

When Rules Learn: A Self-Evolving Agent for Legal Case Retrieval

Mingxu Tao^{1,2}, Jiawei Hu^{1,2,3}, Xian Zhou^{1,2}, Wenpeng Hu^{1,2}, Jiajun Cheng^{1,2}
Yunbo Cao^{1,2}^{*}, Zhunchen Luo^{1,2}^{*}, Guotong Geng^{1,2}

¹Center of Information Research, AMS

²Discipline and Technology Research Center for Large Model Intelligence Applications

³Hebei University of Engineering

{thomas.mx.tao, zhunchenluo}@gmail.com, caoyunbo@hotmail.com

Abstract

Legal case retrieval remains challenging due to the complexity of legal language and the need for precise lexical alignment between queries and relevant cases. Although dense retrieval models have achieved notable progress, empirical studies show that BM25 continues to serve as a strong baseline in this domain. It motivates us to propose a self-evolving framework for rule-driven query rewriting that enhances BM25 without any parameter training. The framework equips an LLM-based agent with an automatic evaluation environment, enabling it to iteratively **create rewriting rules**, **plan validation experiments** over rule combinations, and **eliminate ineffective rules** based on historical feedbacks. We evaluate our method on the Chinese legal case retrieval benchmark LeCaRD-v2. Experimental results demonstrate that the proposed framework outperforms non-evolutionary baselines, including human-designed rules and greedy rule selection, particularly when powered by a high-capacity core LLM. We also conduct detailed analyses to investigate the mechanisms underlying self-evolution. Our findings reveal that LLM’s capabilities to leverage previous experimental results and its intrinsic knowledge of rule elimination play critical roles in refining the rule set via self-evolution.

1 Introduction

Legal case retrieval plays a vital role in supporting judicial decision-making, legal consultation, and a wide range of downstream legal applications. Given a natural-language description of disputes, retrieval systems are expected to identify relevant prior cases that share similar facts, legal issues, or applicable statutes. This task remains challenging due to the complexity of legal language, the length of legal documents, and the requirement for precise matching of legal facts, statutes, and judicial reasoning.

^{*}Corresponding author.

Despite the rapid progress of dense retrieval methods (Hu et al., 2024; Su et al., 2024) based on neural embeddings (Chen et al., 2024; Li et al., 2023; Tang et al., 2025), their effectiveness in legal case retrieval remains limited. Previous works (Rosa et al., 2021; Deng et al., 2024) reveal that lexical matching methods continue to serve as strong and competitive baselines in the legal case retrieval task. Our empirical study on LeCaRD-v2 (Li et al., 2024) also shows that BM25 (Robertson and Zaragoza, 2009) consistently outperforms several representative dense retrieval models. These findings highlight that, rather than replacing BM25 with dense retrievers, a more promising direction is to enhance BM25 by lexically aligning queries with the relevant documents.

Building upon the strengths of BM25, we focus on improving retrieval performance through **query rewriting**. Query rewriting aims to bridge the gap between queries and relevant legal cases by enriching queries with legal terminology, synonymous expressions, or alternative formulations. Recent progress in large language models (LLMs) (Yang et al., 2025; OpenAI et al., 2025; DeepSeek-AI et al., 2025; Gemma Team et al., 2025) make it feasible to perform rule-driven query rewriting, where an LLM rewrites queries by following explicitly rewriting rules. Such rules offer interpretability and controllability, which are particularly desirable in the legal domain. However, designing high-quality rewriting rules typically requires substantial domain expertise, while naive rule generation may lead to suboptimal performance.

To address these limitations, we propose a **self-evolution framework** for the rule-driven query rewriting. The framework equips an LLM-based agent with an automated validation environment that enables iterative optimization of the rule set without any parameter updates. Specifically, the agent can autonomously (1) create new query-rewriting rules, (2) plan validation experiments

by selecting combinations of rules, and (3) eliminates ineffective rules based on experimental history. Through repeated interaction with the environment, the agent continuously refines its rule set, improving the effectiveness of retrieval in a training-free manner.

We evaluate the proposed framework on the LeCaRD-v2 benchmark (Li et al., 2024). Experimental results show that the self-evolution framework can outperform the non-evolutionary baselines, including human-designed rules and the greedy strategy for rule selection. We also conduct further analyses to investigate the mechanisms underlying self-evolution, revealing that the effectiveness of self-evolution depends on the LLM’s ability to plan new experiments by leveraging historical results, as well as its capability to determine appropriate timing for rule elimination.

In summary, our contributions are as follows:

- We revisit legal case retrieval, demonstrating that BM25 still remains a strong baseline, and enhance it with rule-driven query rewriting strategies.
- We propose a novel self-evolution framework, which enables the agent to iteratively optimize rewriting rules through rule generation, experiment planning, and rule elimination, without any gradient-based training.
- We conduct extensive investigations to characterize the behavioral dynamics of self-evolving LLM agents, shedding light on the intrinsic factors affecting agent’s capacity for self-evolution.

2 Related Works

Prior work on autonomous or self-evolving agent systems has recently garnered increasing attention, particularly in contexts requiring adaptive behavior and iterative improvement.

Several studies have explored mechanisms by which agents can refine their internal strategies without heavy human supervision. For instance, AgentEvolver (Zhai et al., 2025) introduces self-questioning, experience-guided exploration, and fine-grained credit attribution to enable agents to generate their own tasks, reuse experiences, and improve exploration efficiency in complex environments. The process can reduce reliance on handcrafted datasets and fixed reinforcement learning pipelines, extending broader paradigms of self-

evolving agent architectures. EvolveSearch (Zhang et al., 2025a) proposes iterative self-evolving search agents. The system refines its retrieval behavior via ongoing self-improvement loops, indicating that iterative evolution paradigms can yield measurable gains for information search tasks. However, these methods depend on reinforcement learning to optimize the model’s action selection strategy. They face challenges when applied to the legal case retrieval task due to limited training data.

Recent work (Suzgun et al., 2025; Zhang et al., 2025b) also focuses on evolving the contextual information inputted into LLMs. They treat the contexts in prompts as dynamic entities that can be incrementally refined. The previous method, Agentic Context Engineering (Zhang et al., 2025b), proposes to generate, reflect on, and select contextual elements in a *playbook* memory. These methods emphasize evolving inputs and contextual representations to improve downstream performance, which resonates with our framework of evolving rule sets for query rewriting rather than training model parameters. However, our work explicitly evolves a set of structural rules, which serve as the instructions for downstream query rewriting, rather than the agent’s intrinsic prompts.

3 Preliminaries

Legal case retrieval poses unique challenges due to the complexity of legal wordings, long document lengths, and the need for precise lexical matching of legal facts, statutes, and judicial reasoning. Despite the rapid advancement of dense retrieval methods based on neural embeddings, their effectiveness in legal case retrieval remains limited. Previous work (Rosa et al., 2021) argues that the BM25 function (Robertson and Zaragoza, 2009) continues to serve as a strong baseline.

We compare BM25 with several representative dense retrieval models on the held-out test set of LeCaRD-v2 (Li et al., 2024), a Chinese legal case retrieval benchmark. We consider three models: bge-m3 (Chen et al., 2024), SAILER_zh (Li et al., 2023), and ReaKase-8B (Tang et al., 2025). Among these models, bge-m3 is a general embedding model for dense retrieval, while the other two are embedding models that have been continually trained on legal-domain data.

