



SyLeR: A Framework for Explicit Syllogistic Legal Reasoning in Large Language Models

Kepu Zhang
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
kepuzhang@ruc.edu.cn

Weijie Yu*
School of Information Technology
and Management
University of International Business
and Economics
Beijing, China
yu@uibe.edu.cn

Zhongxiang Sun
Jun Xu
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
{sunzhongxiang,junxu}@ruc.edu.cn

Abstract

Syllogistic reasoning is a fundamental aspect of legal decision-making, enabling logical conclusions by connecting general legal principles with specific case facts. Although existing large language models (LLMs) can generate responses to legal questions, they fail to perform explicit syllogistic reasoning, often producing implicit and unstructured answers that lack explainability and trustworthiness. To address this limitation, we propose SyLeR, a novel framework that empowers LLMs to engage in explicit syllogistic legal reasoning. SyLeR integrates a tree-structured hierarchical retrieval mechanism to effectively combine relevant legal statutes and precedent cases, forming comprehensive major premises. This is followed by a two-stage fine-tuning process: supervised fine-tuning warm-up establishes a foundational understanding of syllogistic reasoning, while reinforcement learning with a structure-aware reward mechanism refines the model's ability to generate diverse logically sound and well-structured reasoning paths. We conducted extensive experiments across various dimensions, including in-domain and cross-domain user groups (legal laypersons and practitioners), multiple languages (Chinese and French), and different LLM backbones (legal-specific and open-domain LLMs). The results show that SyLeR significantly improves response accuracy and consistently delivers explicit, explainable, and trustworthy legal reasoning.

CCS Concepts

• Information systems → Information retrieval; • Applied computing → Law.

Keywords

Legal Reasoning, Legal RAG, Reinforcement Learning

ACM Reference Format:

Kepu Zhang, Weijie Yu, Zhongxiang Sun, and Jun Xu. 2025. SyLeR: A Framework for Explicit Syllogistic Legal Reasoning in Large Language Models. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25)*, November 10–14, 2025, Seoul, Republic of Korea.

*Weijie Yu is the corresponding author. Work partially done at Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education.



This work is licensed under a Creative Commons Attribution 4.0 International License. *CIKM '25*, November 10–14, 2025, Seoul, Republic of Korea
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2040-6/2025/11
<https://doi.org/10.1145/3746252.3761120>

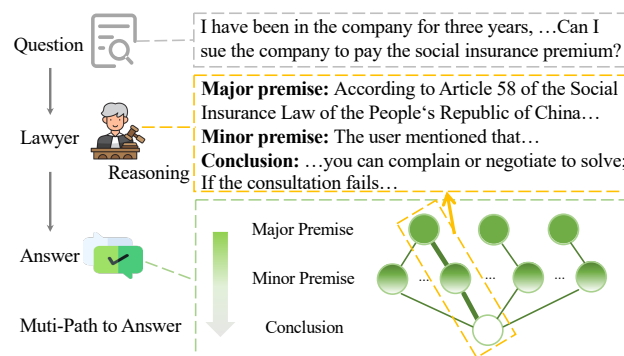


Figure 1: Syllogistic reasoning in legal QA: The lawyer identifies the relevant legal statutes or precedent cases to derive the major premise, formulates the minor premise from the user's question, and then reaches the conclusion. Based on different analysis perspectives, different syllogistic reasoning paths can be derived.

Korea. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3746252.3761120>

1 Introduction

Legal decision-making fundamentally depends on logical reasoning to ensure fairness and trustworthy in judgments. Among the various forms of legal reasoning, syllogistic reasoning [3, 12] stands out as a structured form of deductive reasoning that derives conclusions from two interconnected premises. In legal contexts, the major premise typically comprises general legal rules or principles—such as statutes and precedents—while the minor premise is drawn from the specific facts of a case. The logical connection between these premises leads to a well-founded legal conclusion. This reasoning process closely reflects how legal professionals analyze cases, ensuring that decisions are both grounded in established laws and logically aligned with case facts, Figure 1 is an example.

Despite the fundamental role of syllogistic reasoning in legal decision-making, existing large language models (LLMs) struggle to replicate this explicit reasoning process in legal tasks. On the one hand, legal-specific LLMs [33, 34, 38] have primarily focused on enhancing legal knowledge through supervised fine-tuning on domain-specific datasets. These models not only require a substantial amount of annotated data but also lack clear, logically structured

explanations that mirror how legal professionals connect statutes and case facts to reach decisions. This limitation undermines both the explainability and trustworthiness of legal LLMs, hindering their practical deployment in scenarios where transparent and well-justified reasoning is critical. On the other hand, research aimed at enhancing the reasoning capabilities of LLMs has introduced methods such as Chain-of-Thought (CoT) prompting [30, 32], supervised fine-tuning [8], and retrieval-augmented fine-tuning [44]. These techniques enhance the model’s reasoning ability by either implicitly guiding the model or explicitly adjusting the model parameters, thereby improving performance in tasks involving mathematics, commonsense reasoning, and code generation. However, these approaches remain largely domain-agnostic and fail to address the distinct challenges of legal reasoning, particularly the need for generating explicit syllogistic reasoning paths that align legal rules with case-specific facts.

Enabling LLMs to perform explicit syllogistic legal reasoning involves addressing several complex challenges. **First**, effectively incorporating comprehensive legal knowledge into LLMs is inherently difficult. Legal reasoning depends on both statutes and precedent cases [42], which are not only diverse in form but also deeply interconnected. Statutes provide general legal principles, while precedents interpret and apply these statutes to specific cases. Treating these sources separately can disrupt their contextual relationships, making it difficult for models to capture the nuanced interplay between legal rules and their applications. Therefore, designing models that can holistically incorporate and reason over both statutes and precedents within a structured framework is essential. **Second**, achieving explicit syllogistic reasoning requires the model to learn how to construct valid reasoning paths. However, for any given legal question, there may exist multiple valid syllogistic reasoning paths leading to the same conclusion, as shown in Figure 1. Annotating diverse reasoning paths for supervised fine-tuning is expensive and labor-intensive, particularly in the legal domain where expert annotation is required. More importantly, how to train LLMs to learn from and generalize across multiple reasoning strategies—without overfitting to a single annotated path—is a complex challenge that must be addressed. **Third**, evaluating an LLM’s performance in syllogistic legal reasoning demands more than standard text generation metrics. Conventional metrics such as accuracy and ROUGE fall short in assessing the logical consistency, coherence, and legal validity of the model’s reasoning processes. Without specialized evaluation criteria, it is difficult to determine whether a model truly understands and applies syllogistic reasoning. Therefore, it is crucial to develop task-specific evaluation methods that can effectively measure the soundness and transparency of the model’s reasoning chains, ultimately guiding the model toward producing more trustworthy and explainable legal outcomes.

