RLJP: Legal Judgment Prediction via First-Order Logic Rule-enhanced with Large Language Models

Yue Zhang[1], Zhiliang Tian[1], Shicheng Zhou[2], Haiyang Wang[1],
Wenqing Hou[1], Yuying Liu[1], Xuechen Zhao[3], Minlie Huang [4], Ye Wang[1],
Bin Zhou[1]

[1] College of Computer Science and Technology, National University of Defense and Technology.

No.137 Yanwachi Street, Changsha, Hunan, 410073, P R.China,

[2] College of Electronic Engineering, National University of Defense Technology.

No. 460, Huangshan Road, Shushan District, Hefei. 230037, P. R. China,

[3] School of Data and Computer Science, Shandong Women's University. No.2399 Daxue Road, Ji'nan, Shandong, 250300, P R.China

[4] Institute for Artificial Intelligence, Tsinghua University.

Dept. of Computer Science, Tsinghua University, Beijing 100084, China

Correspondence: binzhou @nudt.edu.cn

Abstract

Legal Judgment Prediction (LJP) is a pivotal task in legal AI. Existing semantic-enhanced LJP models integrate judicial precedents and legal knowledge for high performance. But they neglect legal reasoning logic, a critical component of legal judgments requiring rigorous logical analysis. Although some approaches utilize legal reasoning logic for high-quality predictions, their logic rigidity hinders adaptation to case-specific logical frameworks, particularly in complex cases that are lengthy and detailed. This paper proposes a rule-enhanced legal judgment prediction framework based on first-order logic (FOL) formalism and comparative learning (CL) to develop an adaptive adjustment mechanism for legal judgment logic and further enhance performance in LJP. Inspired by the process of human exam preparation, our method follows a three-stage approach: first, we initialize judgment rules using the FOL formalism to capture complex reasoning logic accurately; next, we propose a Confusion-aware Contrastive Learning (CACL) to dynamically optimize the judgment rules through a quiz consisting of confusable cases; finally, we utilize the optimized judgment rules to predict legal judgments. Experimental results on two public datasets show superior performance across all metrics. The code is publicly available[1].

1 Introduction

The application of AI in law is growing annually, demonstrating capabilities in both assisting judicial professionals and providing legal consultation services to the public. Legal Judgment Prediction (LJP) aims to predict the outcome of a case based on its facts, typically including the applicable legal provisions, accusation, and prison terms. As legal judgments are highly professionalized, legal knowledge is essential for LJP predictions. Researchers often employ deep learning models enhanced with legal knowledge or judicial precedents to achieve LJP.

The current methods of LJP mainly consist of two approaches: modeling legal knowledge and retrieving judicial precedents or legal knowledge. (1) Methods modeling legal knowledge typically use neural networks to model legal provisions and extract semantic associations between case facts and legal knowledge(Xu et al., 2020, 2024). While these models can efficiently extract relevant legal knowledge for LJP relying on semantic similarity, they overlook the intrinsic logic of cases. (2) Other methods leveraged semantic techniques to search for related judicial precedents or legal knowledge(Wu et al., 2023a; He et al., 2024). Judicial precedents and legal knowledge as references to enhance the model's understanding of the case circumstances, but they tend to study similar cases or knowledge instead of modeling the logical reasoning process. In summary, the above two types of methods rely on textual and semantic matching but ignore capturing the logic of judging legal cases, where logic in reasoning is crucial in legal judgment.

To address the aforementioned issues, researchers applied legal judgment logic to enhance reasoning in traditional deep learning models or large language models (LLMs) for LJP. Some researchers employ deep learning models to extract logical premises from complex cases (Yue et al., 2021, 2024) or handle the legal judgment using

---

[1] https://anonymous.4open.science/r/RLJP-FDF1

rules defined by experts (Gan et al., 2021). The integration of legal judgment logic has improved the reasoning capabilities of deep learning models. However, the ability of deep learning models is still limited by their model capacity and training data scale. As LLMs have remarkable abilities in text comprehension and logical reasoning, researchers combined judgment logic with LLMs for LJP. Researchers(Deng et al., 2023) use LLMs to retrieve relevant legal provisions and extract four types of criminal elements from case facts for judgment based on the classical legal logic syllogism. Prompts constructed by keyword structure from legal experts' determination guide LLMs to classify legal cases(Izzidien et al., 2024). In summary, while existing methods using legal logic effectively analyze cases through established rules, their logic rigidity limits adaptation to case-specific contexts in complex scenarios.

In this paper, we propose a dynamically optimized judgment rule method with adaptive adjustment mechanisms, formalizing rules(§3.3) through First-Order Logic (FOL) - a symbolic language for complex legal reasoning. Our framework integrates tree-splitting operations with contrastive learning, where confusable cases systematically generate correct/erroneous judgment pairs for Confusion-Aware Contrastive Learning (CACL) (§3.4). Iterative optimization produces enhanced rules with superior discriminative power in complex cases, enabling LLMs to achieve precise judgment predictions. This culminates in **RLJP** (**R**ule-enhanced **L**egal **J**udgment **P**rediction), a logic-enhanced system operationalizing FOL rules for legal judgment prediction (§3.5).

Our contributions are: (1) We propose a dynamic rule optimization method. Pioneer the modeling of judgment rules optimization as tree splitting, and use the CACL mechanism for self-adaptive rules, overcoming the limitations of fixed rules in handling complex cases; (2) We propose RLJP, a method that novelly integrates FOL judgment rules to enhance reasoning ability for LJP; (3) Comprehensive evaluations on two public datasets demonstrate state-of-the-art performance across all metrics compared to baseline methods. To facilitate future research, we make our code publicly available.

2 Related Work

LJP aims to predict judicial outcomes by employing traditional models or LLMs to analyze legal cases (He et al., 2023; Deng et al., 2024; Zhang et al., 2024). Broadly, LJP can be divided into two main approaches: one based on semantic similarity (Liu et al., 2024a) and the other guided by judgment logic (Cheng et al., 2024).

2.1 LJP Assisted by Semantic Similarity

The semantic similarity-assisted method for LJP involves modeling domain knowledge and applying retrieval techniques (Zhang and Dou, 2023).