Table 1 presents the results under different cut-offs. BM25 achieves the highest average recall across different scales of retrieved candidates, out-

Method	\mathcal{L}	Recall@k					Avg.
		k=50	100	200	500	1000	
BM25	✗	38.29	48.79	58.89	70.14	77.32	58.69
bge-m3	✗	27.88	37.05	47.51	63.33	74.45	50.04
SAILER	✓	19.22	28.28	41.18	63.45	78.93	46.21
ReaKase	✓	30.69	42.05	54.82	73.32	84.47	57.07

Table 1: Recall scores for BM25 and dense retrieval based on distinct embedding models on LeCaRD-v2. \mathcal{L} denotes whether the model is trained on legal data.

performing both general and legal-domain embedding models. Although ReaKase-8B and SAILER_zh demonstrate competitive performance in Recall@1000, their effectiveness remains significantly inferior to BM25 when limiting the amount of retrieved candidates. Compared with BM25, the two models exhibit performance drops of 7.60%~19.07% on Recall@50 and 6.74%~20.51% on Recall@100.

The strong performance of BM25 motivates us to enhance it through query rewriting rather than replacing it with dense retrieval models. Query rewriting aims to bridge the gap between user’s queries and relevant legal cases by enriching or reformulating queries with legal terms. We leverage a reasoning LLM to follow given rules and rewrite queries. Figure 1 shows an example of the rules. See the prompt of query rewriting in Appendix B.

The original rule in Chinese:

同义词和术语变体扩展：对于每个关键法律概念，添加其同义词、近义词或常见变体。
格式：[术语1]、[术语2]、[术语3]、.....

The translated rule in English:

Synonym and Terminology Variant Expansion: For each key legal concept, augment the query by incorporating its synonyms, semantically related terms, or commonly used variants.

Format: [Term A], [Term B], [Term C], ...

Figure 1: An example of query-rewriting rules.

4 Self-evolution Framework

We propose a self-evolution framework that enables an LLM-based agent to autonomously discover, examine, and refine query-rewriting rules for legal case retrieval. The framework is a closed-loop agent-environment system, where adaptation emerges from iterative interaction, rather than gradient-based optimization. Figure 2 illustrates

the self-evolution framework.

4.1 Action Decision

The agent selects actions to drive the self-evolution process. The action space consists of three categories: (1) **creating a new rule**, (2) **planning an experiment**, and (3) **eliminating an ineffective rule**.

At the interaction step t , the agent selects an action based on its internal memories, which comprises a sequence of recent actions $A^{(t)} = \langle a_{t-k}, \dots, a_{t-1} \rangle$ truncated by a fixed length k , the current set of rules $R^{(t)}$, and the accumulated experimental results $S^{(t)}$. We denote the action selected at time t as:

$$a_t = \pi(A^{(t)}, R^{(t)}, S^{(t)}),$$

where π is the decision policy that is based solely on the prompt in Appendix A.1. The policy operates without any parameter updates during the evolution process.

4.2 Rule Generation

Creating a new rule enables the agent to expand the search space of query-rewriting strategies. At step t , the agent generates a candidate rule by jointly analysing the current active rule set $R^{(t)}$, the set of eliminated rules $\bar{R}^{(t)}$, and the accumulated experimental scores $S^{(t)}$. This analysis focuses on identifying which rewriting operations contribute positively to recall improvements and diagnosing failure patterns exhibited by removed rules. The agent then generate single new rule:

$$r_n = f_{\text{create_rule}}(R^{(t)}, \bar{R}^{(t)}, S^{(t)}),$$

where n equals to $|R^{(t)}| + |\bar{R}^{(t)}| + 1$.

The agent may generate a new rule through two mechanisms. First, it can edit an existing rule by modifying its description of rewriting operation, with the goal of amplifying observed effective behaviors. Second, the agent can propose a novel rule that introduces a completely different rewriting strategy not present in $R^{(t)}$. To prevent degeneration and repeated exploration of unproductive patterns, we require that the generated rule should not be similar to the eliminated ones in $\bar{R}^{(t)}$.

4.3 Experiment Planning

Planning an experiment allows the agent to assess the effectiveness of different combinations of rules for query rewriting. At step t , the agent selects a

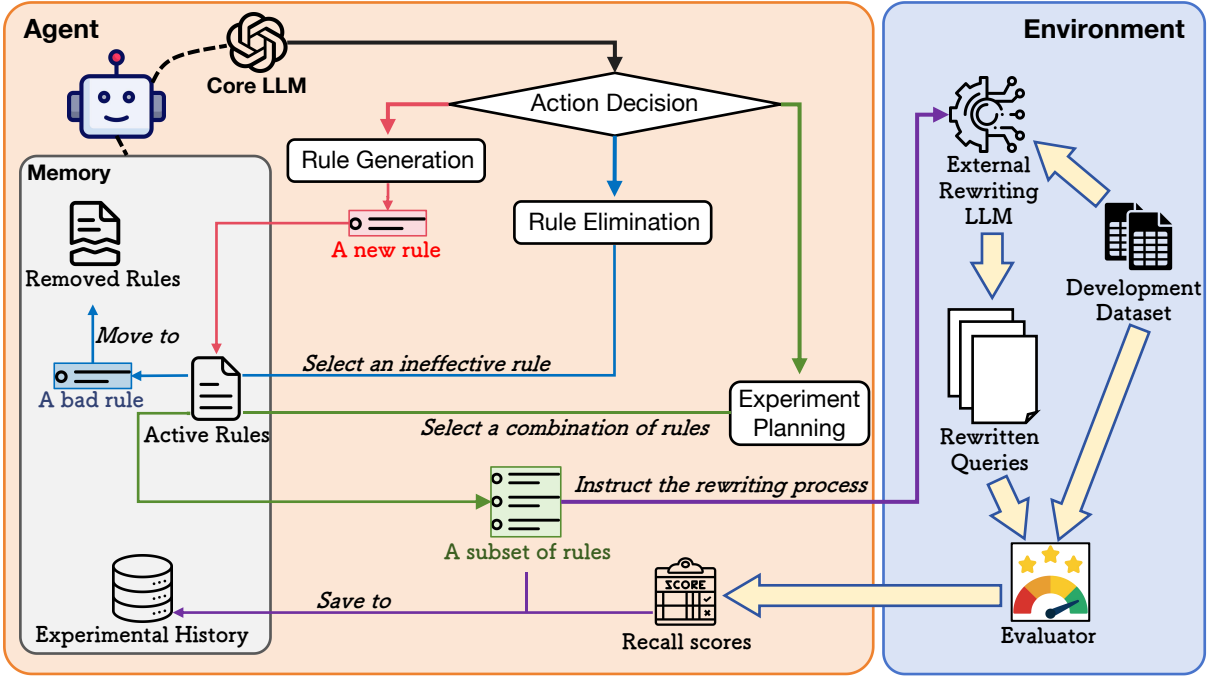


Figure 2: Self-evolution based on rule generation, experiment planning, and rule elimination.

subset of active rules from the current rule set $R^{(t)}$, denoted as $C^{(t)} \subseteq R^{(t)}$. The environment receives these rules and rewrites the query by invoking an external model. It then feeds back the test results of rewritten queries $s_{C^{(t)}}$ to agent.

We employ the experiment history $S^{(t)}$ to guide the selection of rule combinations. Based on this history, the agent estimates which combinations are likely to yield further recall improvements, such as complementary effects. To ensure efficient exploration, the agent is restricted from selecting any combination that has already been evaluated in prior experiments.

We denote the experiment planning as:

$$C^{(t)} = f_{\text{plan_exp}}(R^{(t)}, S^{(t)}).$$

The agent then appends the experimental result $\langle C^{(t)}, s_{C^{(t)}} \rangle$ to $S^{(t)}$.