To address these challenges, we propose **SyLeR**, a novel framework designed to enable LLMs to perform explicit **Sy**llogistic **Le**gal **R**easoning. SyLeR integrates a two-stage training process that systematically incorporates legal knowledge and guides the model to generate logically structured reasoning. Specifically, we first introduce a tree-structured hierarchical retrieval mechanism that organizes legal knowledge by connecting relevant statutes and their corresponding precedent cases. This structure preserves the

inherent relationship between legal rules and their practical applications, allowing the model to construct a comprehensive major premise that reflects both legal principles and real-world interpretations. Building on this, SyLeR employs a two-stage reinforcement fine-tuning strategy. In the warm-up stage, we utilize GPT-4o to generate high-quality syllogistic reasoning paths by reverse engineering from retrieved legal documents, user questions, and ground-truth answers. These reasoning paths explicitly follow the structure of major premise, minor premise, and conclusion, providing the foundation for supervised fine-tuning that teaches the model how to produce logically coherent and structured responses. In the reinforcement learning (RL) stage, we refine the model’s reasoning ability using the Proximal Policy Optimization (PPO) [27] algorithm. This stage allows the model to explore diverse syllogistic reasoning paths, addressing the challenge of reasoning diversity. A carefully designed reward function evaluates the model’s output based on the alignment between the generated major premise and the retrieved legal knowledge, the consistency of the minor premise with the user question, and the correctness of the conclusion. To ensure strict adherence to the syllogistic reasoning format, responses that deviate from the expected structure receive zero rewards, guiding the model toward generating explicit and legally sound reasoning.

We conducted extensive experiments to evaluate SyLeR across multiple dimensions, including datasets from legal laypersons and practitioners, multiple languages (Chinese and French), and comparisons between legal-specific and open-domain LLMs. Additionally, cross-domain tests were performed by training on one user group and evaluating on another to assess adaptability. The results show that SyLeR significantly improves answer accuracy and provides explicit syllogistic reasoning, ensuring both trustworthiness and explainability across diverse legal scenarios.

In summary, the contributions of this paper are as follows:

- We pioneered the explicit incorporation of syllogistic legal reasoning into LLMs through reinforcement fine-tuning, enabling the LLM to generate responses in the format of major premise, minor premise, and conclusion.
- We propose SyLeR, a novel framework that achieves explicit syllogistic legal reasoning through a tree-structured hierarchical retrieval mechanism for legal knowledge integration and a two-stage fine-tuning process. This process combines supervised learning for foundational reasoning and reinforcement learning with a structure-aware reward mechanism that evaluates the alignment of the major premise, minor premise, and conclusion, ensuring logically sound and structured outputs.
- We conduct extensive experiments across diverse settings including different user groups (legal laypersons and practitioners), multiple languages (Chinese and French), and model types (legal-specific and open-domain LLMs) demonstrating that SyLeR significantly improves response accuracy and provides explicit and trustworthy legal reasoning across various legal contexts.

2 Related Work

2.1 Legal LLM

With the rise of LLMs, a large number of studies investigating the use of LLMs in the legal domain have emerged [2, 12, 40, 41]. Early works explored the capabilities of LLMs in various legal tasks, using prompt-based methods [31, 37] to provide examples to guide the LLM in solving tasks, including legal judgment prediction, legal examinations, and more. Some works have also designed benchmarks [5, 18] to test the capabilities of LLMs in legal tasks, evaluating them from multiple dimensions. As publicly available legal-related data resources have become more abundant, constructing fine-tuning and training data based on these datasets has become easier. As a result, many studies have fine-tuned open-domain LLMs on large amounts of legal-related data, injecting legal knowledge into the models to create LLMs specifically for the legal field, enabling them to tackle legal-related tasks. For instance, [38] fine-tuned LLM using high-quality supervised datasets constructed from judgment prediction, information extraction, and legal QA datasets. [34] conducted secondary pre-training and instruction fine-tuning on open-domain LLMs using datasets such as legal documents and legal cases. [21] obtained high-quality judicial QA data through ChatGPT for fine-tuning. [9] performed full-parameter fine-tuning to create legal-specific LLMs. Despite the promising results achieved by these legal LLMs on legal tasks, the final answers they provide are still a form of implicit reasoning. None of these methods have enabled the LLMs to perform explicit legal reasoning, which affects the trustworthiness of the LLMs and hinders their deployment in real-world scenarios.

2.2 Reasoning in LLM

Existing work has explored various strategies to enhance the reasoning capabilities of large language models (LLMs). One such strategy is the Chain-of-Thought (CoT) prompting technique [30, 45], which encourages the model to generate a series of intermediate reasoning steps through specific prompts. For example, the simple prompt "Let's think step by step" [14] has been shown to improve the reasoning ability of LLMs. Further, In-Context Learning (ICL) [32] involves providing examples containing intermediate reasoning steps to teach the model how to perform reasoning. In the retrieval-augmented generation scenarios, enhancing the LLM's reasoning ability can be achieved by improving the prompts input to the LLM using its feedback, including compressing the retrieved documents [13, 19] and iteratively retrieval to augment generation [29]. Additionally, supervised fine-tuning using task-related data [8], enhancing supervised fine-tuning with retrieved document [44], and incorporating CoT data into fine-tuning [23] can all strengthen the reasoning ability of LLMs by adjusting their parameters. Recently, the release of Deepseek-R1 [7] and OpenAI's o1 has inspired many works [39, 47] to explore reasoning techniques similar to o1, emphasizing the importance of scaling CoT and reinforcement learning to strengthen LLM reasoning abilities. However, all of the aforementioned works primarily focus on open-domain problems such as mathematics, code, and commonsense reasoning, without addressing explicit reasoning in the legal domain. This gap is the focus of our paper, which aims to equip LLMs with the ability to perform explicit syllogistic legal reasoning.

3 Method

3.1 Problem Statement

We focus on the legal question-answering (QA) task in this paper. Formally, suppose we are given a set of labeled data tuples $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{T}|}$, where $|\mathcal{T}|$ is the total number of tuples. For each tuple, x_i represents the i -th user's question, and y_i is the manually annotated answer corresponding to the question.

Unlike current Legal QA studies, which directly generate an answer y_i from a given user question x_i , we aim to empower LLMs with explicit syllogistic legal reasoning capabilities. Specifically, the objective of our work is to produce \hat{y}_i to x_i by constructing a structured reasoning path \mathcal{P}_i . This path includes a major premise p_i^{major} , derived from relevant legal statutes and precedent cases, and a minor premise p_i^{minor} , extracted from the specific facts in x_i , which are combined to logically infer \hat{y}_i as the conclusion.

3.2 Overall Framework

As illustrated in Figure 2, the proposed SyLeR framework consists of two primary components designed to achieve explicit syllogistic legal reasoning.

First, given a user's legal question x_i , SyLeR retrieves the relevant legal foundation to construct the major premise in the syllogistic reasoning process. This is achieved through a hierarchical retrieval mechanism applied to the constructed legal knowledge tree \mathcal{D} :

$$\mathcal{K}_i = \text{Retriever}(x_i, \mathcal{D}), \quad (1)$$

where $\mathcal{K}_i \subseteq \mathcal{D}$ represents the subset of retrieved legal knowledge, including legal statutes and precedent cases relevant to x_i . This retrieval mechanism ensures the interpretive connection between statutes and cases is preserved. The details of the corpus construction and retrieval process are elaborated in § 3.3.

