additionally, for terms of penalty prediction at $l = 10$ and $l = 16$, F1-score increases while accuracy decreases, suggesting the introduction of ambiguities due to the absence of pre-training knowledge in randomly initialized pseudo tokens.

4.4.2 The Effect of Number of Negative Prompts.

In our experiments, we noticed a significant impact on prediction performance based on the number of negative prompts (n) used during training. We varied n from 1 to 10 and focused on predicting law articles and terms of penalty. Our experiments solely employed the discrete template and analyzed F1-score, considering the imbalanced data distribution in the CAIL2018 dataset.

Results, as depicted in Figure 4, illustrate that increasing n enhances the model's ability to predict low-frequency labels, particularly evident in law articles prediction with its larger label set (101 labels). This improvement is attributed to the model's capacity to learn relevant characteristics between the fact and the target label, alongside irrelevant characteristics between the fact and non-target labels. Consequently, the model captures more discriminative information, benefiting overall prediction performance. Conversely, in terms of penalty prediction, which involves a smaller label set (11 labels), setting n to its maximum value (i.e., $n = 10$) leads to overfitting, thereby diminishing prediction performance.

4.4.3 The Impact of Different Discrete Prompt Templates

In the prompt learning paradigm, using different templates can impact task performance. To investigate this, we constructed additional two prompt discrete templates

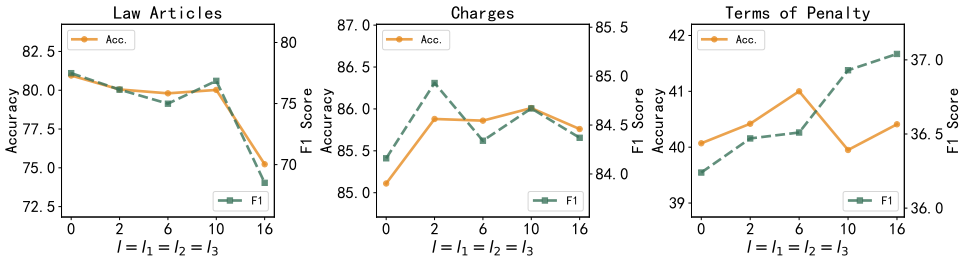

Fig. 3 Impact of the number of pseudo tokens in the continuous templates for three subtasks.

for each of the three subtasks. The difference between the two discrete templates and the one used in our original work lies in the answer words. Besides "yes" and "no" (referred to as Prompt-1), we also chose "right" and "wrong" (referred to as Prompt-2), and "have" and "not have" (referred to as Prompt-3). For each pair of answer words, we constructed corresponding discrete prompt templates that fit the context. The specific experimental results on CAIL2018-small are shown in Table 9.

We observed that for charge prediction and term of penalty prediction, the choice of answer words had minimal impact on the experimental results. The differences in Acc. and F1-score between the different templates were all below 1%. Our main experiments demonstrated that the proposed method outperformed SOTA approaches in both tasks, indicating that our Prompt4LJP effectively extracts features that distinguish charge and term of penalty labels, reducing the sensitivity of performance to prompt templates.

In contrast, for the law article prediction, the choice of answer words had a more significant effect on results. The best-performing template was not Prompt-1 but rather Prompt-3, which uses "have" and "not have" as answer words. This template led to improvements of 2.02% in accuracy and 1.11% in F1-score compared to the original template. We hypothesize that this difference arises due to the numerical nature of law article labels, which lack semantic content, making it harder for the model to extract contextual cues. Therefore, in the law article prediction, the prompt template plays a more crucial role in providing the contextual information needed by the model, significantly influencing performance.

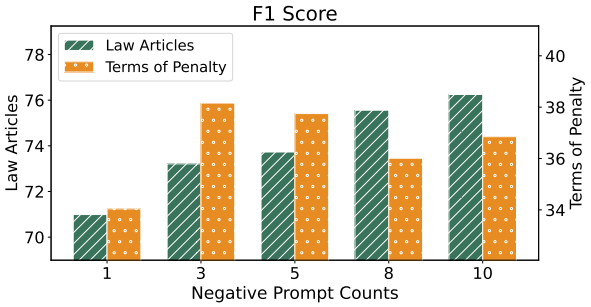

Fig. 4 Impact of number of negative prompts for model training in the discrete templates for the prediction of law articles and terms of penalty.

Table 9 Results of different prompt templates on the CAIL2018-small dataset.