4.4 Rule Elimination

Eliminating ineffective rules plays a vital role to control the growth of the strategy space, preventing the agent from exploring unproductive combinations of rules. However, a rule may bring more improvements in part of strategies but not in others. The premature removal of potentially effective rules can hinder long-term performance. To mitigate this risk, we adopt a two-stage mechanism with self-consistency (Wang et al., 2023).

At step t , the agent first determines whether the rule set $R^{(t)}$ contains any rule that should be eliminated. This decision is repeated independently for n_1 trials based on the same memories $\langle R^{(t)}, S^{(t)} \rangle$. If the agent concludes that at least one rule should be removed in more than $\delta_1 \times n_1$ times, the elimination process proceeds to the second stage; otherwise, no rule is removed at this step.

In the second stage, the agent identifies a specific rule for removal. It selects a candidate rule from $R^{(t)}$ over n_2 independent trials, again conditioned on $\langle R^{(t)}, S^{(t)} \rangle$. A rule $r \in R^{(t)}$ can be eliminated if the agent selects it in more than $\delta_2 \times n_2$ times. The removed rule is transferred from the active rule set $R^{(t)}$ to the eliminated rule set $\bar{R}^{(t)}$, and is excluded from subsequent experiment planning.

We formulate the elimination process as:

$$r_{\text{elim}} = \begin{cases} \arg \max_{r \in R^{(t)}} \text{Count}(r), & \text{if } m > \delta_2 \times n_2, \\ \emptyset, & \text{otherwise,} \end{cases}$$

where m equals to $\max_{r \in R^{(t)}} \text{Count}(r)$. We use $\text{Count}(r)$ to denote the number of times in which the rule r is selected for removal across n_2 trials. We believe this two-stage consensus mechanism can mitigate the risk of discarding effective rules due to stochastic.

4.5 Interactive Environment

The environment serves as an automatic test bed that executes and evaluates the agent’s decisions. The environment can invoke a specific LLM for query rewriting, whose parameters are **frozen** and does not perform any optimization.

If the agent plans to experiment at step t , it yields a subset of rules $C^{(t)} \subseteq R^{(t)}$ to the environment. The environment employs the external LLM to rewrite queries according to the selected rules. It subsequently evaluates the effectiveness of these refined queries for legal case retrieval, providing the recall scores $s_{C^{(t)}}$ back to the agent.

5 Experiments

5.1 Experimental Setup

Benchmark We evaluate the self-evolution framework on the Chinese legal case retrieval benchmark LeCaRD-v2 (Li et al., 2024), which contains 800 queries and a retrieval pool with 55,192 documents. We partition the dataset into two sets: a development set of 100 instances and a test set of 700 instances.

Baselines We compare our framework against three baseline methods to assess the effectiveness of self-evolution.

(1) **Human-written rules.** We manually design three rules based on expert knowledge in legal case retrieval, and apply them jointly to query rewriting.

(2) **LLM-generated rules without evolution (LLMGEN).** We directly employ the core LLM of the agent to generate multiple rules in a single pass, with a human-written rule as exemplar. We adopt the combination of generated rules and the human-written rule for rewriting.

(3) **Greedy Strategy (GREEDY).** Based on the rules generated in Method (2) across multiple runs, we examine each individual rule on the development set. We select three top-performing rules as a group for rewriting.

Implementation Details In the environment, we employ Qwen3-4B-Thinking (Yang et al., 2025) to rewrite queries in all experiments. We concatenate the rewritten query with the original query, and then use the BM25 algorithm for retrieval. We set k_1 to 1.2 and b to 0.75.

In the self-evolution process, we utilise four LLMs as the agent’s core model, respectively, including Qwen3-4B-Thinking, Qwen3-30B-A3B-Thinking (Yang et al., 2025),

gpt-oss-20b, and gpt-oss-120b (OpenAI et al., 2025). Following the baseline methods, we initially introduce one human-written rule as a seed to the agent’s active rule set R . To ensure self-consistency in rule elimination, we set n_1 and n_2 to 7, δ_1 and δ_2 to 0.5. We conduct the self-evolution process entirely on the development set. The agent explores for up to 500 steps. We then select the subset of rules with the best performance from experiment history S and evaluate its effectiveness on the held-out test set.

For each setting, we conduct five independent runs. Given the inherent stochasticity of LLM-based query rewriting, we perform the rewriting process five times and report the average performance. See more details in Appendix C.

5.2 Main Results

Table 2 reports the retrieval recalls on the LeCaRD-v2 held-out test set. We include BM25 as a vanilla retrieval baseline, along with two heuristic baselines: HUMAN-1, which applies single human-provided rule for query rewriting, and HUMAN-3, which combines three human-written rules. Across all cutoff k , HUMAN-3 consistently outperforms BM25 and HUMAN-1, indicating that a combination of high-quality rules with expert knowledge can effectively improve legal case retrieval.

Beyond heuristic baselines, we evaluate the proposed self-evolution framework using different core LLMs for the agent and compare it against two non-evolutionary baselines introduced in Section 5.1. These baselines represent strong alternatives that exploit LLM generation or empirical rule selection without iterative evolution.

When using smaller models with less than 20B parameters as the agent’s core LLM, the self-evolution framework does not yield consistent improvements, particularly at lower cutoffs. Self-evolution performs worse than GREEDY by Recall@50 of 0.05%~0.43% and Recall@100 of 0.08%~0.26%. The results suggest that smaller models may lack sufficient reasoning capacity to balance performance across multiple recall thresholds. And instead, they may over-optimize for metrics such as Recall@500 and Recall@1000, which are easier to improve during evolution.

However, when we employ gpt-oss-120b as the agent’s core LLM, the self-evolution framework consistently outperforms all baseline methods across all recall cutoffs. It surpasses the greedy strategy by +0.39%~+1.04% on the recall scores, as

Method	Recall@k					Avg.
	k=50	k=100	k=200	k=500	k=1000	
<i>Heuristic Baselines</i>						
BM25	38.29	48.79	58.89	70.14	77.32	58.69
HUMAN-1	38.64 \pm 0.17	49.37 \pm 0.18	59.77 \pm 0.16	71.74 \pm 0.13	79.07 \pm 0.19	59.72
HUMAN-3	38.76 \pm 0.03	49.91 \pm 0.10	60.40 \pm 0.15	72.91 \pm 0.16	80.63 \pm 0.18	60.52
<i>Core Model: Qwen3-4B-Thinking</i>						
LLMGEN	38.41 \pm 0.11	49.41 \pm 0.23	59.83 \pm 0.35	72.05 \pm 0.56	79.68 \pm 0.76	59.88
GREEDY	38.64 \pm 0.01	49.61 \pm 0.07	60.05 \pm 0.14	71.97 \pm 0.12	79.46 \pm 0.13	59.95
EVOLVE (Ours)	38.59 \pm 0.26	49.53 \pm 0.35	60.07 \pm 0.41	72.62 \pm 0.49	80.66 \pm 0.74	60.29
<i>Core Model: Qwen3-30B-A3B-Thinking</i>						
LLMGEN	38.40 \pm 0.30	49.49 \pm 0.32	59.94 \pm 0.31	72.17 \pm 0.61	79.72 \pm 0.73	59.95
GREEDY	38.53 \pm 0.19	49.87 \pm 0.22	60.52 \pm 0.01	73.05 \pm 0.13	80.88 \pm 0.11	60.57
EVOLVE (Ours)	38.38 \pm 0.19	49.61 \pm 0.20	60.38 \pm 0.42	73.39 \pm 0.31	81.75 \pm 0.41	60.70
<i>Core Model: gpt-oss-20b</i>						
LLMGEN	38.04 \pm 0.46	49.17 \pm 0.45	59.93 \pm 0.41	72.65 \pm 0.22	80.70 \pm 0.41	60.10
GREEDY	39.42 \pm 0.30	50.40 \pm 0.28	61.16 \pm 0.14	73.55 \pm 0.26	81.47 \pm 0.20	61.20
EVOLVE (Ours)	38.99 \pm 0.17	50.25 \pm 0.20	61.47 \pm 0.16	74.04 \pm 0.33	81.96 \pm 0.42	61.34
<i>Core Model: gpt-oss-120b</i>						
LLMGEN	38.65 \pm 0.53	49.68 \pm 0.54	60.24 \pm 0.51	72.58 \pm 0.58	80.45 \pm 0.74	60.32
GREEDY	39.33 \pm 0.28	50.13 \pm 0.23	60.81 \pm 0.29	73.19 \pm 0.16	81.25 \pm 0.15	60.94
EVOLVE (Ours)	39.69 \pm 0.34	50.64 \pm 0.33	61.22 \pm 0.16	73.90 \pm 0.06	82.29 \pm 0.17	61.55

Table 2: Recall scores on the held-out test set of LeCaRD-v2. We report the standard errors across five runs and five times of repeatedly query rewriting.

well as an average recall improvement of +0.61%.