With the major premise established, SyLeR is responsible for deducing the conclusion based on the user's question, which serves as the minor premise. This step integrates the retrieved knowledge \mathcal{K}_i with the question x_i to produce the final legally coherent syllogistic reasoning path \mathcal{P} :

$$\mathcal{P}_i = \text{LLM}(\mathcal{K}_i, x_i). \quad (2)$$

where $\mathcal{P}_i = \{p_i^{\text{major}}, p_i^{\text{minor}}, \hat{y}_i\}$ is the structured syllogistic output including the generated major premise, minor premise, and conclusion. To enhance reasoning diversity and ensure logical consistency, the framework leverages multiple reasoning paths and a structured reward mechanism, which are detailed in § 3.4.

3.3 Tree-Structured Hierarchical Retrieval

The first step in SyLeR is to retrieve relevant legal knowledge to construct the major premise required for explicit syllogistic legal reasoning. This step involves a hierarchical retrieval mechanism applied to the legal knowledge tree \mathcal{D} . As discussed, this tree is designed to preserve the interpretive connections between legal statutes and precedent cases, ensuring a robust and logically sound foundation for reasoning. In this section, we describe how the legal knowledge tree is constructed and how the retrieval mechanism operates.

Construction of legal knowledge tree \mathcal{D} . The legal knowledge corpus \mathcal{D} is organized into a hierarchical tree structure to

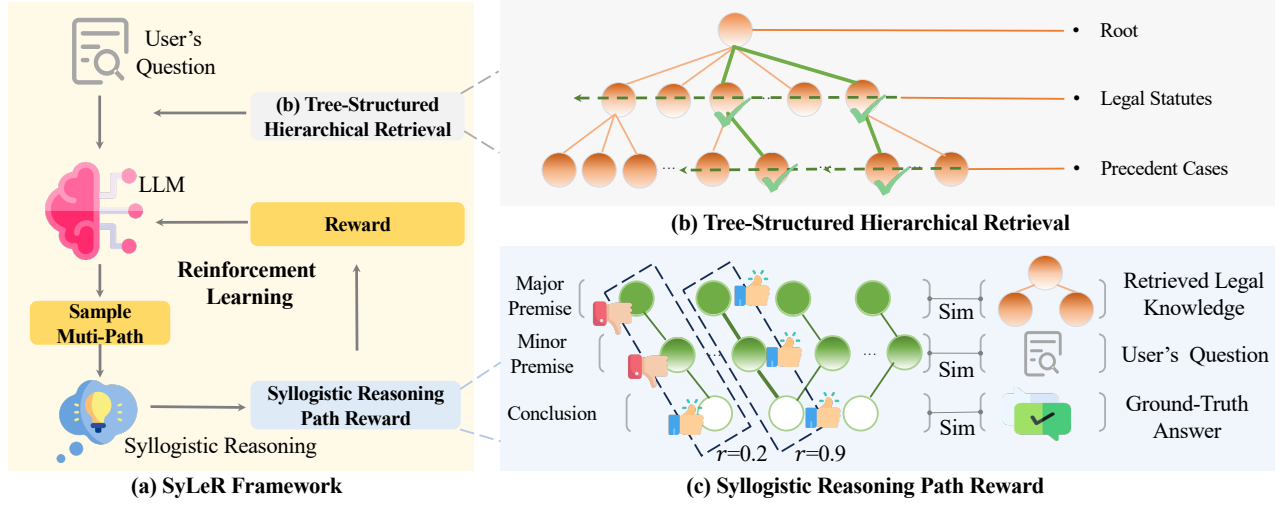


Figure 2: The architecture of SyLeR. (a) The framework of SyLeR, which includes a tree-based retriever and trains the LLM by sampling multiple syllogistic reasoning paths through reinforcement learning, with rewards based on the reasoning paths. (b) The tree-based retriever, which first retrieves legal statutes and then retrieves corresponding cases under those statutes. (c) The calculation of reasoning path rewards, which is obtained by measuring the similarity between the model-generated major premise, minor premise, and conclusion with the retrieved legal knowledge, the user’s question, and the answer.

capture the relationships between general legal principles (legal statutes) and specific applications (precedent cases). As shown in Figure 2 (b), the tree consists of three key components: (1) The root of the tree is a virtual node representing the starting point of retrieval. (2) The first layer contains nodes representing legal statutes \mathcal{D}_L , which provide general legal principles relevant to the user’s question. These statutes form the backbone of the major premise by defining the foundational rules of the legal system. (3) The second layer contains nodes representing precedent cases \mathcal{D}_C , which are linked to their corresponding statutes in \mathcal{D}_L . Precedent cases illustrate how the general principles from statutes have been interpreted and applied in specific legal scenarios. This hierarchical structure ensures that the major premise is comprehensive, combining both general principles and contextualized interpretations. Additionally, the tree structure allows for efficient navigation and retrieval of relevant knowledge.

To construct \mathcal{D} , we use the legal corpus provided by [42], which contains 54,114 statutes and 435,579 cases. Each statute and case is encoded into a dense vector representation using BGE [35]. Given the vector representation for each case, we identify the most relevant statute from the corpus based on the cosine similarity of their embeddings, creating an edge between the statute and the case. To enhance connectivity, statutes with fewer than five connected cases are supplemented by linking additional cases. These additional cases are determined by identifying the most relevant statute and its associated cases based on cosine similarity. This approach ensures that the tree captures meaningful relationships between statutes and cases while maintaining sufficient connectivity for robust retrieval and reasoning.

Hierarchical retrieval. Given a user question x_i , the retrieval process leverages the hierarchical tree structure to retrieve relevant

statutes and cases efficiently, as illustrated in Figure 2 (b). The retrieval mechanism proceeds in two steps. In the first-layer statute retrieval, the user question x_i is embedded using BGE [35]. We compute the cosine similarity between the question embedding and the embeddings of all statutes in the statute layer. The top- k most similar statutes are selected as retrieval results, which are denoted by $L = \{l_1, l_2, \dots, l_{K_L}\}$, where K_L represents the number of retrieved statutes. In this paper, as the legal question typically relate to a single statute, we set $K_L = 1$. In the second-layer precedent cases retrieval, for each retrieved statute l_i , we traverse its associated case nodes in the tree. For each case under l_i , we compute the cosine similarity between the question embedding and the case embeddings. We select the top- k most similar cases for each statute, which are denoted by $C_i = \{c_{i1}, c_{i2}, \dots, c_{iK_C}\}$, where C_i represents the set of retrieved cases under statute l_i . To address the input length limitations of LLMs, we set $K_C = 3$, ensuring that only the most relevant cases are included in the retrieved subset. By combining these two steps, the retrieval mechanism outputs \mathcal{K}_i , a subset of legal knowledge containing both statutes and precedents that are logically and semantically connected. This ensures that the major premise for syllogistic reasoning is comprehensive, cohesive, and tailored to the user’s legal question. The retrieved knowledge \mathcal{K}_i is then passed to the next stage of SyLeR for structured reasoning.