Tasks	Law Articles				Charges				Terms of Penalty			
Metrics	Acc.	MP	MR	F1	Acc.	MP	MR	F1	Acc.	MP	MR	F1
Prompt-1	77.62	76.98	75.15	73.73	83.70	84.55	83.60	82.89	40.44	39.40	37.02	37.75
Prompt-2	79.93	76.49	75.21	74.04	83.95	85.11	82.70	82.53	39.90	40.15	35.56	37.25
Prompt-3	79.64	77.69	76.31	74.84	83.73	85.71	83.18	83.19	40.31	42.29	32.93	36.13

4.5 Impact of Training Dataset Size

Several studies have highlighted the efficacy of prompt learning with smaller datasets in various NLP tasks [7, 8, 11]. Our study examines the Prompt4LJP model's impact on prediction performance across three subtasks with limited data, using both discrete and continuous prompt templates.

We trained the three subtasks using 10%, 30%, 50%, 70%, and 90% of the available data. The results, as shown in Table 10, demonstrate that both the continuous and discrete templates exhibit similar trends. Specifically, under the continuous prompt template, significant improvements in accuracy and F1-score from 10% to 30% of the training data for charges and law articles. However, gains diminish from 50% to 100%, indicating robust generalization capabilities by harnessing the knowledge embedded in pre-trained MLs to support the LJP task, irrespective of training data quantity. Furthermore, our approach surpasses the top-performing baseline EPM, trained on the entire dataset, utilizing only 50% of the training data for predicting penalty terms and 70% for predicting charges. These findings underscore the advanced capabilities of prompt learning methods in managing situations with limited training data, showcasing their superiority compared to shallow neural networks.

Table 10 Experimental results with fewer training data under continuous and discrete prompt templates. The data in **bold** indicates that our method outperformed the best baseline model EPM when using less than 70% of the training data.

Tasks	Law Articles		Charges		Terms of Penalty	
Metrics	Acc.	F1	Acc.	F1	Acc.	F1
Best Baseline						
EPM	84.65	78.10	84.10	81.43	36.69	32.99
Continuous Template						
Train-10%	73.85	64.52	75.86	74.34	27.27	23.79
Train-30%	78.01 (+4.16)	71.87 (+7.35)	81.48 (+5.62)	80.66 (+6.32)	29.28 (+2.01)	29.35 (+5.56)
Train-50%	78.77 (+0.76)	72.45 (+0.58)	83.34 (+1.86)	**81.70** (+1.04)	**40.83** (+11.55)	**35.12** (+5.77)
Train-70%	79.06 (+0.29)	74.62 (+2.17)	**85.41** (+2.07)	**84.24** (+2.54)	**38.93** (-1.90)	**36.80** (+1.68)
Train-90%	79.98 (+0.92)	76.09 (+1.47)	84.99 (-0.42)	84.11 (-0.13)	41.52 (+2.59)	36.80 (+0.00)
Train-Full	80.95 (+0.97)	77.49 (+1.40)	86.01 (+1.02)	84.67 (+0.56)	40.41 (-1.11)	37.04 (+0.24)
Discrete Template						
Train-10%	71.35	62.67	75.39	73.16	27.33	23.38
Train-30%	77.93 (+6.58)	73.05 (+10.38)	79.08 (+3.69)	79.56 (+6.40)	27.38 (+0.05)	28.49 (+5.11)
Train-50%	78.74 (+0.81)	73.68 (+0.63)	83.64 (+4.56)	**82.49** (+2.93)	**40.43** (+13.05)	**35.46** (+7.08)
Train-70%	78.13 (-0.61)	74.74 (+1.06)	**84.30** (+0.66)	**83.18** (+0.69)	34.61 (-5.82)	**33.01** (-2.45)
Train-90%	79.85 (+1.72)	75.95 (+1.21)	85.13 (+0.83)	84.06 (+0.88)	40.71 (+6.10)	37.00 (+3.99)
Train-Full	79.24 (-0.61)	76.25 (+0.30)	84.66 (-0.47)	83.68 (-0.38)	40.44 (-0.27)	37.75 (+0.75)

4.6 Error Analysis

In this section, for each subtask, we randomly selected 400 samples with incorrect predictions and analyzed the true labels, all candidate labels, and the correlation scores for each candidate label for each sample. Below are the error analysis results for each subtask.