The former mainly utilizes neural networks to model legal knowledge (Yue et al., 2021; Yao et al., 2024) for feature extraction and to establish semantic associations among the knowledge. To improve the accuracy of crime prediction, the prevailing methods primarily focus on refining the attention mechanism (He et al., 2023; Yu and Qiu, 2023; He et al., 2025). Additionally, some methods infuse domain-specific legal knowledge into LLMs (Wang et al., 2024; Li et al., 2025), thereby enhancing the models' capability to identify and leverage critical information. Regarding confusion issue in criminal charges, Zhang et al. (2023) and Gan et al. (2023) utilized contrastive learning to capture fine-grained distinctions. Using retrieval techniques refers to utilizing search engines (Wu et al., 2023a; Yue et al., 2023; Liu et al., 2024b) to obtain external knowledge from semantically similar judicial precedents. He et al. (2024) introduced the retrieval technology into the SimCourt multi-agent framework. Qin et al. (2024) constructed a hierarchical semantic ID for the retrieved candidate documents.

Legal judgment fundamentally relies on rigorous, logic-bound reasoning demanding systematic analysis. This process necessitates precise legal interpretation, detailed case fact scrutiny, and accurate application of legal principles, whereas solely relying on textual semantic similarity fails to ensure the accuracy and logical rigor required in LJP.

2.2 LJP Assisted by Case Judgment Logic

The legal judgment logic-enhanced method for LJP leverages models to summarize the knowledge embedded in case details (Chen et al., 2020; Yang et al., 2022) and extract the underlying reasoning logic.

Existing methods focus on using the entire factual description (Chai et al., 2020; Hong and Chang,

2023) to generate judgment results, overlooking the actual judicial scenario where judges consider various criminal circumstances (Wu et al., 2022) to determine the verdict and sentencing. Consequently, Yue et al. (2021) proposed a legal judgment framework, NeurJudge, which can divide facts and the prediction results of intermediate subtasks into different scenarios. Yue et al. (2024) further proposed NeurJudge+, which integrates the semantic labels of accusations and legal provisions into facts. Gan et al. (2021) injected legal knowledge as an additional logical rule into the neural network. Deng et al. use LLMs to retrieve relevant legal provisions and extract four types of criminal elements from case facts for judgment based on the classical legal syllogism. Prompts constructed by keyword structure from legal experts' determination guide LLMs to classify legal cases(Izzidien et al., 2024).

In summary, the inherent structural rigidity of previous methods limits their ability to dynamically adapt to diverse, case-specific logical frameworks. Thus, when confronted with complex legal cases, these inflexible frameworks struggle to reconcile conflicting evidence or interpretative contradictions. In light of this, this paper aims to develop an adaptive adjustment mechanism for legal judgment logic, thereby further enhancing LJP performance in complex and contradictory situations.

3 Method

3.1 Task Definition

In this paper, we focus on the problem of legal judgment prediction. First, we clarify the definition of the terms as follows:

- **Judgment** is the final decision made by a judge in a legal case based on the facts and the judgment rule. It typically consists of the law article, the charge, and the prison term. We represent the judgment of a case as $j = (article, charge, prison\_term)$, where $(article, charge, prison\_term)$ refers to the labels of provisions, accusation, and prison term, respectively.

- **Precedent** is the previous case with a similar fact. For a given judgment label, there can be several precedents, which can be denoted as $S = \{s_1, s_2, ..., s_n\}$, where n is the number of precedents.

- **Rule** refers to formalized patterns derived from summarizing recurrent case develop-

ment trajectories across similar judicial decisions in our study.

Problem 1 Induction Rule. Extract causal factors from case facts for the content of the rule, and construct the confusable case fact to optimize the rule.

Problem 2 Legal Judgment Prediction. Given the case fact f, our task is to analyze f based on $rule$, then predict judgment $j = (article, charge, prison\_term)$.

3.2 Model Overview

During the learning process of human students, they typically first acquire examples and knowledge from textbooks, then summarize problem-solving logic or methods. These methods are later tested through quizzes to assess their accuracy, and the final learning outcomes are evaluated in the final exam. Inspired by this, we propose the RLJP framework (Figure 1) to enhance reasoning ability for the LJP task. The framework contains three modules.

- **Rules Initialization Module** (§3.3). This module aims to utilize an LLM agent to generate rude judgment rules in the FOL formalism, leveraging legal provisions and precedents. This process mirrors how students acquire knowledge and summarize problem-solving logic.

- **Rules Optimization Module** (§3.4). This module aims to use the Confusable Case Set to optimize the FOL judgment rules through Confusion-Aware Contrastive Learning (CACL), which is similar to how students optimize their problem-solving logic through quiz experience.

- **Examination Module** (§3.5). This module uses an LLM agent to predict legal judgments by applying optimized judgment rules in combination with candidate labels generated by a lightweight model, which mirrors the final exam of the students.

3.3 Rules Initialization

Inspired by the cognitive process where students first encounter a specific problem type and subsequently extract solution logic, we develop reasoning rules for determining categories of legal judgments through context learning. To precisely
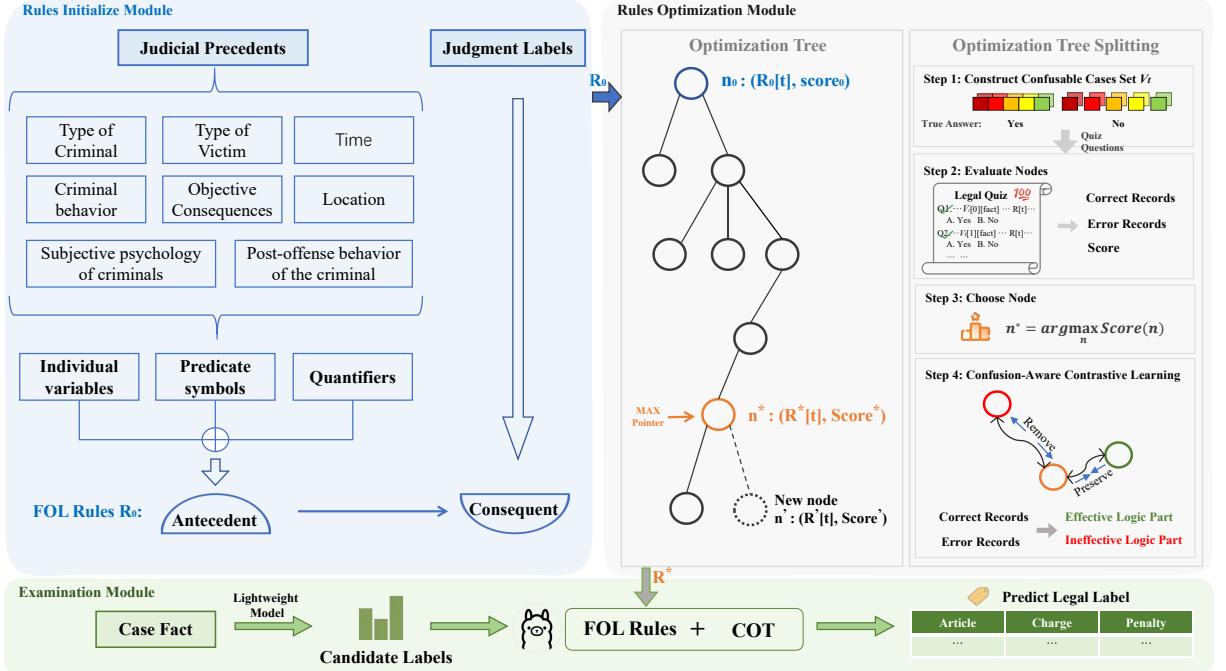
Figure 1: RLJP framework. The blue box is the Rule Initialization Module on the upper left(§3.3), the gray box is the Rule Optimization Module(§3.4) with Confusion-Aware Contrastive Learning, and the green box is the Examination Module(§3.5) for the completion of LJP tasks based on the optimized rules.