Comparing the performance of EVOLVE and HUMAN-3, we further find that when using gpt-oss-20b and gpt-oss-120b as the core LLMs, the rule combinations explored by the agent outperform the manually designed rules on all recall metrics, with average improvements of +0.82% and +1.03%, respectively. These two large LLMs may have strong capabilities, enabling them to follow the procedures defined in the self-evolution framework. They may generate better rules and explore more effective rule combinations.

6 Discussions

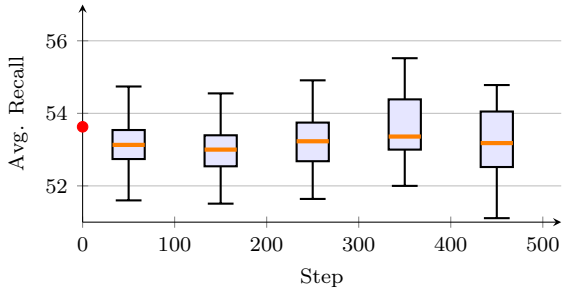
In this section, we dive into the underlying mechanisms that drive self-evolution. We investigate three research questions: (1) how the quality of individual rules evolves during exploration; (2) how the capacity of the core LLM influences experimental planning when exploiting previously validated rule combinations; and (3) whether the agent can reliably perform conservative, evidence-driven rule

elimination. Through these analyses, we aim to shed light on the behavioral characteristics and limitations of self-evolving LLM agents beyond aggregate retrieval metrics.

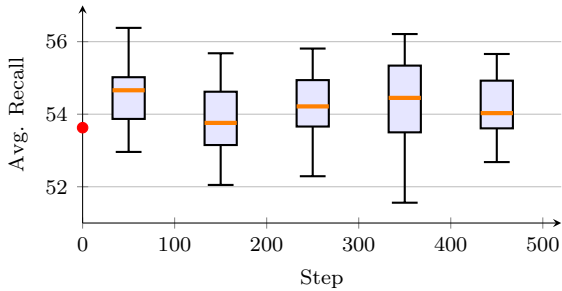
6.1 RQ1: Evolution of Rule Quality?

The success of GREEDY motivates us to examine whether the agent can propose better rules as the self-evolution process conducts. We analyze the performance of each individual rule and record the time of its creation. For each run, the agent evolves for 500 steps, and we partition the process into five phases of 100 steps each.

We study the highest- and lowest-performing self-evolving agents for analysis, which employ gpt-oss-120b and Qwen3-4B-Thinking as the core LLM, respectively. Figure 3 illustrates the distribution of rule performance within each phase by boxplot. Across the two agents, we find that they fail to consistently generate increasingly effective rules over time. Instead, at each phase of evolution, the agent produces a mixture of both effective and ineffective rules. Both the median and the maxi-



(a) Rules created by Qwen3-4B-Thinking.



(b) Rules created by gpt-oss-120b.

Figure 3: Performance on dev set of individual rules created by self-evolving agents at distinct phases. The red dot at $t = 0$ shows the performance of a human-written rule, which is provided as the seed rule for agents.

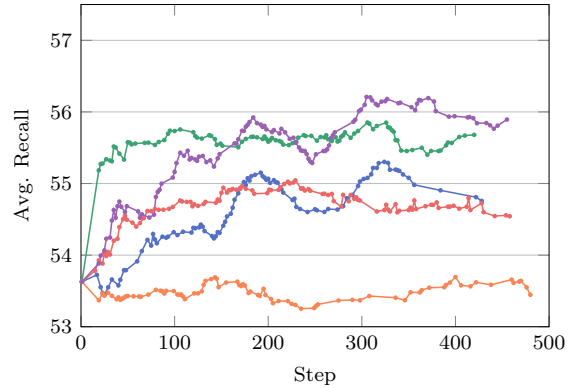
imum recall of newly created rules fluctuate across the five phases, without monotonicity.

The results suggest that the self-evolution framework does not primarily rely on progressive improvement of new rules. The process of rule generation may introduce both promising and unproductive rules. We thus highlight the importance to study downstream mechanisms, such as how the agents with distinct core LLMs plan experiments and eliminate ineffective rules.

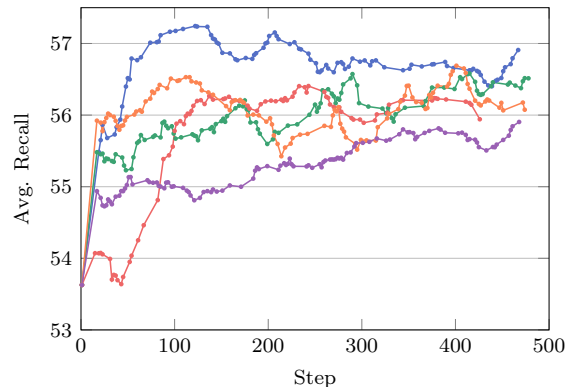
6.2 RQ2: Effect of Core LLM Capacity on Experimental Planning

We investigate how the agents plan experiments by exploiting previously validated rule combinations. Different from rule generation, experiment planning requires the agent to reason over the results in $S^{(t)}$, identify promising combinations, and design new experiments that extend previous successes rather than exploring unproductive regions of the search space.

To analyze this process, we examine the chain-of-thought texts (Wei et al., 2022) generated by the core LLM when the agent plans each new experiment. We find that the agent frequently justifies its choice of a new rule combination by explicitly



(a) Using Qwen3-4B-Thinking as the core LLM.



(b) Using gpt-oss-120b as the core LLM.

Figure 4: Performance of the rule combinations mentioned by the agent when planning new experiments. We illustrate the curves for five runs respectively.

mentioning one or more previously validated combinations. The new experiment is typically framed as incremental modifications of these prior combinations, such as adding or removing one rule. See more case studies in Appendix D.

We extract the validated combinations referenced by the agents when planning experiments. We then compare these *anchor* combinations and record the one with the best performance at each step. We track the recall scores of referenced combinations over time in the evolution process of Qwen3-4B-Thinking and gpt-oss-120b.

Figure 4 shows the performances of anchor combinations in five runs. We find in several runs of Qwen3-4B-Thinking, the agent fails to select progressively better combinations of rules as references. Specifically, in the fifth run (orange curve in Figure 4a), the agent continuously plans new experiments based on those previous combinations worse than the seed rule. However, the results in Figure 3a shows that Qwen3-4B-Thinking is ca-

pable to generate superior individual rules than the seed. It indicates that this model’s reasoning capabilities may be insufficient to leverage the successful results in the history $S^{(t)}$ for the design of new experiments. As a result, the agent struggles to explore combinations with better performance and yields suboptimal recalls to the greedy strategy.