4.6.1 Charge Prediction Issues

The charge prediction faces some key challenges. First, the model often assigns highly correlation scores to confusable charges (e.g., "fraud" vs. "contract fraud" and "arson" vs. "manslaughter"), leading to frequent misidentifications. For instance, in cases involving economic crimes like "embezzlement of funds" vs. "misappropriation of public funds," the model assigns a higher score to the incorrect charge ("misappropriation of public funds" at 0.9965) instead of the true charge ("embezzlement of funds" at 0.8842), indicating limited ability to distinguish between similar charges related to official duties. Second, the model struggles with low-frequency tail charges due to insufficient training data, leading to poor performance and misclassification. To address these issues, future work will focus on refining scoring mechanisms using semantic similarity, charge-specific feature matching, and contextual indicators to improve correlation scores.

4.6.2 Law Article Prediction Issues

Similar to charge prediction, law article prediction faces difficulties in distinguishing between confusable law articles and performs poorly on low-frequency tail law articles. First, correlation scores between candidate law articles are often very close, leading to inaccurate selections. For example, in theft-related cases, the model might incorrectly associate the true law article "264" (theft) with "266" (fraud) or "133" (intentional injury), both of which have correlation scores close to or higher than the true law article. This reflects the model's inability to differentiate law articles with similar behavioral descriptions but different legal applicability. Second, the model struggles to accurately parse behavioral characteristics from case descriptions. For instance, when a suspect steals a bicycle by breaking a lock, the correct law article is "264" (theft), but the model often associates it with "246" (robbery) or "312" (concealment of crime proceeds), highlighting difficulties in distinguishing between law articles. Law articles are numerically represented and lack sufficient contextual information. Future research could focus on integrating semantic information, such as linking law articles to legal terms, case law, and definitions, using legal knowledge graphs.

4.6.3 Term of Penalty Prediction Issues

SOTA methods currently demonstrate relatively low performance in term of penalty prediction tasks. Although our Prompt4LJP method improves accuracy, performance remains below optimal levels. Analysis reveals that, unlike charge and law article prediction, many predicted correlation scores are below 0.7, and in some cases, the model fails to generate accurate candidate term of penalty labels, indicating significant

difficulties in extracting discriminative features related to term of penalty label. To ensure consistency with benchmark evaluations, we adopted a common data processing pipeline, including the standardization of term of penalty labels. Currently, term of penalty labels are categorized into mutually exclusive intervals, such as "7-10 years of fixed-term imprisonment." While this simplifies the term of penalty prediction, it introduces some limitations. On the one hand, the granularity of these intervals may hinder the model's ability to capture nuanced relationships between case details and specific term of penalty labels. On the other hand, some cases may fall into intervals that are not covered, leading to inaccuracies and reduced prediction performance. To address these issues, future research could explore more fine-grained interval definitions or adopt continuous term of penalty predictions that better reflect case-specific characteristics.

4.7 Computational Efficiency Analysis

In this section, we analyze the training efficiency of the bert-base-chinese model [23] on the CAIL2018-small dataset [14], using the charge prediction subtask as an example. The dataset contains 96,540 samples and covers 117 charge labels. For prompt construction, in addition to positive prompts for each label, we select $n = 10$ negative prompts, resulting in a total of:

$$96,540 \times (1 + 10) = 1,061,940 \,\text{prompts}.$$

The experiment was conducted on 4 NVIDIA GeForce RTX 3090 GPUs, each with a batch size of 16, leading to a total batch size of 64. The model was trained for 8 epochs, each consisting of 16,593 iterations, totaling 132,744 iterations. A weight decay strategy was applied with a decay coefficient of 1×10^{-3}.

Each epoch took about 2.5 hours, with a total training time of 20 hours. Therefore, the fine-tuning time per prompt is calculated as:

$$\frac{20\,\text{hours} \times 60\,\text{minutes/hour} \times 60\,\text{seconds/minute}}{1,061,940\,\text{prompts}} \approx 0.068\,\text{seconds}\,(68\ \text{milliseconds}).$$

This approach increases the number of training samples by a factor of n, leading to additional hardware and time overhead. As discussed in Section 4.4.2, we randomly select n labels to construct negative prompts. In practice, the choice of n can significantly impact the experimental results. Furthermore, as shown in Section 4.5, reducing the training set size to 70% of the original still yields good performance in both charge and term of penalty prediction tasks. In future work, we plan to explore more effective strategies for selecting negative labels, such as choosing labels that are easily confused with the actual labels, to address the issue of label ambiguity. By optimizing the selection of negative prompts and reducing n , we can enhance the model's ability to capture discriminative information. Additionally, we aim to reduce the dataset size to alleviate the training burden while maintaining or improving model performance.