3.4 Two-Stage Reinforcement Fine-Tuning

With the relevant legal knowledge \mathcal{K}_i retrieved using the tree-structured hierarchical retrieval mechanism, the next step in SyLeR is to enable the LLM to generate explicit syllogistic reasoning paths and produce the legally coherent answer \hat{y}_i . This is achieved through a reinforcement fine-tuning process that consists of two stages: a warm-up stage using supervised fine-tuning and a reinforcement learning (RL) stage. This fine-tuning process ensures

that the model generates logically sound and interpretable reasoning paths that adhere to the major premise \rightarrow minor premise \rightarrow conclusion structure.

Warm-up stage establishes the LLM’s foundational ability to perform syllogistic legal reasoning. Specifically, we employ GPT-4o [11] to generate 10 syllogistic reasoning paths for each of $(x, y) \in \mathcal{T}$ using the corresponding retrieved documents \mathcal{K} as shown in Eq. 1. Each reasoning path \mathcal{P} consists of the major premise, derived by summarizing the retrieved statutes and precedent cases in \mathcal{K} ; the minor premise, which originates from the specific facts provided in the question x ; and the conclusion, corresponding to the ground-truth answer y . Using these reasoning paths, we perform supervised fine-tuning with LoRA [10] to train the LLM to generate responses in the structured format as follows:

$$\mathcal{L}_{\text{FT}} = -\frac{1}{|\mathcal{T}|} \sum_{\mathcal{T}} \log (P_{\theta+\theta_L}(\mathcal{P}_t | x, \mathcal{K}, \mathcal{P}_{<t})), \quad (3)$$

where θ and θ_L are the parameters of LLM and LoRA; \mathcal{P}_t and $\mathcal{P}_{<t}$ respectively denote the t -th token and tokens before \mathcal{P}_t in the corresponding syllogistic reasoning path. This stage provides the model with an initial understanding of syllogistic reasoning and prepares it for further refinement during the next stage.

Reinforcement learning. Building upon the warm-up stage, we employ the Proximal Policy Optimization (PPO) algorithm [27] to refine the model’s reasoning capabilities. The reinforcement learning stage focuses on enabling the LLM to explore multiple reasoning paths for each question, avoiding overfitting to a single reasoning strategy and enhancing the diversity of logical reasoning. As shown in Figure 2 (c), for the response where only the correct answer is provided without a reasonable reasoning path, we assign a lower reward, as the lack of a correct reasoning process can affect the reliability of the reasoning. During training, LLM samples reasoning paths for a given question x_i by combining the retrieved knowledge \mathcal{K}_i with the question. Each sampled reasoning path is evaluated based on a reward mechanism designed to assess its quality and adherence to the syllogistic structure. PPO optimizes the model by iteratively adjusting the policy to maximize the expected reward while ensuring the updates remain within a stable region to avoid overfitting or catastrophic updates.

Reward design for syllogistic reasoning. The reward mechanism plays a crucial role in guiding LLM to generate logically sound syllogistic reasoning paths. For each sampled reasoning path, the reward is computed based on the following criteria:

$$r = \begin{cases} 0 & \text{if not syllogistic,} \\ \frac{1}{2} \text{sim}(p^{\text{major}}, l) + \frac{1}{2} \sum_{i=1}^3 \frac{1}{3} \text{sim}(p^{\text{major}}, c_i) \\ \quad + \text{sim}(p^{\text{minor}}, x) + \text{rouge}_{\text{sum}}(\hat{y}, y) & \text{otherwise,} \end{cases} \quad (4)$$

$\text{sim}(\cdot)$ represents the cosine similarity computed using Sentence-BERT embeddings [25], while $\text{rouge}_{\text{sum}}(\cdot)$ denotes the exponential form of combined Rouge-1, Rouge-2, and Rouge-L scores. The terms in Eq. 4 are defined as follows:

- $\text{sim}(p^{\text{major}}, l)$ measures the alignment between the generated major premise (p^{major}) and the retrieved statute (l).

- $\sum_{i=1}^3 \frac{1}{3} \text{sim}(p^{\text{major}}, c_i)$ averages the similarity between the major premise and the three retrieved precedent cases $\{c_1, c_2, c_3\}$.
- $\text{sim}(p^{\text{minor}}, x)$ evaluates how well the generated minor premise (p^{minor}) aligns with the question (x).
- $\text{rouge}_{\text{sum}}(\hat{y}, y)$ measures the similarity between the generated conclusion (\hat{y}) and the ground-truth answer (y) using ROUGE metrics.

The reward r is only assigned at the terminal state, i.e., when the generation is completed. All intermediate actions in non-terminal states receive a reward of zero. To further enforce the adherence to syllogistic reasoning, any invalid output—such as one that deviates from the expected major premise \rightarrow minor premise \rightarrow conclusion structure—is penalized with a reward of zero.

Following [23, 46], the total reward also incorporates a KL divergence [15] term between the RL model and the warm-up model. This regularization stabilizes training by penalizing the RL policy for diverging excessively from the warm-up policy. The final reward is defined as:

$$r_{\text{final}} = r - \beta \text{KL}(\pi_{\theta}(\cdot | s_t), \pi_{\theta_{\text{SFT}}}(\cdot | s_t)), \quad (5)$$

where π_{θ} and $\pi_{\theta_{\text{SFT}}}$ represent the RL policy and the warm-up stage policy, respectively, and β is a scaling coefficient that controls the contribution of the KL divergence to the overall reward.

4 Experiment

In this section, we conduct experiments to answer the following research questions: **RQ1:** How does SyLeR perform compared to existing legal LLMs and open-domain LLM baselines in legal layperson and practitioner datasets? **RQ2:** What is the impact of the three modules in SyLeR on the model’s performance? **RQ3:** How does SyLeR perform across different domains? **RQ4:** How robust is SyLeR across different languages? **RQ5:** Can SyLeR be adapted to various LLM backbones? **RQ6:** Are the responses generated by SyLeR truly trustworthy?

4.1 Experimental Setup

4.1.1 Dataset. We conducted experiments on three legal QA datasets: two Chinese datasets, Practitioner and Layperson, targeting different user groups, and one French dataset, LLeQA.

Practitioner [17]: This dataset is derived from LexEval, which collects questions and answers from the subjective sections of the National Unified Legal Professional Qualification Examination. It represents questions from legal professionals, including detailed case descriptions that require LLMs to thoroughly understand the context and generate answers based on relevant legal knowledge.

Layperson [5]: This dataset is sourced from LawBench, which compiles user questions and responses from the Hualv website¹, where non-legal professionals seek advice from lawyers and legal practitioners. The questions are shorter and more conversational in nature, requiring LLMs to leverage legal knowledge for appropriate responses. For the corpus associated with these two datasets, we utilized a knowledge base provided by [42], which includes legal statutes and precedent cases.