describe the reasoning rules, we utilize the FOL formalism, which can accurately express complex logic.

Each FOL judgment rule $Rule : A \to C$ comprises an antecedent A and a consequent C. The content of A is determined by the circumstance logic. And C is constructed by one or two legal judgment labels, including $article$, ($article$, $charge$), and ($article$, $prison\_term$).

Since legal provisions offer a clear legal basis for accusation determination, we select precedents violating the same legal provisions and receiving the same accusation to generate accusation judgment rules. As different legal provisions set different sentencing standards, we extract precedents violating the same legal provisions and receiving the same prison term to help generate prison-term judgment rules. This combination reminds the LLM agent that the content of judgment rules should strictly adhere to established legal provisions when initializing rules.

To systematically summarize the underlying logical patterns in case developments of similar legal cases, we employ FOL symbols to formalize legal judgment rules through a three-step process.

(1) Summarize circumstance logic. This step aims to analyze some causal factors from similar legal cases. Specifically, we use the LLM agent to summarize causal factors causing the legal judgment from some precedents in the same legal judgment, including the category of criminal subject, the category of the victim, the time and location of the crime, the criminal behavior, the objective consequences of the crime, and the criminal's subjective mental state. These factors are crucial for constructing the antecedent A of the FOL judgment rules.

(2) Define FOL symbols. The objective of this step is to establish the logical relationship between the causal factors and the legal judgment in the case. For this purpose, we use the LLM agent to define causal factors with FOL symbols, including variables, predicate symbols, and quantifiers. Variables can represent the entities, time, location, events, consequences, and mental state in factors. Predicate symbols describe specific attributes of the criminal subject, the victim, and the criminal behavior. The universal quantifier $\forall$ and the existential quantifier $\exists$, express the rule's generality or specificity.

(3) Construct FOL judgment rules. The purpose of this step is to standardize the format of rules to facilitate subsequent optimization and reasoning processes. The judgment rules are constructed using the antecedent A and the consequent C. $Rule : A \to C$ in this step. The antecedent

A comprises multiple FOL symbols and logical connectives ($\vee$, $\wedge$, and $\neg$). The consequent C is the same as LJP labels, including legal provisions $article$, (legal provisions $article$, accusation $charge$), and (legal provisions $article$, prison terms $prison\_term$).

3.4 Confusion-Aware Rule Optimize Engine

In RLJP, we propose a Confusion-Aware Rule Optimization Module to eliminate ambiguous boundaries among rules, which can learn from correct and incorrect experiences in confusable cases reasoning by iteratively optimizing rules with CACL. The full procedure of this Optimization is described in Algorithm 1. Specifically, this module works in two-stage steps: construction confusable case set(§3.4.1), and rules optimization(§3.4.2) depends on Confusion-Aware Contrastive Learning(CACL).

Algorithm 1: Confusion-Aware Rule Optimization

Require: $Precedents$: Precedent set;
Require: $target$: Target label which is the consequence of the rules;
Require: $RULE$: Initial rule set ;
Require: $DefinedScore$: A predefined score threshold serves as the stopping criterion for terminating the iterative updating process;
Require: Num: The required number of confusable cases.
1: $S_{postive} \leftarrow Precedents[target]$
2: $S_{other} \leftarrow Precedents[other]$
3: $S_{negative} \leftarrow$
$\quad RankSimilarity(S_{postive}, S_{other}, Num)$
4: Confusable case set $V_{target} \leftarrow S_{negative} \cup S_{negative}$
5: $r_0 \leftarrow RULE[target]$
6: $MaxScore \leftarrow 0, MaxPointer \leftarrow Null$
7: $R_{tree} \leftarrow [r_0]$
8: **for** r in R_{tree} **do**
9: $\quad$ **if** Is_Evaluate(r) **then**
10: $\quad\quad$ ($score$, reasoning records) $\leftarrow history(r)$
11: $\quad$ **else**
12: $\quad\quad$ ($score$, reasoning records)
$\quad\quad\quad \leftarrow Evaluate(V_{target}, r)$
13: $\quad$ **end if**
14: $\quad$ **if** $score > MaxScore$ **then**
15: $\quad\quad MaxScore \leftarrow score$
16: $\quad\quad MaxPointer \leftarrow r$
17: $\quad\quad R^* \leftarrow MaxPointer$
18: $\quad$ **end if**
19: $\quad$ **if** $MaxScore >= DefinedScore$ **then**
20: $\quad\quad$ break;
21: $\quad$ **end if**
22: $\quad R' \leftarrow CACL(R^*, $ reasoning records $)$
23: $\quad R_{tree} \leftarrow$ Add_ChildNode(R', R^*)
24: **end for**
25: **return** R^*;

3.4.1 Construction of the Confusable Cases Validation Set

The confusable case set is defined as cases characterized by highly similar factual circumstances but differing legal outcomes in our paper. In the real world, confusable cases often pose significant challenges to professionals tasked with making judgment decisions. Thus, we construct the confusable case set as a benchmark to evaluate the quality of legal rules generated in RLJP. Specifically, to measure the similarity between cases, we use the Bi-directional Generative Encoder(BGE) model to generate vector representations of case fact texts and calculate the cosine distance between vectors. The construction process of this set is as follows.