In contrast, across all five runs, `gpt-oss-120b` can select stronger validated combinations as the basis for new experiments, as it self-evolves on more steps. We also find that, across multiple runs, `gpt-oss-120b` can achieve more convergent recall scores at the final stage compared to other LLMs. We believe that the capabilities to maintain and exploit high-quality planning anchors is a key factor underlying why the agent using `gpt-oss-120b` is the only one that consistently outperforms the greedy strategy across all recall cutoffs.

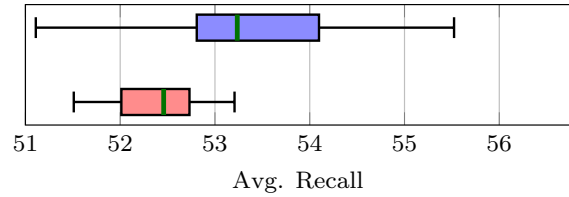
6.3 RQ3: Faithfulness in Rule Elimination

Rule elimination is a critical component of the self-evolution framework, since premature removal of potentially effective rules may constrain the search space, while permissive retention may lead to ineffective exploration.

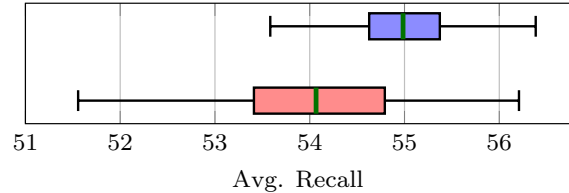
We examine which rules are retained or discarded in the evolution process. Figure 5 shows the performance distributions of the retained rules and the eliminated rules after self-evolution. Compared with the discarded rules, the retained rules have higher median and higher maximum recalls. The self-evolving agents are capable of discerning the quality of rules and attempt to eliminate ineffective ones. However, we find that `gpt-oss-120b` may eliminate several effective rules, which almost achieve the best performance of the retained rules.

We further examine the number of rules retained at the end of evolution. When using `Qwen3-4B-Thinking` as the core LLM, the agent retains an average of 37.8 ± 3.9 rules, although the prompts in Appendix A.4 instruct it to keep the scale of active rules not significantly larger than 6 rules. Consequently, the search space for rule combinations grows exponentially, preventing `Qwen3-4B-Thinking` from identifying effective combinations. `gpt-oss-120b` can consistently control the number of rules in memory $R^{(t)}$ at 5.8 ± 1.0 rules, but it sometimes discards a potentially effective rule without sufficient experiments.

When exposed to lengthy evolving principles, current LLMs usually overemphasize a limited subset of these principles. For instance,



(a) Rules created by Qwen3-4B-Thinking.



(b) Rules created by `gpt-oss-120b`.

Figure 5: Distributions of the retained rules and the eliminated rules at the end of self-evolution. We employ blue boxes to present the retained rules, while red boxes for the eliminated rules.

`gpt-oss-120b` tends to focus on control the number of active rules, which `Qwen3-4B-Thinking` concentrates more on not eliminating the potentially promising rules. As a result, during the self-evolution process, even under explicit and detailed decision-making constraints, the agents still fail to align with human expert decisions.

7 Conclusion

We presented a self-evolution framework that enables an agent to refine query-rewriting rules for legal case retrieval in a training-free setting. By interacting with an automated evaluation environment, the agent iteratively generates new rules, plans experiments to assess rule combinations, and eliminates ineffective strategies.

Experimental results on the LeCaRD-v2 benchmark demonstrate that the proposed framework, when employing a capable core LLM such as `gpt-oss-120b`, can consistently outperform non-evolutionary baselines. Our work highlights the potential of self-evolving agents for interpretable and adaptive enhancement for legal case retrieval.

8 Limitations

Legal case retrieval is a critical task that supports real-world practices such as judicial decision-making and legal consultation. It has been widely adopted across jurisdictions that differ in language and legal tradition. Since the structure of case doc-

uments, the terminology, and other characteristics may be different across judicial systems, we note that when applying the proposed method to other legal case retrieval benchmarks, it would be better to provide the agent with appropriate seed rules at the initialization stage of self-evolution.

We also note that the core LLM’s capabilities to follow instructions play a vital role in self-evolution. We have tried to employ non-reasoning models to serve as the core LLM, for instance, Qwen3-30B-A3B-Instruct (Yang et al., 2025). However, this LLM cannot understand the format constraints in the prompts in Appendix A, merely generating meaningless rules. For instance, in the description of the rewriting process, the model fails to provide any descriptive texts but merely enumerates several semantically redundant keywords, as shown in Figure 6. Thus, we employ four reasoning LLMs in this work, rather than further exploring whether there are non-reasoning LLMs which can serve as the core model in self-evolution.

The original rule in Chinese:
[诉讼时效届满]、[时效经过]、[超过诉讼时效]、[时效已过]、[时效失效]、[诉讼时效已过]
格式：[术语1]、[术语2]、[术语3]、[术语4]、[术语5]、[术语6]

The translated rule in English:
[expiration of the statute of limitations], [lapse of the limitation period], [exceeding the statute of limitations], [limitation period has passed], [limitation invalid], [statute of limitations has passed]
Format: [Term A], [Term B], [Term C], [Term D], [Term E], [Term F]

Figure 6: An example of the meaningless rule generated by Qwen3-30B-A3B-Instruct.

We also note that the core LLM’s capabilities to understand and to generate Chinese texts are important. We have tried to employ Magistral-Small-2509 (Mistral-AI et al., 2025), a multi-lingual reasoning LLM with 24 billion parameters, as the core model. However, the model usually provides rewriting rules with language-switching and fragmented phrases, as shown in Figure 7. We do not further investigate how LLM’s language-specific capabilities affect its performance in self-evolving. Instead, we empirically employ the four LLMs with promising capabilities to understand Chinese instructions and generate Chinese texts.

The original rule in Chinese:
扩展性语义域规则：将原始理解后的转换等效性基准，以同义复述成定义标准化。若需区分初中类公式，可覆盖近Uri和底层
格式：[拓符ca कान][共现][规范整合][metaJesus shakes 모습이 的合规变体]

The translated rule in English:
Extensibility semantic-domain rule: Transform the equivalence baseline obtained from the original interpretation into a standardized definition through synonymous paraphrasing. **If need to distinguish Middle School Kind Formula, it can cover Near Uri and Bottom**
Format: [Printmaking Symbol ca कान], [Co-occurrence], [Standard Combination], [metaJesus shakes 모습이 's Legal Variants]

Figure 7: An example of the fragmented rule generated by Magistral-Small-2509.

Acknowledgments

This work is supported in part by Self-Determined Project of the Discipline and Technology Research Center for Large Model Intelligence Applications. We thank the anonymous reviewers for their helpful discussions and suggestions.