¹www.66law.com

Table 1: The statistical analysis of the datasets used in this paper. #Train and #Test represent the number of data points in the training and testing sets, respectively. Len_Q and Len_A refer to the average question length and answer length.

Dataset	#Train	#Test	Len_Q	Len_A
Layperson	100	400	57.62	107.40
Practitioner	100	400	618.96	193.46
LLeQA	1472	195	15.39	305.92

LLeQA [22]: This is a French dataset with expert-annotated legal questions covering diverse topics such as housing, family, and employment. The questions often require longer answers, demanding that LLMs refer to relevant legal statutes to provide detailed responses. For the corpus of this dataset, we used [22], which contains 27,942 French legal statutes.

Statistical analyses of the three datasets are shown in Table 1. We fine-tuned LLMs using the training subsets of these datasets and evaluated performance on their respective test sets.

4.1.2 Evaluation Metrics. We employed widely used metrics to evaluate the similarity between model-generated responses and reference answers, including Rouge [20] (Rouge-1, Rouge-2, and Rouge-L), which assess recall; BLEU [24], which evaluates precision; and BERTScore (BERT) [43], which measures semantic similarity by leveraging contextual embeddings.

4.1.3 Baselines. We compare two types of LLMs: legal-specific LLMs and open-domain LLMs. Legal LLMs include **zhilai** [34], **fuzi.mingcha** [33], **DISC-LawLLM** [38], **LawGPT_zh** [21], **Tailing²**, **LexiLaw³**, **HanFei** [9]. These models have been fine-tuned on extensive datasets from the legal domain, enabling them to handle legal tasks.

We select Qwen2 [36] as the representative base LLM for our experiments. For open-domain LLMs, in addition to the **standard RAG** [6, 16] approach, where the LLM generates responses based on retrieved laws and related cases, we compared prompt-based methods designed to handle the long document lengths common in legal scenarios: **Selective-Context** [19] removes redundant information from the provided context using self-information [28]. **LongLLMLingua** [13] employs a coarse-to-fine, question-aware compression strategy to condense lengthy documents. To address the limitation of single-pass retrieval, which may fail to provide all the knowledge needed for legal questions, we also compared the iterative retrieval method **Iter-RetGen** [29].

For fine-tuning methods, we compared the following approaches: **Naive SFT** [8]: A straightforward supervised fine-tuning approach using question-answer pairs for training. **RAG-SFT** [44]: Fine-tuning the LLM using questions combined with retrieved documents as input and answers as output. **CoT-SFT** [23]: This method uses GPT-4o to generate direct reasoning paths based on the question, retrieved legal documents and answer. During fine-tuning, the question and documents are used as input, while the reasoning path and conclusion are used as output to train the LLM.

²<https://github.com/DUTIR-LegalIntelligence/Tailing>

³<https://github.com/CSHaitao/LexiLaw>

4.1.4 Implementation Details. For all models in the main experiments, we use BGE (specifically, bge-base-zh-v1.5) [35] to retrieve one legal statute and three precedent cases. We use the training dataset for fine-tuning and reinforcement learning training. For our method, following [23], during the warm-up stage, we fine-tune for 3 epochs. During the RL stage, we train for 10 epochs. For fine-tuning baselines, we train for 10 epochs to ensure full convergence. In the fine-tuning and warm-up stage, we set the maximum input length to 4096, the learning rate to $5e-5$, and the batch size to 16. For both the fine-tuning and RL stages, we use LoRA [10] for efficient training of the LLM, setting LoRA parameters with a rank of 8 and an α of 16. For the RL stage, we use PPO [27] for training, with the learning rate set to $1e-5$, β set to 0.02, the clip range set to 0.2, and the batch size set to 16. During LLM generation, we set `do_sample` to false to enhance the reproducibility of the results. For the calculation of BERTScore, we use bert-base-chinese [4] for the Layperson and Practitioner datasets, and bert-base-multilingual-cased for the LLeQA dataset. All experiments are conducted on A6000 GPUs. More details can be found at <https://github.com/ke-01/SyLeR>.

4.2 Overall Performance (RQ1)

Table 2 presents the results of SyLeR and the baselines on the Layperson and Practitioner datasets. We can draw the following conclusions:

SyLeR has the best overall performance. It can be seen that SyLeR achieves the best performance across all metrics on both datasets, which demonstrates the effectiveness of SyLeR. Compared to legal LLMs, which have been fine-tuned with a large amount of legal knowledge, SyLeR outperforms them due to they primarily rely on implicit reasoning to obtain answers, which results in inferior performance. Compared to baseline methods in open-domain, these methods may rely solely on a single reasoning path, such as SFT. In contrast, SyLeR offers various reasoning paths, which leads to improved reasoning accuracy in the final model outputs.

Legal LLMs performer well. Although the base LLM used for the legal LLM is an earlier version of open-domain LLM and performs worse than Qwen2 on various tasks, the legal LLM still demonstrates strong performance. This is because it has been fine-tuned on a large amount of legal-related datasets, embedding a significant amount of legal knowledge. Even without adding any strategies, such as compressing the retrieved documents, it performs second only to our proposed SyLeR method on certain metrics, such as the Lexilaw task on the Layperson dataset. This result highlights the importance of legal knowledge in enhancing the performance of LLMs for legal tasks.

The benefits of fine-tuning. As observed, methods that enhance LLM reasoning through fine-tuning have outperformed methods that enhance reasoning based on prompts across both datasets. This is because legal reasoning is significantly different from traditional reasoning scenarios. Legal problems typically involve more complex legal concepts, rules, and cases, requiring more specialized knowledge. After retrieving relevant legal knowledge, effectively using it to solve specific legal issues is not an easy task. However, the amount of training data in legal scenarios is relatively limited, making it a key challenge to explore and provide efficient methods to equip LLMs with legal reasoning capabilities. This is the focus

Table 2: Performance comparison between SyLeR and the baseline on the Layperson and Practitioner datasets, with the best performance highlighted in bold and the second-best method underlined.