To find confusable cases for legal judgment $target$, we first find the case set, whose legal judgment label is $target$, denoted as $S_{positive}$, and the case set for other legal judgment labels, denoted as S_{other}. Then, to asses the similarity score between $S_{positive}$ and S_{other}, we convert each case fact set into fixed-dimensional embedding vectors using the BGE model: $E_t = \{E_t^1, E_t^2, ..., E_t^m\}$ for $S_{positive}$ and $E_o = \{E_o^1, E_o^2, ..., E_o^n\}$ for S_{other}, $E_t^i, E_o^i \in R^d$. Then, we compute the cosine similarity between E_t and E_o. The results are organized into a similarity matrix A, where $A(i, j) = \frac{E_t^i \cdot E_o^i}{\|E_t^i\|\|E_o^i\|}$ represents the similarity score of two cases $S_{positive}^i$ and S_{other}^j.

The higher the similarity score between cases in different judgment labels means that they contain similar crime circumstances, the more likely they are to be confusable for legal judgment. For this reason, we sort the similarity matrix A in descending order based on the similarity scores of each row, and the first column is extracted to obtain a set of negative sample cases for judgment $target$: $S_{negative} \in S_{other}$. Finally, the confusable case set V_{target} can be constructed: $V_{target} = S_{negative} \cup S_{positive}$.

3.4.2 Optimization Tree Splitting

To iteratively optimize legal judgment rules, we formulate the process of rule optimization as the process of tree-splitting. These nodes of the optimization tree are different versions rule during optimization. In one iteration, we first create some reasoning quizzes constructed by the confusable cases set, and calculate the score of the judgment rules for best-first splitting. And then, we collect correct and incorrect reasoning experience in quizzes

for CACL, which autonomously optimizes rules to keep effective logic parts and update ineffective logic parts by analyzing experience.

Optimization Tree Definition. We formalize judgment rule optimization as a weighted tree structure $T_{target} = (N, E, W)$, where N denotes rule in different versions, $E \in N * N$ encodes rule optimization relationships, and $W : N \to [0, 1]$ quantifies rules' performance using quiz score using the confusable case set. The root node $n_0 \in N$ corresponds to the initial rule formulation from §3.3, with directed edges $(n_p, n_c) \in E$ representing optimization paths where child rule n_c evolves from parent n_p through experience-based refinements. Node weights $W(n)$ establish a partial ordering over rule versions, enabling performance-guided choice in the optimization cycle.

Optimization Tree Splitting. Legal judgment rules optimization contains three iterative steps.

The first step aims to evaluate and choose the node. To select the basis node for optimization, in this step, we evaluate all rules in the current tree using the confusable case set, and choose the highest accuracy node n^* (rule R^*) to mark with the MAX pointer. We propose the CACL method to optimize R^*. The CACL method finds optimization directions based on evaluation experience and generates optimized rules. Finally, the third step aims to integrate the node. Attach the new rule as a child node to n^*, repeating these three phases until either a predefined accuracy threshold is achieved by the optimized rule or the maximum iteration count is reached. Detailed explanations of each step are as follows.

Step 1: Evaluate and Choose Node. Inspired by staged assessments in educational settings, we design a quiz using confusable cases to evaluate the rules. First, we format case facts from the V_{target} into single-choice quiz questions. The template for the construction of single-choice questions is provided in Figure 4. Then the LLM agent using R_n chooses a predicted option and generates a reasoning process for each question. The capability of R_n is assessed based on the accuracy of predicted options. The experience of choosing correctly consists of True Positives (TP) and True Negatives (TN). The experience of choosing incorrectly consists of False Positives (FP) and False Negatives (FN). The weight of node $W(n)$ is computed as formalized in Equation 1.

$$W(n) = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

The selected node $n^* = arg \max_{n \in N} W(n)$ is the node with the highest weight in the tree, corresponding to the rule R^* that achieves the highest accuracy on the validation set.

Algorithm 2: Confusion-Aware Contrastive Learning

Require: R^*: Current rule (Anchor);
Require: $Correct(R^*)$: Set of positive samples (valid reasoning records);
Require: $Incorrect(R^*)$: Set of negative samples (invalid reasoning records).
1: $\triangledown keep \leftarrow LLM(R^*, Correct(R^*))$
2: $\triangledown imp \leftarrow LLM(R^*, Incorrect(R^*))$
3: $\triangledown R \leftarrow LLM(\triangledown keep, \triangledown imp)$
4: $R' \leftarrow LLM(\triangledown R, R^*)$
5: **return** R'

Step 2: CACL Method. To optimize the current optimal rule R^*, we propose the CACL to generate an optimized new rule, which simulates students reflecting on the correct and incorrect problem-solving process. The full procedure of the CACL method is described in Algorithm 2, which consists of three core steps as follows.

Step 2-1: Construct triplets. CACL constructs the experience of rule evaluation as contrast triplets. The key triplet in CACL is (Anchor R^*, Positive Samples $Correct(R^*)$, Negative Samples $Incorrect(R^*)$). Anchor is the current rule R^*. Positive samples are correct reasoning records $Correct(R^*)$ (TP and TN) in evaluation, which contain quiz questions, reasoning process, correct option, and predicted Option. Negative samples are composed of the same quadruples and use wrong reasoning records $Incorrect(R^*)$ (FP and FN) in the evaluation. The quadruple construction is provided in Figure 2.

Step 2-2: Generate optimization direction. CACL analyzes contrast triplets for optimization, the LLM agent generates optimization direction $\triangledown R$ of R^* based on the positive and negative triplets, including effective logic parts $\triangledown keep$ and ineffective logic parts $\triangledown imp$. $\triangledown R = (\triangledown keep, \triangledown imp)$. The prompt template for generating optimization direction is shown in Figure 5.

Step 2-3: Optimize rules. The optimization direction generated from CACL guides the LLM agent to maintain the effective logic part and improve the ineffective logic part. The prompt tem-

plate for the rule optimization is detailed in Figure 3.