5 Conclusion and Future Work

In this paper, we introduce Prompt4LJP, a novel framework that leverages the prompt learning paradigm to tackle the complex task of LJP. The key challenge addressed is the dynamic word lengths of LJP labels, which complicate the direct application of traditional prompt learning methods. Our framework employs a dual-slot prompt structure, combining factual descriptions and labels, along with a correlation scoring mechanism to effectively handle varying label lengths. Extensive experiments conducted on the CAIL2018-small dataset [14] demonstrate that Prompt4LJP significantly outperforms state-of-the-art methods. Specifically, we observe F1-score improvements of 2.25% and 4.76% in charge prediction, and 3.24% and 4.05% in term of penalty prediction compared to the best baseline model, EPM. These findings highlight the capability of our approach to leverage pre-trained LMs and adapt effectively to tasks involving dynamic word lengths.

In our efforts to integrate the prompt learning paradigm into LJP, we recognize several key limitations, including insufficient model interpretability, suboptimal performance in law article prediction, and a lack of consideration for potential interdependencies among the three sub-tasks. To address these issues, we propose the following strategies. First, we will employ attention mechanism visualization or feature importance analysis to enhance model transparency and improve interpretability. Second, to improve law article prediction, we will explore richer representations by associating legal provisions with relevant legal terms, cases, and semantic contexts to provide deeper semantic understanding. Additionally, we will utilize external legal knowledge bases, leveraging existing legal documents and case information to further enrich the model's knowledge base. Finally, we will investigate the application of joint prompting techniques, exploring how shared features and representations can be leveraged to optimize performance across multiple sub-tasks.

6 Ethical Statement

LJP is a research domain of significant social relevance but also considerable ethical sensitivity. Therefore, it is crucial to discuss the ethical implications of our work. The proposed model, Prompt4LJP, is designed to enhance the accuracy of LJP tasks and outperform current SOTA methods. To some extent, our approach can provide reference suggestions to legal practitioners, such as judges, to reduce their workload and improve efficiency. However, it is important to emphasize that this model is not intended to replace judicial decision-making but rather to serve as an auxiliary tool. Judicial practitioners remain the primary decision-makers and retain full responsibility for final verdicts, ensuring fairness and justice in the legal process. Moreover, our research relies on publicly available legal case datasets, including CAIL2018 [14] and LAIC2021[1], which have been anonymized by their original publishers to protect personal privacy (e.g., names and contact details). Nonetheless, we acknowledge that, despite anonymization, inherent biases in the training data may influence the model's fairness. Such biases could lead to unjust outcomes for certain groups, an issue that requires continuous attention and iterative improvements.

Although our research introduces methodological innovations, the model has several limitations, including susceptibility to data bias, lack of interactivity, and insufficient interpretability of its predictions. These challenges limit its practical applicability and raise concerns about potential biases or unfair results in certain scenarios. Given the high-stakes nature of the legal domain, we explicitly state that this model is intended solely for academic research purposes and will not be used in real-world legal decision-making scenarios.

References

[1] Luo, B., Feng, Y., Xu, J., Zhang, X., Zhao, D.: Learning to predict charges for criminal cases with legal basis. arXiv preprint arXiv:1707.09168 (2017)

[2] Feng, Y., Li, C., Ng, V.: Legal judgment prediction via event extraction with constraints. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 648–664 (2022)

[3] Zhang, H., Dou, Z., Zhu, Y., Wen, J.-R.: Contrastive learning for legal judgment prediction. ACM Transactions on Information Systems **41**(4), 1–25 (2023)

[4] Fei, Z., Shen, X., Zhu, D., Zhou, F., Han, Z., Zhang, S., Chen, K., Shen, Z., Ge, J.: Lawbench: Benchmarking legal knowledge of large language models. arXiv preprint arXiv:2309.16289 (2023)

[5] Schick, T., Schütze, H.: Exploiting cloze questions for few shot text classification and natural language inference. arXiv preprint arXiv:2001.07676 (2020)

[6] Zhu, Y., Wang, Y., Qiang, J., Wu, X.: Prompt-learning for short text classification. IEEE Transactions on Knowledge and Data Engineering (2023)