Category	Models	Practitioner					Layperson				
		Rouge-1	Rouge-2	Rouge-L	BLEU	BERT	Rouge-1	Rouge-2	Rouge-L	BLEU	BERT
Legal LLM	zhihai	21.89	6.19	16.30	5.13	64.56	14.65	1.72	11.00	2.74	59.64
	fuzi.mingcha	28.94	9.51	22.61	8.39	66.94	23.42	4.83	16.50	5.99	65.72
	DISC-LawLLM	23.07	7.28	18.21	6.14	62.55	13.23	2.12	9.80	3.12	58.00
	LawGPT_zh	29.50	9.14	22.93	7.85	67.57	22.67	4.12	16.12	4.85	65.02
	Tailing	27.93	9.58	23.50	5.37	65.89	21.09	4.09	16.65	4.26	62.94
	LexiLaw	29.90	9.27	24.07	7.09	65.71	<u>24.62</u>	5.03	18.27	<u>6.25</u>	<u>65.71</u>
	Hanfei	33.20	12.39	27.06	9.68	68.74	23.94	4.74	<u>18.51</u>	5.63	65.31
Qwen2 (Qwen2-7B-Instruct)	Standard RAG	29.97	10.19	22.82	9.08	68.77	14.34	2.62	10.54	2.58	58.72
	Selective-context	30.02	10.76	23.43	9.59	66.61	9.89	1.13	7.63	1.31	52.27
	LongLLMLingua	32.15	10.65	24.62	10.31	69.52	18.78	3.33	13.80	4.01	62.54
	Iter-Retgen	29.65	9.88	22.61	8.69	68.47	13.87	2.47	10.12	3.17	58.88
	Naïve SFT	32.79	<u>13.43</u>	26.61	10.33	69.70	22.42	<u>5.92</u>	17.60	3.73	64.65
	RAG-SFT	<u>33.70</u>	13.01	26.39	<u>11.41</u>	<u>69.90</u>	19.90	4.07	14.16	5.58	64.30
	CoT-SFT	32.58	11.64	<u>27.09</u>	7.87	<u>67.76</u>	21.65	5.06	17.48	4.80	64.75
	SyLeR (Ours)	36.01	13.83	29.56	11.81	70.55	27.19	7.05	21.11	9.05	67.90

Table 3: Ablation experiments of SyLeR on the Layperson and Practitioner datasets. w/o Path Reward indicates the removal of the reward that ensures the reasoning path. w/o Tree Retrieve means replacing the tree retrieve with a regular retrieve. w/o RL refers to the removal of the RL training that explores multiple syllogistic reasoning paths. The best performance is indicated in bold.

Model	Rouge-1	Rouge-2	Rouge-L	BLEU	BERT
Practitioner					
SyLeR	36.01	13.83	29.56	11.81	70.55
w/o Path reward	34.60	13.06	27.52	11.45	69.83
w/o Tree Retrieve	35.46	13.06	29.03	9.03	69.33
w/o RL	34.24	13.03	27.18	10.70	69.48
Layperson					
SyLeR	27.19	7.05	21.11	9.05	67.90
w/o Path reward	26.50	6.25	19.76	9.00	67.77
w/o Tree Retrieve	26.56	7.01	20.32	7.48	66.80
w/o RL	25.79	6.44	19.00	7.27	66.76

of this paper. SyLeR offers an efficient solution by exploring explicit syllogistic reasoning paths through reinforcement learning, effectively utilizing legal knowledge and enhancing the model’s performance in complex legal reasoning tasks.

4.3 Ablation Study (RQ2)

SyLeR primarily consists of three key components: reasoning path validation, tree-based retrieval of legal statutes and precedent cases, and RL-based exploration of multiple reasoning paths. To explore how these three components impact the performance of SyLeR, we conducted ablation experiments on the Layperson and Practitioner datasets by removing each component from the overall method. The results, as shown in Table 3, are analyzed in detail below:

w/o Path Reward: We removed the reward that ensures the reasoning path is valid, specifically eliminating the rewards for the major and minor premises while keeping only the reward for the conclusion during RL training. The results show a decline in both datasets, indicating the importance of the path reward. Ensuring that the major premise aligns with the retrieved legal statutes and precedent cases and that the minor premise aligns with the user’s question, helps stabilize the model’s reasoning process and guides the model toward more accurate answers.

w/o Tree Retrieve: In this experiment, we replaced the tree-based retrieval for retrieving legal statutes and precedent cases with individual searches for each component (i.e., separately retrieving legal statutes and precedent cases). Other training settings remained unchanged. The results show a noticeable decline, emphasizing the importance of tree-based retrieval. The tree-based retrieval provides a more systematic and structured set of legal knowledge, helping the LLM better understand and solve legal problems by retrieving more relevant documents. This approach also aids in enhancing the model’s reasoning abilities.

w/o RL: SyLeR samples multiple syllogistic reasoning paths during the RL training stage to fully train the LLM. In this ablation, we removed this stage and only tested the model trained after the warm-up stage. This allows us to assess the role of RL in SyLeR. The results show a significant drop in performance without RL, indicating the importance of sampling multiple reasoning paths. Simply relying on the warm-up stage’s SFT does not equip the LLM with strong reasoning capabilities. RL is crucial in enhancing the model’s ability to generate diverse reasoning paths, improving its overall performance in legal reasoning tasks.

These ablation experiments demonstrate that each component plays a vital role in SyLeR, and removing any of them leads to a noticeable decrease in performance, underscoring their importance in enhancing the model’s reasoning abilities.

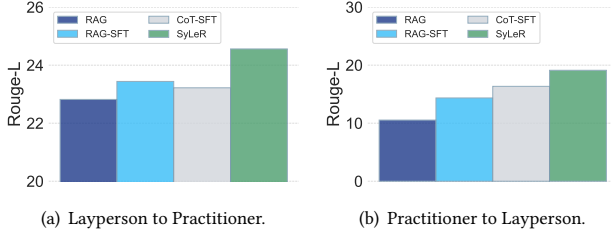


Figure 3: Results of the cross-domain experiments. (a) Applying the model trained on the Layperson dataset to test on the Practitioner dataset. (b) Applying the model trained on the Practitioner dataset to test on the Layperson dataset.

Table 4: Comparison of SyLeR and baselines on the French QA dataset LLeQA. The best performance is indicated in bold.

Model	Rouge-1	Rouge-2	Rouge-L	BLEU	BERT
Standard RAG	23.40	4.58	21.50	2.15	68.49
Naive SFT	27.81	8.05	26.33	2.86	69.62
RAG-SFT	27.75	8.21	26.33	3.09	69.30
SyLeR	29.04	8.54	27.49	3.61	70.16

4.4 Cross Domain Experiment (RQ3)

The cross-domain performance is crucial for legal tasks, where legal knowledge and reasoning skills must be applied across a variety of case types and user profiles. The main experiment has validated the effectiveness of our method within the target domain. To further assess the generalization and robustness of SyLeR, we conduct cross-domain experiments in this section. Specifically, we apply the model trained on the Layperson dataset to the Practitioner dataset and vice versa, testing how well the model performs when transferred across domains.

As shown in Figure 3, the results indicate that SyLeR achieves optimal performance on both datasets in the cross-domain setting. This demonstrates the strong generalization ability of our method, suggesting that SyLeR not only performs well within a specific domain but can also effectively transfer its reasoning capabilities to a different but related domain. Furthermore, the results indicate that SyLeR’s tree retrieval of legal statutes and precedent cases, along with its two-stage reinforcement fine-tuning, contribute to its robustness in adapting to different contexts, reinforcing the model’s utility in diverse legal scenarios. This confirms that SyLeR’s methodology is not domain-specific but has the potential to be applied to a wide range of legal reasoning tasks, further enhancing its practical value in real-world applications.