Step 3: Integrate Node. The optimized R' is added to the tree as a child node of R^*. We continue to optimize rules until the optimized rule achieves the predefined accuracy or the maximum iteration count is reached.

3.5 Examination

Inspired by the final examination process after many quizzes in educational settings, this module executes the LJP task with the optimized rules. In this module, we start using a lightweight BERT model to output the top 10 probable legal labels. Then we apply FOL judgment rules against each candidate label with the Chain-of-Thought method to predict legal judgment. The prompt template for completing the LJP task, as Figure 2 shows which is the same as the quiz template. If none of the candidates meet the logical constraints, we initiate a stochastic traversal of the remaining labels based on their rules. Furthermore, we activate the abstract template as Figure 6 shows to generate the abstract of the case fact when the length of the fact exceeds a defined threshold. The abstract template aims to generate condensed case abstracts that retain legally relevant features while eliminating redundant details.

4 Experiments

4.1 Settings

Datasets. Following previous LJP works (Zhong et al., 2018; Xu et al., 2020; Yue et al., 2021; Wu et al., 2023b), we conduct experiments on datasets CAIL2018 [2] and CJO22(Wu et al., 2023b). For the CAIL2018 dataset, we randomly divide it into training, validation, and test sets with a ratio of 8:1:1Wu et al. (2023b). For the CJO22 dataset, following Wu et al. (2023b), we use it solely as an additional test set. The table in Appendix F presents the details of two datasets.

Evaluation Metrics. We use Accuracy (Acc), Macro-Precision (Ma-P), Macro-Recall (Ma-R), and Macro-F1 (Ma-F) as the evaluation metrics following previous LJP works (Zhong et al., 2018; Xu et al., 2020; Yue et al., 2021; Wu et al., 2023b).

Baselines. We compare with: **(1) CNN**(LeCun et al., 1989) use different kernel convolutional operations to extract text features for classification; **(2) BERT**(Devlin et al., 2019) can be easily

fine-tuned on downstream tasks such as LJP; **(3) TopJudge**(Zhong et al., 2018) employs multi-task learning and captures the dependencies among the three sub-tasks in LJP; **(4) R-Former**(Dong and Niu, 2021) formalizes LJP as a node classification problem over a global consistency graph; **(5) LADAN**(Xu et al., 2020) uses graph distillation to extract discriminative features of the fact; **(6) NeurJudge**(Yue et al., 2021) splits the fact description into different parts for making predictions; **(7) EPM**(Feng et al., 2022) locates event-related information essential for judgment; **(8) CTM**(Liu et al., 2022) uses case triple modeling from contrastive case relations; **(9) PLJP**(Wu et al., 2023b) uses the collaboration between LLMs and domain-specific models for LJP; **(10) Llama3**(Wu et al., 2023b), which is Meta's advanced open-source language model; **(11) D-LADAN**(Xu et al., 2024) solves LJP confusion via graph and memory mechanisms.

Implementation Details. We directly adopted Bert-base-chinese[3] for candidate labels, Gpt-4o for rules optimization, and Llama3-chinese(Zhichen Zhang, 2024) for other steps because the language of the cases in the data set is Chinese. For the length limit, we use three precedents in Section 3.3. For the parameter settings of the baseline, we strictly follow the original paper. More details are in Appendix G.

4.2 Main Results

Tables 1 and Table 2 show the performance comparison results of RLJP and various baseline models on the CAIL2018 and CJO22 datasets, respectively, which are the average values obtained from five test turns. The experimental results show that RLJP achieved optimal performance in all metrics, verifying the enhancing effect of FOL judgment rules on LJP. Specifically, compared to the suboptimal model, RLJP achieved an average improvement of 1.43% in Acc and 14.98% in Ma-F on the CAIL2018 and CJO22 datasets. These experimental results fully validate the effectiveness and superiority of our method in LJP, indicating its significant advantages in legal case judgment.

It is worth noting that compared to the tasks of predicting legal provisions and criminal charges, the task of prison term prediction still presents significant challenges, mainly reflected in its relatively small performance improvement and lower overall prediction accuracy compared to the other two

---

[2] https://github.com/china-ai-law-challenge

[3] https://huggingface.co/google-bert/bert-base-chinese

tasks. This phenomenon may be related to the more complex sentencing factors and subjective judgments of judges involved in sentence prediction and is worth further exploration and improvement in subsequent research.

Method	Law Article				Charge				Prison Term			
	Acc	Ma-P	Ma-R	Ma-F	Acc	Ma-P	Ma-R	Ma-F	Acc	Ma-P	Ma-R	Ma-F
CNN(LeCun et al., 1989)	80.50	40.10	38.33	38.49	87.52	88.23	88.31	88.17	34.42	32.22	30.53	31.05
BERT(Devlin et al., 2019)	82.77	36.82	35.94	35.82	89.10	90.10	89.48	89.63	40.00	37.53	33.66	33.58
Roberta(Zhong et al., 2018)	83.08	48.09	44.25	44.87	90.30	91.02	90.97	90.94	40.84	38.62	38.55	38.50
TopJudge(Zhong et al., 2018)	80.46	40.96	40.96	38.24	87.31	88.68	87.84	88.20	35.54	33.55	31.08	32.00
R-Former(Dong and Niu, 2021)	87.82	56.13	56.57	55.81	91.54	91.61	91.96	91.58	40.70	36.09	36.76	35.04
LADAN(Xu et al., 2020)	82.82	42.57	39.00	40.71	88.09	90.12	88.82	89.47	38.03	33.66	30.08	31.77
Neurjudge(Yue et al., 2021)	76.91	55.95	52.92	53.56	82.13	82.71	82.30	82.36	33.53	36.46	37.26	36.53
EPM(Feng et al., 2022)	85.80	49.08	45.76	47.32	91.20	90.81	89.99	90.46	40.25	37.96	37.00	37.34
CTM(Liu et al., 2022)	84.72	46.46	44.83	45.10	90.28	90.34	88.08	86.30	39.56	38.66	38.02	37.84
Llama3-chinese(Zhichen Zhang, 2024)	2.62	4.41	1.66	1.11	22.63	22.53	9.58	10.04	12.00	14.91	19.04	10.53
PLJP(Bert)(Wu et al., 2023b)	87.07	58.81	57.29	56.63	94.99	92.12	91.10	91.33	48.72	42.64	36.80	35.43
D-LADAN(Xu et al., 2024)	91.05	56.37	54.91	58.38	90.82	66.26	62.51	62.08	38.31	27.48	26.01	25.00
RLJP(ours)	**91.27**	**85.72**	**91.27**	**88.32**	**96.00**	**96.53**	**96.00**	**96.10**	**54.72**	**48.01**	**54.73**	**48.45**

Table 1: Experiment results of LJP in CAIL2018 dataset. "**Bold**" indicates optimal results, and "underline" indicates sub-optimal results. The experimental results represent the average values obtained from five test rounds.