References

- Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. **M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation.** In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. **Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.** *Preprint*, arXiv:2501.12948.
- Chenlong Deng, Zhicheng Dou, Yujia Zhou, Peitian Zhang, and Kelong Mao. 2024. **An element is worth a thousand words: Enhancing legal case retrieval by incorporating legal elements.** In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2354–2365, Bangkok, Thailand. Association for Computational Linguistics.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey

- Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. [Gemma 3 technical report](#). *Preprint*, arXiv:2503.19786.
- Yutong Hu, Kangcheng Luo, and Yansong Feng. 2024. [ELLA: Empowering LLMs for interpretable, accurate and informative legal advice](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 374–387, Bangkok, Thailand. Association for Computational Linguistics.
- Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. [Sailer: Structure-aware pre-trained language model for legal case retrieval](#). In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23*, page 1035–1044, New York, NY, USA. Association for Computing Machinery.
- Haitao Li, Yunqiu Shao, Yueyue Wu, Qingyao Ai, Yixiao Ma, and Yiqun Liu. 2024. [Lecardv2: A large-scale chinese legal case retrieval dataset](#). In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '24*, page 2251–2260, New York, NY, USA. Association for Computing Machinery.
- Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. [Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations](#). In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362.
- Mistral-AI, Abhinav Rastogi, Albert Q. Jiang, Andy Lo, Gabrielle Berrada, Guillaume Lample, Jason Rute, Joep Barmantlo, Karmesh Yadav, Kartik Khandelwal, Khyathi Raghavi Chandu, Léonard Blier, Lucile Saulnier, Matthieu Dinot, Maxime Darrin, Neha Gupta, Roman Soletskyi, Sagar Vaze, Teven Le Scao, and 81 others. 2025. [Magistral](#). *Preprint*, arXiv:2506.10910.
- OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, and 107 others. 2025. [gpt-oss-120b & gpt-oss-20b model card](#). *Preprint*, arXiv:2508.10925.
- Stephen Robertson and Hugo Zaragoza. 2009. [The probabilistic relevance framework: Bm25 and beyond](#). *Found. Trends Inf. Retr.*, 3(4):347–369.
- Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto Lotufo, and Rodrigo Nogueira. 2021. [Yes, bm25 is a strong baseline for legal case retrieval](#). *Preprint*, arXiv:2105.05686.
- Weihang Su, Qingyao Ai, Yueyue Wu, Yixiao Ma, Haitao Li, Yiqun Liu, Zhijing Wu, and Min Zhang. 2024. [Caseformer: Pre-training for legal case retrieval based on inter-case distinctions](#). *Preprint*, arXiv:2311.00333.
- Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. 2025. [Dynamic cheat-sheet: Test-time learning with adaptive memory](#). *Preprint*, arXiv:2504.07952.
- Yanran Tang, Ruihong Qiu, Xue Li, and Zi Huang. 2025. [Reakase-8b: Legal case retrieval via knowledge and reasoning representations with llms](#). *Preprint*, arXiv:2510.26178.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. [Chain of thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Yunpeng Zhai, Shuchang Tao, Cheng Chen, Anni Zou, Ziqian Chen, Qingxu Fu, Shinji Mai, Li Yu, Jiaji Deng, Zouying Cao, Zhaoyang Liu, Bolin Ding, and Jingren Zhou. 2025. [Agentevolver: Towards efficient self-evolving agent system](#). *Preprint*, arXiv:2511.10395.
- Ding-Chu Zhang, Yida Zhao, Jialong Wu, Liwen Zhang, Baixuan Li, Wenbiao Yin, Yong Jiang, Yu-Feng Li, Kewei Tu, Pengjun Xie, and Fei Huang. 2025a. [EvolveSearch: An iterative self-evolving search agent](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13134–13147, Suzhou, China. Association for Computational Linguistics.
- Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. 2025b. [Agentic context engineering: Evolving contexts for self-improving language models](#). *Preprint*, arXiv:2510.04618.

A Prompts Employed in the Self-Evolution Framework

This section details the prompts employed in our self-evolution framework, including those for action decision, rule generation, experiment planning, and the elimination of ineffective rules.

At each step t of the self-evolution process, the agent need to make a decision based on its memory, i.e., the active rule set $R^{(t)}$, the eliminated rule set $\bar{R}^{(t)}$ and the experimental history $S^{(t)}$. We use the mark $\{\text{rules}\}$ to denote the context representing $R^{(t)}$, while the mark $\{\text{elim_rules}\}$ for $\bar{R}^{(t)}$ and the mark $\{\text{scores}\}$ for $S^{(t)}$.

We also provide the performance of vanilla BM25 to the agent, whose context is inserted into the slot $\{\text{baseline_scores}\}$.

A.1 Prompts of Action Decision

We present the Chinese prompt in Appendix A.1.1 and the corresponding English translated prompt in Appendix A.1.2.

A.1.1 The Original Chinese Prompt of Action Decision

The original prompt of action decision

你是一名资深的法律专家。你的任务是构造高效的查询改写规则集合，提升法律文档检索的效果。你需要不断调整现有的检索查询改写规则，在此过程中，你可以采取**引入新的规则**、**开展实验验证规则的有效性**、**剔除低效规则**等方式。现在，你需要对接下来的工作进行规划，决定下一步应该优先采取哪一种方式。

现有的查询改写规则:

$\{\text{rules}\}$

未进行改写时的结果:

$\{\text{baseline_score}\}$

利用已有改写规则的排列组合开展实验的结果:

$\{\text{scores}\}$

决策历史:

$\{\text{action_history}\}$

要求:

在决策前，你可以参考至多向前10步（即T-10时刻，当前为T-0时刻）的决策历史，并从中吸取教训，避免频繁失败。请你结合现有的查询改写规则及上述实验结果，选择下一步的动作。你可以选择**开展实验验证**、**剔除低效规则**、**创造新的规则**三种方式中的一种。

低效规则是指：根据该规则进行查询改写之后，检索的结果不仅没有得到改进，其召回率反而低于不使用该规则的实验。请注意，在判断某条规则是否为低效规则时，你应当非常小心，尽管引入某条规则在部分实验中可能导致检索性能下降，但在另一些实验中，引入这些规则可能有助于提高检索的召回率，这些规则不应被判定为低效规则。同时，请你尽量将现有的查询改写规则控制在6条之内，如果现有规则超过上限，此时某条改写规则A在单独使用或与任意其他规则组合时带来的提升均小于另一条改写规则B，则也可以将改写规则A视作低效规则。

你在选择下一步动作时，应遵循如下原则:

- 如果现有规则的排列组合还未得到充分验证，此时应当选取某一条规则或者某几条规则的集合，测试按照这些规则改写查询语的效果，你应优先选择**开展实验验证**;

- 如果你认为存在部分规则明显是低效的、会导致改写后的检索效果变差，你可以选择**剔除低效规则**;

- 如果基于现有规则的排列组合基本都已得到验证，且没有明显的低效规则需要剔除，你可以选择**创造新的规则**。

请你按照下列**模板**的形式，先简要说明为什么选择该方式作为下一步的动作，然后明确地输出该动作的名称，你必须从三个备选动作中选取下一步的动作，且只能选取一个。再次强调，请你严格按照**模板**中的形式分段输出，并且严禁提供其他内容或者做任何额外说明。

模板:

分析:

{{一段文本，简要说明为什么选取该动作}}

下一步动作:

{{开展实验验证/剔除低效规则/创造新的规则}}

A.1.2 The Translated Prompt of Action Decision

The translated prompt of action decision

You are a senior legal expert. Your task is to construct an efficient set of query rewriting rules to improve the effectiveness of legal document retrieval. You need to continuously adjust the existing query rewriting rules. In this process, you may adopt actions such as **introducing a new rule**, **conducting experimental validation of rule effectiveness**, and **removing an ineffective Rule**. Now, you need to decide which action should be prioritized next.

Active Query Rewriting Rules:

`#{rules}`

Baseline Recall Scores without Rewriting:

`#{baseline_score}`

Experimental Results of the Combination of Existing Rules:

`#{scores}`

Action History:

`#{action_history}`

Instructions:

Before making a decision, you may refer to up to the previous ten steps of decision history (i.e., from time step T-10, with the current step being T-0) and draw lessons from past outcomes to avoid repeated failures. Based on the existing query-rewriting rules and the experimental results described above, please select the next action. You may choose one of the following options: **conduct experimental validation**, **eliminate an ineffective rule**, or **create a new rule**.