[7] Wang, C., Wang, J., Qiu, M., Huang, J., Gao, M.: Transprompt: Towards an automatic transferable prompting framework for few-shot text classification. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 2792–2802 (2021)

[8] Zhang, Z., Wang, B.: Prompt learning for news recommendation. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 227–237 (2023)

[9] Ding, N., Chen, Y., Han, X., Xu, G., Xie, P., Zheng, H.-T., Liu, Z., Li, J., Kim, H.-G.: Prompt-learning for fine-grained entity typing. arXiv preprint arXiv:2108.10604 (2021)

[10] Zhu, T., Qin, Y., Chen, Q., Hu, B., Xiang, Y.: Enhancing entity representations with prompt learning for biomedical entity linking. In: IJCAI, pp. 4036–4042 (2022)

[11] Xiang, W., Wang, Z., Dai, L., Wang, B.: Connprompt: Connective-cloze prompt learning for implicit discourse relation recognition. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 902–911 (2022)

[12] Sahoo, P., Singh, A.K., Saha, S., Jain, V., Mondal, S., Chadha, A.: A systematic survey of prompt engineering in large language models: Techniques and applications. arXiv preprint arXiv:2402.07927 (2024)

[13] Sabbatella, A., Ponti, A., Giordani, I., Candelieri, A., Archetti, F.: Prompt

optimization in large language models. Mathematics **12**(6), 929 (2024)

[14] Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., Feng, Y., Han, X., Hu, Z., Wang, H., et al.: Cail2018: A large-scale legal dataset for judgment prediction. arXiv preprint arXiv:1807.02478 (2018)

[15] Kort, F.: Predicting supreme court decisions mathematically: A quantitative analysis of the "right to counsel" cases. American Political Science Review **51**(1), 1–12 (1957)

[16] Segal, J.A.: Predicting supreme court cases probabilistically: The search and seizure cases, 1962-1981. American Political Science Review **78**(4), 891–900 (1984)

[17] Ulmer, S.S.: Quantitative analysis of judicial processes: Some practical and theoretical applications. Law and Contemporary Problems **28**(1), 164–184 (1963)

[18] Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., Sun, M.: Legal judgment prediction via topological learning. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3540–3549 (2018)

[19] Yang, W., Jia, W., Zhou, X., Luo, Y.: Legal judgment prediction via multi-perspective bi-feedback network. arXiv preprint arXiv:1905.03969 (2019)

[20] Xu, N., Wang, P., Chen, L., Pan, L., Wang, X., Zhao, J.: Distinguish confusing law articles for legal judgment prediction. arXiv preprint arXiv:2004.02557 (2020)

[21] Yue, L., Liu, Q., Jin, B., Wu, H., Zhang, K., An, Y., Cheng, M., Yin, B., Wu, D.: Neurjudge: A circumstance-aware neural framework for legal judgment prediction. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 973–982 (2021)

[22] Liu, Y., Wu, Y., Zhang, Y., Sun, C., Lu, W., Wu, F., Kuang, K.: Ml-ljp: Multi-law aware legal judgment prediction. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1023–1034 (2023)

[23] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[24] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., *et al.*: Language models are few-shot learners. In: Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901 (2020)

[25] Sun, J., Huang, S., Wei, C.: Chinese legal judgment prediction via knowledgeable

prompt learning. Expert Systems with Applications **238**, 122177 (2024)

[26] Rubinstein, R.Y., Kroese, D.P.: The Cross-entropy Method: a Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning vol. 133. Springer, ??? (2004)

[27] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)

[28] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., *et al.*: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45 (2020)

[29] Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., Hui, B., Ji, L., Li, M., Lin, J., Lin, R., Liu, D., Liu, G., Lu, C., Lu, K., Ma, J., Men, R., Ren, X., Ren, X., Tan, C., Tan, S., Tu, J., Wang, P., Wang, S., Wang, W., Wu, S., Xu, B., Xu, J., Yang, A., Yang, H., Yang, J., Yang, S., Yao, Y., Yu, B., Yuan, H., Yuan, Z., Zhang, J., Zhang, X., Zhang, Y., Zhang, Z., Zhou, C., Zhou, J., Zhou, X., Zhu, T.: Qwen technical report. arXiv preprint arXiv:2309.16609 (2023)

[30] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)

[31] Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., Ma, Y.: Llamafactory: Unified efficient fine-tuning of 100+ language models. arXiv preprint arXiv:2403.13372 (2024)