4.3 Ablation Experiment

To systematically evaluate the contributions of different components to the performance of LJP, we designed ablation experiments. Results of ablation experiments in Table 3 and Table 4, we can conclude that: (1) "w/o R" represents removing judgment rules (§3.3) and causes the decrease of all metrics, which proves judgment rules greatly help the LLM agent in reasoning legal judgment. (2) We find removing the optimization module ("w/o Optimize") causes the drop in metrics, proving that the dynamical Optimization module (§3.4) is beneficial for improving the rules used in the process of judgment prediction. (3) In row 3 ("w/o CL"), results show some metrics drop and some metrics up when we optimize the rule based on the latest version rather than CACL (§3.4.2). The up phenomenon shows the validity of optimization. The down phenomenon, caused by the content of the rule overfitting in some cases, shows that the performance-guided optimization method is important. (4) "w/o Candidate" removes candidate labels in the Examination module (§3.5), performing worse than RLJP. This indicates the importance of the lightweight model.

4.4 Analytical Experiment

To validate the performance advantages of RLJP in handling complex case facts, we selected the top 5% of cases as the test cases based on case length, which contain more details and complex circumstances. The table in Appendix H presents the statistics of these two subsets. We compared the performance of the second-ranked PLJP (§4.2) with our RLJP.

The experimental results in Table 5 and Table 6 demonstrate that the proposed RLJP method is over-performing in judging complex facts. We can conclude that FOL judgment rules can effectively capture key elements in fact of cases, reducing interference from redundant information and focusing on critical facts that are decisive for judgment prediction. In contrast, PLJP, which uses fixed three-type reorganized case facts, may ignore some important details. FOL judgment rules help the model better understand the logical structure and legal terminology in complex facts, which can focus on important logical details to reduce incorrect judgments caused by excessively long texts.

Method	Law Article				Charge				Prison Term			
	Acc	Ma-P	Ma-R	Ma-F	Acc	Ma-P	Ma-R	Ma-F	Acc	Ma-P	Ma-R	Ma-F
CNN(LeCun et al., 1989)	76.14	35.48	38.55	35.39	74.91	74.00	78.12	73.97	27.38	18.48	17.51	17.44
BERT(Devlin et al., 2019)	82.62	45.89	47.91	45.83	80.50	80.34	81.09	78.36	36.80	29.83	27.50	27.03
Roberta(Zhong et al., 2018)	80.32	42.36	44.22	41.80	79.26	78.93	81.25	78.18	29.74	24.73	24.76	23.22
TopJudge(Zhong et al., 2018)	78.73	40.38	41.47	40.09	76.67	74.00	77.40	74.62	27.14	19.76	17.69	17.94
R-Former(Dong and Niu, 2021)	87.69	53.03	49.35	50.23	90.71	_93.06_	88.66	_89.82_	38.63	32.63	32.76	29.51
LADAN(Xu et al., 2020)	79.44	48.43	44.13	46.18	79.64	48.43	44.13	46.18	33.69	26.40	22.94	24.55
Neurjudge(Yue et al., 2021)	71.38	52.86	53.52	52.62	71.85	69.37	71.09	68.66	26.80	26.81	26.85	25.97
EPM(Feng et al., 2022)	84.19	47.21	43.79	44.39	83.49	80.36	83.29	81.87	36.91	30.65	31.61	30.20
CTM(Liu et al., 2022)	79.44	47.83	42.25	43.43	79.33	82.39	83.12	82.81	36.81	27.10	25.96	26.46
Llama3-chinese(Zhichen Zhang, 2024)	3.01	2.21	1.09	1.36	27.72	37.12	27.72	28.28	15.66	19.18	15.66	15.56
PLJP(Bert)(Wu et al., 2023b)	_94.18_	_74.65_	_76.23_	_74.84_	94.18	90.25	_88.67_	89.05	43.52	_33.37_	_35.67_	_31.98_
D-LADAN(Xu et al., 2024)	90.65	44.95	48.45	44.49	88.95	29.41	27.64	28.48	_46.18_	24.80	23.78	22.73
RLJP(ours)	**94.55**	**94.30**	**90.94**	**91.28**	**96.12**	**97.60**	**96.76**	**96.83**	**48.50**	**49.44**	**50.19**	**49.18**

Table 2: Experiment results of LJP in CJO22 dataset. "**Bold**" indicates optimal results, and "underline" indicates sub-optimal results. The experimental results represent the average values obtained from five test rounds.

Table 3: Results of ablation experiments on CAIL2018. "w/o R", "w/o Optimize", "w/o CACL", and "w/o Candidate" denote removing judgment rules, optimization modules, CACL method, and candidate labels, respectively.

Method	Law Article		Charge		Prison Term	
	Acc	Ma-F	Acc	Ma-F	Acc	Ma-F
w/o R	18.92	26.53	82.43	81.25	28.05	16.89
w/o Optimize	85.9	82.98	83.13	84.28	39.74	41.16
w/o CACL	86.58	82.31	87.91	86.2	20.81	24.28
w/o Candidate	37.25	30.17	45.64	35.74	17.45	19.00
ours(RLJP)	**91.27**	**88.32**	**96.00**	**96.10**	**54.72**	**48.45**

Table 4: Results of ablation experiments on CJO22. "w/o R", "w/o Optimize", "w/o CACL", and "w/o Candidate" denote removing judgment rules, optimization modules, CACL method, and candidate labels, respectively.

Method	Law Article		Charge		Prison Term	
	Acc	Ma-F	Acc	Ma-F	Acc	Ma-F
w/o R	14.31	15.59	71.08	68.71	20.12	18.17
w/o Optimize	85.44	83.98	89.7	89.00	37.35	37.52
w/o CACL	84.98	84.08	91.8	90.36	31.52	32.63
w/o Candidate	21.05	15.06	71.58	68.86	23.53	23.17
ours(RLJP)	**94.55**	**91.28**	**96.40**	**96.58**	**48.50**	**49.18**

Table 5: Results of analytical experiments on CAIL2018_long. The experimental results represent the average values obtained from five test rounds.