Definition of Ineffective Rule: When applying this rule to rewrite a query, the retrieval results are not improved; instead, the recall scores are lower than that of experiments in which the rule is not used. Please note that when determining whether a rule is ineffective, you should be very cautious. Although introducing a rule may result in degraded retrieval performance in some experiments, the same rule may help improve retrieval recall in other experiments; such rules should not be classified as ineffective rules. Meanwhile, please try to keep the total number of existing query rewriting rules within six. If the number of existing rules exceeds this limit, you can regard a rule *A* as an ineffective rule, if the improvement brought by rule *A* is consistently smaller than that brought by another rule *B*, when used alone or in combination with any other rules.

When choosing the next action, you should adhere to the following principles.

- If the combinations of the existing rules have not been sufficiently validated, you should prioritize to **conduct experimental validation**. Specifically, you can select an individual rule or a subset of rules and evaluate the effectiveness of query rewriting under these rules.

- If certain rules are deemed to be evidently ineffective and are likely to degrade retrieval performance after query rewriting, they may be excluded to improve overall effectiveness, and you can choose to **eliminate an ineffective rule**.

- If the combinations of the existing rules have largely been validated and no clearly ineffective rules remain to

be eliminated, you can choose to **create a new rule** to further enhance the rewriting strategy.

Please follow the format of the **template** below. First, briefly explain why this action is chosen for the next step, and then explicitly output the name of that action. You must select the next action from three candidate actions, and you can select only one. Emphasize again that you must strictly follow the format in the **template** for segmented output, and you are strictly prohibited from providing any other content or making any additional explanations.

Template:

Analysis:

{{A segment of text that briefly explains why this action is selected.}}

The Next Action:

{{Conduct Experimental Validation / Eliminate an Ineffective Rule / Create a New Rule}}

A.2 Prompts of Rule Generation

We present the Chinese prompt in Appendix A.2.1 and the corresponding English translated prompt in Appendix A.2.2.

A.2.1 The Original Chinese Prompt of Rule Generation

The original prompt of rule generation

你是一名资深的法律专家。你的任务是根据已有的查询改写规则，设计出一个新的查询改写规则，以提升法律文档检索的效果。

已被证实改写效果较差并被剔除的改写规则:

`#{elim_rules}`

现有的查询改写规则:

`#{rules}`

未进行改写时的结果:

`#{baseline_score}`

利用已有改写规则的排列组合开展实验的结果:

`#{scores}`

要求:

请你结合现有的查询改写规则及上述实验结果，设计出一个新的查询改写规则。你需要检视那些被剔除的规则，它们很有可能无法带来有效的提升，你在创造新规则时，应避免给出与这些规则较为相似的新规则。你可以对现有规则进行修改，也可以提出全新的思路。请确保新规则能够有效提升法律文档的检索效果。

请注意，在执行检索时，会自动将改写后的查询与原始查询进行拼接，因此你设计的新查询规则中**禁止**重复输出原始查询。

请你只提供一条新的规则，形式仿照**现有的查询改写规则**，你需要按照下列**模板**的形式，先简要说明该规则的设计思路和预期效果，再明确写出该规则的名称、具体内容、输出格式。再次强调，请你严格按照**模板**中的形式分段输出，并且严禁提供其他内容或者做任何额外说明。

模板:

****设计思路**:**

{{一段文本，简要说明该规则的设计思路}}

****预期效果**:**

{{一段文本，说明该规则预期提升的检索效果}}

****新规则内容**:**

{{规则名称}}: {{一句话或几句话，描述具体的改写方法}}

格式: {{一行文本，顿号间隔的一个或多个使用半角方括号"[]"做标记的格式类型，描述输出文本的格式}}

A.2.2 The Translated Prompt of Rule Generation

The translated prompt of rule generation

You are a senior legal expert. Your task is to design a new rule for query rewriting based on the existing ones, in order to improve the effectiveness of legal document retrieval.

The eliminated rules that have been proved to be ineffective:

`#{elim_rules}`

Active Query Rewriting Rules:

`#{rules}`

Baseline Recall Scores without Rewriting:

`#{baseline_score}`

Experimental Results of the Combination of Existing Rules:

`#{scores}`

Instructions:

Please design a new query rewriting rule based on the existing rules and the experimental results provided above. You need to examine those rules that have been discarded, as they are likely unable to bring effective improvements. When creating new rules, you should avoid proposing rules that are highly similar to these discarded ones. You may modify existing rules or propose entirely new ideas. Please ensure that the new rule can effectively improve the retrieval performance of legal documents.

Please note that during retrieval, the rewritten query will be automatically concatenated with the original query. Therefore, you are ****prohibited**** to provide a new rule using the repeated inclusion of the original query.

Please provide only one new rule, following the format of the ****active query rewriting rules****. You need to first briefly explain the design rationale and expected effect of the rule according to the ****template**** below. Then clearly specify the rule's name, detailed content, and output formatting. Emphasize again that you must strictly follow the format in the ****template**** for segmented output, and you are strictly prohibited from providing any other content or making any additional explanations.

Template:

****Design Concept**:**

{{A brief description outlining the design rationale behind this rule.}}

****Expected Outcome**:**

{{A brief description explaining the anticipated improvement in retrieval performance resulting from this rule.}}

****New Rule**:**

{{Rule Name}}: {{one or more sentences describing the process of query rewriting.}}

Formatting: {{A single line of text A single line of text containing one or more format types enclosed in half-width square brackets "[]" and separated by commas, describing the format of the rewritten query.}}

A.3 Prompts of Experiment Planning

We present the Chinese prompt in Appendix A.3.1 and the corresponding English translated prompt in Appendix A.3.2.

A.3.1 The Original Chinese Prompt of Experiment Planning

The original prompt of experiment planning

你是一名资深的法律专家。你的任务是构造高效的查询改写规则集合，提升法律文档检索的效果。你需要不断调整现有的检索查询改写规则，在此过程中，你可能需要开展实验，验证规则组合的有效性。

现有的查询改写规则:

`#{rules}`

未进行改写时的结果:

`#{baseline_score}`

利用已有改写规则的排列组合开展实验的结果:

`#{scores}`

要求:

请你结合现有的查询改写规则及上述实验结果, 决定下一步应利用哪些现有规则开展实验, 以验证其有效性。你需要按照下列**模板**的形式, 先简要说明为什么选择这些规则进行实验验证, 然后明确地输出这些规则的编号列表, 编号之间用逗号分隔。注意, 你可以选择一条规则, 也可以选择多条规则的组合进行实验验证, 但你目前仅能开展一次实验, 你只可以输出一组规则编号列表。再次强调, 请你严格按照**模板**中的形式分段输出, 并且严禁提供其他内容或者做任何额外说明。

模板:

分析:

{{一段文本, 简要说明为什么选取这些规则进行实验}}

下一步实验需使用的规则编号列表:

{{{1/2/3/1,2/1,3/2,3/1,2,3/.....}}}

A.3.2 The Translated Prompt of Experiment Planning

The translated prompt of experiment planning

You are a senior legal expert. Your task is to construct an efficient set of query rewriting rules to improve the effectiveness of legal document retrieval. You need to continuously adjust the existing query rewriting rules. In this process, you may need to conduct experiments to verify the effectiveness of the rule combinations.

Active Query Rewriting Rules:

#{rules}

Baseline Recall Scores without Rewriting:

#{baseline_score}

Experimental Results of the Combination of Existing Rules:

#{scores}

Instructions:

Please select a combination of rules to validate its effectiveness based on the existing rules and the experimental results provided above. You need to follow the template below to briefly explain why these rules are chosen for experimental validation, then clearly output the list of rule numbers, separated by commas. Note that you may select one single rule or one combination of multiple rules for experimental validation. Since you can only conduct one experiment at present, so you may only output one set of rule numbers. Emphasize again that you must strictly follow the format in the **template** for segmented output, and you are strictly prohibited from

providing any other content or making any additional explanations.