4.5 Robustness across Various Languages (RQ4)

The previous experiments mainly focused on evaluating the performance of our method on Chinese legal tasks. In this section, we explore the performance of SyLeR on the French dataset LLeQA. The LLeQA dataset’s document corpus only includes legal statutes, so in this experiment, we omitted the tree-based retrieve module in SyLeR and replaced the retriever with BM25 [26] to retrieve the

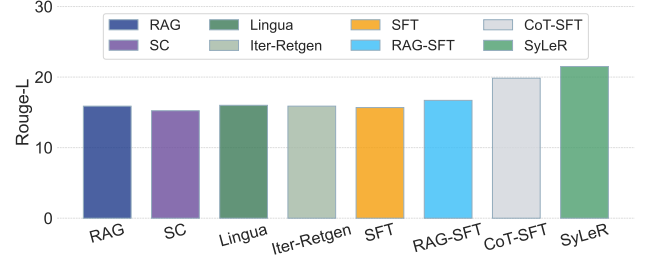


Figure 4: Comparison of the performance between SyLeR and the baselines on the Layperson dataset when the LLM backbone is Llama3. SC is short for Selective-context method. Lingua is short for LongLLMLingua.

Table 5: Human evaluation of the response performance, with each item scored from 1 to 5, with 5 being the highest. The best performance is indicated in bold.

Model	Correct	Logic	Explain	Trust	Average
Naive SFT	4.18	3.76	3.37	3.69	3.75
RAG-SFT	4.21	3.97	3.83	3.88	3.97
CoT-SFT	4.41	4.26	4.12	4.23	4.26
SyLeR	4.65	4.59	4.66	4.72	4.66

most relevant legal statute for the given question. Given that in LLeQA, typically only one related legal statute is needed for a given question to supplement the LLM’s knowledge, we retrieve only the most relevant legal statute to enhance the LLM’s legal knowledge in this experiment. We compare SyLeR with several other methods, including standard RAG, Naive SFT, and RAG-SFT to assess the performance of different approaches on the French legal QA task.

The experimental results are shown in Table 4. It can be observed that our method still achieves the best performance on the LLeQA dataset. This indicates the effectiveness of SyLeR. Despite the differences in task types between the LLeQA dataset and the Chinese dataset, SyLeR demonstrates exceptional performance on the French legal task, proving its cross-lingual generalization capability. Moreover, even after omitting the tree-based retrieve module and using BM25 for retrieval, our model still maintains excellent performance. This suggests that SyLeR performs well not only in complex tree-search retrieval tasks but also with simpler retrieval strategies. Through the comparison with baseline methods, we can see that although these baseline methods can provide some degree of reasoning support, SyLeR’s multi-path reasoning enables it to generate more precise and reliable reasoning when handling French legal tasks.

4.6 Different LLM Backbone (RQ5)

In the main experiment, we used Qwen, an LLM with strong performance on Chinese tasks, as the representative model. To further test the performance of our method on other LLMs, in this section, we replace the LLM backbone with Llama3 [1] (Llama3-8B-Instruct) and compare SyLeR with standard RAG, methods for enhancing the reasoning capabilities of LLMs based on prompting and fine-tuning

<p>Question</p> <p>I have been in the company for three years, the company has not helped me buy social security, now I quit, can I sue the company to pay the social insurance premium for three years?</p>	<p>CoT-SFT</p> <p>Reasoning: According to relevant regulations, employers should pay social insurance for employees...</p> <p>Conclusion: If the company does not pay social security, the workers ask for social security payment...</p>
<p>Naive SFT</p> <p>You can ask the unit to pay back....</p>	<p>SyLeR</p> <p>Major premise: According to Article 58 of the Social Insurance Law of the People's Republic of China, the employer ... shall pay or make up the time limit.</p> <p>Minor premise: The user mentioned that ..., asking ...</p> <p>Conclusion: ...you can complain or negotiate to solve; If the consultation fails, it may apply for arbitration according to law...</p>
<p>RAG-SFT</p> <p>According to the regulations, If your company fails to pay your social security contributions while you are working, you have the right to ask your company to make back payments...</p>	

Figure 5: An example of the responses from SyLeR and baselines.

in Layperson dataset. As shown in Figure 4, SyLeR still achieves optimal performance, highlighting its effectiveness across different LLM architectures. This result demonstrates that SyLeR is not limited to a specific LLM, and it can be successfully applied to a wide range of LLMs, including those with different underlying architectures like Llama3. The consistent performance across models further validates the robustness of SyLeR in enhancing reasoning capabilities, irrespective of the LLM backbone. It also underscores the importance of our explicit reasoning approach in improving the model’s ability to handle complex legal tasks, even when the backbone LLM may differ in structure.

4.7 Trustworthiness Verification (RQ6)

To explore the trustworthiness of the model’s responses, we conducted a human evaluation in this section. We randomly selected 50 questions from the Layperson dataset and tested the following models: SyLeR, Naive SFT, RAG-SFT, and CoT-SFT. We hired three graduate students majoring in Chinese law to perform the evaluation. After anonymizing the methods, we provided them with multiple 4-tuples (question, retrieved document, true answer, model’s response) and asked them to score each response from 1 to 5 on the following aspects, with 5 being the highest: 1) Correctness (Correct): Does the model’s response fully address the key points of the question and closely match the true answer? 2) Logicity (Logic): Does the model’s response exhibit a clear and reasonable reasoning process? 3) Explainability (Explain): Does the model’s response reasonably apply the content of the retrieved document to provide an explanation? 4) Trustworthiness (Trust): Does the model’s response inspire enough trust, including the use of legal knowledge and cases to demonstrate a high level of professionalism and leave the user with a sense of trustworthiness? We averaged the scores from the evaluators, and the results are shown in Table 5. As can be seen, SyLeR achieved the best performance in all four aspects, indicating that SyLeR not only provides accurate and reasonable answers when handling legal questions but also excels in the explainability of the reasoning process and the trustworthiness of the legal knowledge. SyLeR more effectively combines the retrieved legal documents with the questions, demonstrating stronger reasoning ability and professionalism.

Furthermore, Figure 5 shows an example of the model’s response. As seen, SyLeR starts with the legal statutes and performs reasonable legal reasoning in response to the user’s question, integrating relevant legal knowledge to generate an answer. Through a clear reasoning process and accurate citation of the legal statutes, SyLeR not only provides an answer that aligns with legal logic but also enhances the user’s trust in the response.

5 Conclusion

In this paper, we introduced SyLeR, a novel framework designed to equip LLMs with the capability to perform explicit syllogistic legal reasoning. By integrating a tree-structured hierarchical retrieval mechanism, SyLeR effectively combines legal statutes and precedent cases to construct comprehensive and logically connected major premises. Through a two-stage fine-tuning process, SyLeR progressively enhances the model’s reasoning capabilities. The first stage, supervised warm-up, establishes a foundational understanding of syllogistic reasoning by training the model to generate structured reasoning paths. The second stage leverages reinforcement learning with a structure-aware reward mechanism to refine the model’s outputs further. This mechanism not only encourages the exploration of diverse reasoning paths but also ensures logical consistency and strict adherence to the syllogistic reasoning format. Extensive experiments demonstrate that SyLeR consistently improves response accuracy while delivering and trustworthy outputs across different user groups, languages, and LLM backbones. Cross-domain evaluations further validate SyLeR’s adaptability and robustness in diverse legal scenarios, highlighting its potential for real-world legal applications.