Method	Law Article		Charge		Prison Term	
	Acc	Ma-F	Acc	Ma-F	Acc	Ma-F
PLJP(Bert)	28.06	36.17	40.94	44.74	12.32	13.92
ours(RLJP)	**79.27**	**78.27**	**91.67**	**91.81**	**35.14**	**36.23**

Table 6: Results of analytical experiments on CJO22_long. The experimental results represent the average values obtained from five test rounds.

Method	Law Article		Charge		Prison Term	
	Acc	Ma-F	Acc	Ma-F	Acc	Ma-F
PLJP(Bert)	38.93	58.87	42.64	44.71	23.26	22.29
ours(RLJP)	**41.67**	**47.62**	**97.70**	**91.95**	**31.70**	**35.96**

5 Conclusion

In summary, our proposed RLJP framework introduces three key innovations for LJP: (1) A dynamic rule optimization method that formulates judgment rule optimization with CACL as the process of tree-splitting, effectively addressing fixed rules' adaptability limitations in complex legal cases; (2) RLJP is a logic-semantic co-reasoning architecture combining lightweight semantic prescreening with FOL judgment rules to strengthen judgment reasoning capabilities; (3) Experimental validation demonstrating superior performance over all baseline methods in LJP tasks, particularly in complex, detailed cases.

Limitations

Although our approach produces promising results on two public datasets, there are certain limitations. In the future, we will continue to dig into these concerns.

First, we only evaluate RLJP on two Chinese LJP datasets. We do not conduct experiments on other languages' LJP datasets to evaluate the validity. Second, the interpretability of LJP is crucial, and

users need to understand how the model can get the judgment prediction results. In the examination module, RLJP combined COT with FOL rules and output the explanation for the result, but it lacked sufficient interpretability analysis for the judgment process and results of the model.

Ethical Consideration

While our RLJP framework enhances legal judgment prediction accuracy, its deployment requires rigorous ethical safeguards: (1) Human judges must retain final decision-making authority to mitigate risks from data biases or model errors; (2) Clear accountability protocols should ensure legal responsibility remains with human practitioners, not AI systems; (3) Ongoing fairness audits are necessary to prevent discriminatory outcomes against demographic minorities.

References

Duo Chai, Wei Wu, Qinghong Han, Fei Wu, and Jiwei Li. 2020. Description based text classification with reinforcement learning. In *International conference on machine learning*, pages 1371–1382. PMLR.

Long Chen, Nuo Xu, and Yue Wang. 2020. Legal judgment prediction with label dependencies. In *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 361–365. IEEE.

Yunfen Cheng, Chengyuan Chen, and Yuqi Wang. 2024. Research on legal judgment prediction with multi-task learning and causal logic. In *2024 6th International Conference on Electronic Engineering and Informatics (EEI)*, pages 399–402. IEEE.

Chenlong Deng, Kelong Mao, Yuyao Zhang, and Zhicheng Dou. 2024. Enabling discriminative reasoning in llms for legal judgment prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Findings of the Association for Computational Linguistics: EMNLP 2024, pages 784–796. Association for Computational Linguistics.

Wentao Deng, Jiahuan Pei, Keyi Kong, Zhe Chen, Furu Wei, Yujun Li, Zhaochun Ren, Zhumin Chen, and Pengjie Ren. 2023. Syllogistic reasoning for legal judgment analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13997–14009.

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. Association for Computational Linguistics.

Qian Dong and Shuzi Niu. 2021. Legal judgment prediction via relational learning. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 983–992. ACM.

Yi Feng, Chuanyi Li, and Vincent Ng. 2022. Legal judgment prediction: A survey of the state of the art. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 5461–5469. ijcai.org.

Leilei Gan, Kun Kuang, Yi Yang, and Fei Wu. 2021. Judgment prediction via injecting legal knowledge into neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 12866–12874.

Leilei Gan, Baokui Li, Kun Kuang, Yating Zhang, Lei Wang, Anh Luu, Yi Yang, and Fei Wu. 2023. Exploiting contrastive learning and numerical evidence for confusing legal judgment prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12174–12185, Singapore. Association for Computational Linguistics.

Congqing He, Tien-Ping Tan, Sheng Xue, and Yanyu Tan. 2025. Simulating judicial trial logic: Dual residual cross-attention learning for predicting legal judgment in long documents. *Expert Systems with Applications*, 261:125462.

Congqing He, Tien-Ping Tan, Xiaobo Zhang, and Sheng Xue. 2023. Knowledge-enriched multi-cross attention network for legal judgment prediction. *IEEE Access*, 11:87571–87582.

Zhitao He, Pengfei Cao, Chenhao Wang, Zhuoran Jin, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. 2024. Simucourt: Building judicial decision-making agents with real-world judgement documents. *arXiv preprint arXiv:2403.02959*.

Yu-Xiang Hong and Chia-Hui Chang. 2023. Improving colloquial case legal judgment prediction via abstractive text summarization. *Computer Law & Security Review*, 51:105863.

Ahmed Izzidien, Holli Sargeant, and Felix Steffek. 2024. Llm vs. lawyers: Identifying a subset of summary judgments in a large uk case law dataset. *arXiv preprint arXiv:2403.04791*.

Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551.

Shangyuan Li, Shiman Zhao, Zhuoran Zhang, Zihao Fang, Wei Chen, and Tengjiao Wang. 2025. Basis is also explanation: Interpretable legal judgment reasoning prompted by multi-source knowledge. *Information Processing & Management*, 62(3):103996.

Dugang Liu, Weihao Du, Lei Li, Weike Pan, and Zhong Ming. 2022. Augmenting legal judgment prediction with contrastive case relations. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 2658–2667. International Committee on Computational Linguistics.

Pengjie Liu, Wang Zhang, Yulong Ding, Xuefeng Zhang, and Shuang-Hua Yang. 2024a. Semdr: A semantic-aware dual encoder model for legal judgment prediction with legal clue tracing. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3447–3453. IEEE.

Pengjie Liu, Wang Zhang, Yulong Ding, Xuefeng Zhang, and Shuang-Hua Yang. 2024b. Semdr: A semantic-aware dual encoder model for legal judgment prediction with legal clue tracing. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3447–3453. IEEE.

Weicong Qin, Zelin Cao, Weijie Yu, Zihua Si, Sirui Chen, and Jun Xu. 2024. Explicitly integrating judgment prediction with legal document retrieval: A law-guided generative approach. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2210–2220.

Xuran Wang, Xinguang Zhang, Vanessa Hoo, Zhouhang Shao, and Xuguang Zhang. 2024. Legalreasoner: A multi-stage framework for legal judgment prediction via large language models and knowledge integration. *IEEE Access*.

Yiquan Wu, Yifei Liu, Weiming Lu, Yating Zhang, Jun Feng, Changlong Sun, Fei Wu, and Kun Kuang. 2022. Towards interactivity and interpretability: A rationale-based legal judgment prediction framework. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4787–4799.

Yiquan Wu, Siying Zhou, Yifei Liu, Weiming Lu, Xiaozhong Liu, Yating Zhang, Changlong Sun, Fei Wu, and Kun Kuang. 2023a. Precedent-enhanced legal judgment prediction with llm and domain-model collaboration. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 12060–12075. Association for Computational Linguistics.

Yiquan Wu, Siying Zhou, Yifei Liu, Weiming Lu, Xiaozhong Liu, Yating Zhang, Changlong Sun, Fei Wu, and Kun Kuang. 2023b. Precedent-enhanced legal judgment prediction with LLM and domain-model collaboration. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12060–12075. Association for Computational Linguistics.

Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. 2020. Distinguish confusing law articles for legal judgment prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 3086–3095. Association for Computational Linguistics.

Nuo Xu, Pinghui Wang, Junzhou Zhao, Feiyang Sun, Lin Lan, Jing Tao, Li Pan, and Xiaohong Guan. 2024. Distinguish confusion in legal judgment prediction via revised relation knowledge. *ACM Trans. Inf. Syst.*, 43(1):Article 6.

Shuxin Yang, Suxin Tong, Guixiang Zhu, Jie Cao, Youquan Wang, Zhengfa Xue, Hongliang Sun, and Yu Wen. 2022. Mve-flk: A multi-task legal judgment prediction via multi-view encoder fusing legal keywords. *Knowledge-Based Systems*, 239:107960.

Shunyu Yao, Qingqing Ke, Qiwei Wang, Kangtong Li, and Jie Hu. 2024. Lawyer gpt: A legal large language model with enhanced domain knowledge and reasoning capabilities. In *Proceedings of the 2024 3rd International Symposium on Robotics, Artificial Intelligence and Information Engineering*, pages 108–112.

Yaoyao Yu and Yihui Qiu. 2023. Enhancing legal judgment prediction with attentional networks utilizing legal event types. In *International Conference on Neural Information Processing*, pages 393–404. Springer.

Linan Yue, Qi Liu, Binbin Jin, Han Wu, and Yanqing An. 2024. A circumstance-aware neural framework for explainable legal judgment prediction. *IEEE Transactions on Knowledge and Data Engineering*.

Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. 2021. Neurjudge: A circumstance-aware neural framework for legal judgment prediction. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 973–982. ACM.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, and Xuanjing Huang. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325*.

Han Zhang and Zhicheng Dou. 2023. Case retrieval for legal judgment prediction in legal artificial intelligence. In *China National Conference on Chinese Computational Linguistics*, pages 434–448. Springer.

Han Zhang, Zhicheng Dou, Yutao Zhu, and Ji-Rong Wen. 2023. Contrastive learning for legal judgment prediction. *ACM Transactions on Information Systems*, 41(4):1–25.

Yunong Zhang, Xiao Wei, and Hang Yu. 2024. Hdljp: A hierarchical dependency-based legal judgment prediction framework for multi-task learning. *Knowledge-Based Systems*, page 112033.

Long Chen Zhichen Zhang, Xin LU. 2024. Llama3-chinese. https://github.com/seanzhang-zhichen/llama3-chinese.

Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3540–3549. Association for Computational Linguistics.

A The construction of quiz experience.

Figure 2: The construction of quiz experience.

B The prompt template for optimization direction generation.

Figure 3: The prompt template for optimization direction generation.

C The prompt template for the quiz in the confusable case set.

Figure 4: The prompt template for the quiz in the confusable case set.

D The prompt template for optimization direction generation.

Figure 5: The prompt template for optimization direction generation.

E The prompt template for generating a case abstract.

Figure 6: The prompt template for generating a case abstract.

F The Statistics of LJP Datasets

Table 7 is the statistics of two realistic datasets, CAIL2018 and CJO22, which are widely used in the task of LJP. These datasets have been rigorously preprocessed to ensure compliance with ethical and legal standards. All personally identifying information (PII), including names, addresses, identification numbers, and other sensitive identifiers, has been anonymized or removed. Additionally, these datasets uncover offensive, discriminatory, or harmful content, as such material was systematically filtered during data curation. These measures align with established guidelines for ethical AI research and protect the privacy and dignity of individuals involved in legal cases.

Type	CAIL2018	CJO22
Law Articles	164	164
Charges	42	42
Prison Terms	10	10
Samples	82138	1698
Avg. # Fact words	288.6	461.7

Table 7: Datasets statistics

G The Experimental setting

Our computing infrastructure was robust, featuring two A800 GPUs, each equipped with 80GB of memory, providing the necessary computational power to handle the large-scale data and complex algorithms involved in our experiments. Additionally, we utilized a high-performance computing cluster with a multi-core CPU architecture, high-speed NVMe storage for rapid data access, and a high-bandwidth network interconnect to ensure efficient data transfer and processing. The operating environment was Linux-based, and we employed CUDA and cuDNN libraries to optimize GPU performance. All these details are crucial for understanding the computational resources that underpinned our experimental results.

H The Statistics of Two Extracted Datasets

Type	CAIL2018_long	CJO22_long
Law Articles	43	29
Charges	41	28
Prison Terms	10	10
Samples	7499	83
Avg. # Fact words	1147.5	10815.9
Max. # Fact words	20397	43030

Table 8: Test sets in the analytical experiment, which contain the top 5% of cases in length from CAIL2018 and CJO22

Table 8 shows the two subsets statistics of two realistic datasets CAIL2018 and CJO22 as the test set used in the analytical experiment. The two subsets contain the top 5% of cases sorted by case length from CAIL2018 and CJO22.