Acknowledgments

This work was funded by the National Key R&D Program of China (2023YFA1008704), the National Natural Science Foundation of China (62472426). Supported by fund for building world-class universities (disciplines) of Renmin University of China. Work partially done at Beijing Key Laboratory of Research on Large Models and Intelligent Governance, and Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE. Supported by the Beijing Social Science Foundation Planning Project (Grant No. 24GLC041), the Fundamental Research Funds for the Central Universities in UIBE (Grant No. 24QN06, 24PYTS22).

GenAI Usage Disclosure

I know that the ACM's Authorship Policy requires full disclosure of all use of generative AI tools in all stages of the research (including the code and data) and the writing. No GenAI tools were used in any stage of the research, nor in the writing.

References

- [1] AI@Meta. 2024. Llama 3 Model Card. (2024). https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md
- [2] Jonathan H Choi, Kristin E Hickman, Amy B Monahan, and Daniel Schwarcz. 2021. ChatGPT goes to law school. *J. Legal Educ.* 71 (2021), 387.
- [3] 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*. 13997–14009.
- [4] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [5] Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Songyang Zhang, Kai Chen, Zongwen Shen, and Jidong Ge. 2023. Lawbench: Benchmarking legal knowledge of large language models. *arXiv preprint arXiv:2309.16289* (2023).
- [6] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023).
- [7] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).
- [8] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. [n. d.]. Revisiting Self-Training for Neural Sequence Generation. In *International Conference on Learning Representations*.
- [9] Wanwei He, Jiabao Wen, Lei Zhang, Hao Cheng, Bowen Qin, Yunshui Li, Feng Jiang, Junying Chen, Benyou Wang, and Min Yang. 2023. HanFei-1.0. <https://github.com/siat-nlp/HanFei>.
- [10] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).
- [11] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276* (2024).
- [12] Cong Jiang and Xiaolei Yang. 2023. Legal syllogism prompting: Teaching large language models for legal judgment prediction. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*. 417–421.
- [13] Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *arXiv preprint arXiv:2310.06839* (2023).
- [14] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.
- [15] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.
- [16] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.
- [17] Haitao Li, You Chen, Qingyao Ai, Yueyue Wu, Ruizhe Zhang, and Yiqun Liu. 2024. LexEval: A Comprehensive Chinese Legal Benchmark for Evaluating Large Language Models. *arXiv:2409.20288 [cs.CL]* <https://arxiv.org/abs/2409.20288>
- [18] Haitao Li, You Chen, Qingyao Ai, WU Yueyue, Ruizhe Zhang, and LIU Yiqun. [n. d.]. LexEval: A Comprehensive Chinese Legal Benchmark for Evaluating Large Language Models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- [19] Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing Context to Enhance Inference Efficiency of Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 6342–6353.
- [20] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.
- [21] Hongcheng Liu, Yusheng Liao, Yutong Meng, and Yuhao Wang. 2023. XieZhi: Chinese Law Large Language Model. https://github.com/LiuHC0428/LAW_GPT.
- [22] Antoine Louis, Gijs van Dijk, and Gerasimos Spanakis. 2024. Interpretable long-form legal question answering with retrieval-augmented large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 22266–22275.
- [23] Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. Left: Reasoning with reinforced fine-tuning. *arXiv preprint arXiv:2401.08967* (2024).
- [24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.
- [25] N Reimers. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084* (2019).
- [26] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gattford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp 109* (1995), 109.
- [27] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
- [28] Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal* 27, 3 (1948), 379–423.
- [29] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 9248–9274.
- [30] Kashun Shum, Shizhe Diao, and Tong Zhang. 2023. Automatic Prompt Augmentation and Selection with Chain-of-Thought from Labeled Data. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 12113–12139.
- [31] Dietrich Trautmann, Alina Petrova, and Frank Schilder. 2022. Legal prompt engineering for multilingual legal judgement prediction. *arXiv preprint arXiv:2212.02199* (2022).
- [32] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.
- [33] Shiguang Wu, Zhongkun Liu, Zhen Zhang, Zheng Chen, Wentao Deng, Wenhao Zhang, Jiyuan Yang, Zhitao Yao, Yougang Lyu, Xin Xin, Shen Gao, Pengjie Ren, Zhaochun Ren, and Zhumin Chen. 2023. *fuzi.mingcha*.
- [34] Yiquan Wu, Yuhang Liu, Yifei Liu, Ang Li, Siying Zhou, and Kun Kuang. [n. d.]. *wisdomInterrogatory*. <https://github.com/zhilaiLLM/wisdomInterrogatory> Available at GitHub.
- [35] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. *arXiv:2309.07597 [cs.CL]*
- [36] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuxiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671* (2024).
- [37] Fangyi Yu, Lee Quarrey, and Frank Schilder. 2022. Legal prompting: Teaching a language model to think like a lawyer. *arXiv preprint arXiv:2212.01326* (2022).
- [38] Shengbin Yue, Wei Chen, Siyuan Yang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, and Zhongyu Wei. 2023. DISC-LawLLM: Fine-tuning Large Language Models for Intelligent Legal Services. *arXiv:2309.11325 [cs.CL]*
- [39] Di Zhang, Jianbo Wu, Jingdi Lei, Tong Che, Jiatong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang, Marco Pavone, Yuqiang Li, et al. 2024. Llama-berry: Pairwise optimization for o1-like olympiad-level mathematical reasoning. *arXiv preprint arXiv:2410.02884* (2024).
- [40] Kepu Zhang, Guofu Xie, Weijie Yu, Mingyue Xu, Xu Tang, Yaxin Li, and Jun Xu. 2025. Legal Mathematical Reasoning with LLMs: Procedural Alignment through Two-Stage Reinforcement Learning. *arXiv preprint arXiv:2504.02590* (2025).
- [41] Kepu Zhang, Haoyue Yang, Xu Tang, Weijie Yu, and Jun Xu. 2024. Beyond guilt: Legal judgment prediction with trichotomous reasoning. *arXiv preprint arXiv:2412.14588* (2024).
- [42] Kepu Zhang, Weijie Yu, Sunhao Dai, and Jun Xu. 2024. CitaLaw: Enhancing LLM with Citations in Legal Domain. *arXiv preprint arXiv:2412.14556* (2024).
- [43] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675* (2019).
- [44] Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131* (2024).
- [45] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. [n. d.]. Automatic Chain of Thought Prompting in Large Language Models. In *The Eleventh International Conference on Learning Representations*.

- [46] Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, et al. 2023. Secrets of rlhf in large language models part i: Ppo. *arXiv preprint arXiv:2307.04964* (2023).
- [47] Tianyang Zhong, Zhengliang Liu, Yi Pan, Yutong Zhang, Yifan Zhou, Shizhe Liang, Zihao Wu, Yanjun Lyu, Peng Shu, Xiaowei Yu, et al. 2024. Evaluation of openai o1: Opportunities and challenges of agi. *arXiv preprint arXiv:2409.18486* (2024).