Template:

Analysis:

{{A brief explanation of the rationale behind selecting these rules for the experiment.}}

List of rule numbers needed for the next step experiment:

{{{1/2/3/1,2/1,3/2,3/1,2,3/.....}}}

A.4 Prompts of Rule Elimination

In the step of rule elimination, the agent should first determine whether there is an ineffective rule in R . We present the original prompt in Appendix A.4.1 and the translated prompt in Appendix A.4.3.

If the agent confirm the necessity to eliminate a rule, it should then determine which rule should be removed. We present the original prompt in Appendix A.4.2 and the translated prompt in Appendix A.4.4.

A.4.1 The Original Prompts of Determining Whether to Proceed Elimination

The original prompt of determining the necessity of elimination

你是一名资深的法律专家。你的任务是分析已有的查询改写规则中是否存在可能导致检索效果变差的规则。

现有的查询改写规则:

#{rules}

未进行改写时的结果:

#{baseline_score}

利用已有改写规则的排列组合开展实验的结果:

#{scores}

要求:

请你结合现有的查询改写规则及上述实验结果, 分析是否存在某条低效规则。

低效规则是指: 根据该规则进行查询改写之后, 检索的结果不仅没有得到改进, 其召回率反而低于不使用该规则的实验。请注意, 在判断某条规则是否为低效规则时, 你应当非常小心, 尽管引入某条规则在部分实验中可能导致检索性能下降, 但在另一些实验中, 引入这些规则可能有助于提高检索的召回率, 这些规则不应被判定为低效规则。同时, 请你尽量将现有的查询改写规则控制在6条之内, 如果现有规则超过上限, 此时某条改写规则A在单独使用或与任意其他规则组合时带来

的提升均小于另一条改写规则B，则也可以将改写规则A视作低效规则。

请你谨慎的判断是否存在需要被剔除的改写规则，你需要按照下列**模板**的形式，给出明确的理由说明为什么存在被剔除的规则，再下最终的结论，你的结论只能是“存在需要被剔除的规则”或“不存在需要被剔除的规则”二者之一。再次强调，请你严格按照**模板**中的形式分段输出，并且严禁提供其他内容或者做任何额外说明。

模板:
分析:
{{一段文本, 简要说明为什么存在需要被剔除的改写规则}}
结论:
{{存在需要被剔除的规则/不存在需要被剔除的规则}}

A.4.2 The Original Prompt of Selecting a Rule to Be Eliminated

The original prompt of determining which rule to be eliminated

你是一名资深的法律专家。你的任务是分析已有的查询改写规则中，哪条规则的存在最有可能导致检索效果变差，需要优先被剔除。

现有的查询改写规则:
\${rules}
未进行改写时的结果:
\${baseline_score}
利用已有改写规则的排列组合开展实验的结果:
\${scores}
要求:

请你结合现有的查询改写规则及上述实验结果，分析是否存在某条低效规则。

低效规则是指：根据该规则进行查询改写之后，检索的结果不仅没有得到改进，其召回率反而低于不使用该规则的实验。请注意，在判断某条规则是否为低效规则时，你应当非常小心，尽管引入某条规则在部分实验中可能导致检索性能下降，但在另一些实验中，引入这些规则可能有助于提高检索的召回率，这些规则不应被判定为低效规则。同时，请你尽量将现有的查询改写规则控制在6条之内，如果现有规则超过上限，此时某条改写规则A在单独使用或与任意其他规则组合时带来的提升均小于另一条改写规则B，则也可以将改写规则A视作低效规则。

请你选择一条最应该被优先剔除的规则，你需要按照下列**模板**的形式，给出明确的理由说

明为什么根据该条规则改写之后检索的效果会变差，以及为什么这条规则需要先于其他规则被剔除，再明确地输出该规则的编号。注意，你必需输出且仅能输出**1个**待剔除规则的编号，你输出的编号应为纯数字形式，不要输出规则的名称、具体内容、改写格式等信息。再次强调，请你严格按照**模板**中的形式分段输出，并且严禁提供其他内容或者做任何额外说明。

模板:
分析:
{{一段文本, 简要说明为什么存在需要被剔除的改写规则}}
应优先被剔除的规则编号:
{{1/2/3/.....}}

A.4.3 The Translated Prompt of Determining Whether to Proceed Elimination

The translated prompt of determining the necessity of elimination

You are a senior legal expert. Your task is to analyse whether there exists one or more rules which may result in a deteriorated performance after query rewriting.

Active Query Rewriting Rules:
\${rules}
Baseline Recall Scores without Rewriting:
\${baseline_score}
Experimental Results of the Combination of Existing Rules:
\${scores}
Instructions:
Based on the existing query rewriting rules and the experimental results above, please analyse whether any ineffective rule is present.

Definition of Ineffective Rule: When applying this rule to rewrite a query, the retrieval results are not improved; instead, the recall scores are lower than that of experiments in which the rule is not used. Please note that when determining whether a rule is ineffective, you should be very cautious. Although introducing a rule may result in degraded retrieval performance in some experiments, the same rule may help improve retrieval recall in other experiments; such rules should not be classified as ineffective rules. Meanwhile, please try to keep the total number of existing query rewriting rules within six. If the number of existing rules exceeds this limit, you can regard a rule A as an ineffective rule, if the improvement brought by rule A is consistently smaller

than that brought by another rule B, when used alone or in combination with any other rules.

Please carefully judge whether there exist rewriting rules that need to be removed. You need to provide a clear explanation according to the following **template** on why such rules should be removed, and then give the final conclusion. Your conclusion can only be one of the following two: "There exist rewriting rules that need to be removed" or "There do not exist rewriting rules that need to be removed." Emphasize again that you must strictly follow the format in the **template** for segmented output, and you are strictly prohibited from providing any other content or making any additional explanations.

Template:

Analysis:

{{A brief explanation outlining the necessity of eliminating certain rewriting rules.}}

Conclusion:

{{There exist rewriting rules that need to be removed/There do not exist rewriting rules that need to be removed}}

A.4.4 The Translated Prompt of Selecting a Rule to Be Eliminated

The translated prompt of determining which rule to be eliminated

You are a senior legal expert. Your task is to analyse the existing query rewriting rules and identify which rule is most likely to cause a decline in retrieval performance.

Active Query Rewriting Rules:

#{rules}

Baseline Recall Scores without Rewriting:

#{baseline_score}

Experimental Results of the Combination of Existing Rules:

#{scores}

Instructions:

Please analyze whether there exists any ineffective rule, based on the existing query rewriting rules and the experimental results.

Definition of Ineffective Rule: When applying this rule to rewrite a query, the retrieval results are not improved; instead, the recall scores are lower than that of experiments in which the rule is not used. Please note that when determining whether a rule is ineffective, you should be very cautious. Although introducing a rule

may result in degraded retrieval performance in some experiments, the same rule may help improve retrieval recall in other experiments; such rules should not be classified as ineffective Rules. Meanwhile, please try to keep the total number of existing query rewriting rules within six. If the number of existing rules exceeds this limit, you can regard a rule A as an ineffective rule, if the improvement brought by rule A is consistently smaller than that brought by another rule B, when used alone or in combination with any other rules.

Please select the single rule that should be prioritized for removal. Using the **template** below, provide a clear explanation of why rewriting according to this rule leads to worse retrieval performance, and why this rule must be removed before any other. Then explicitly output the rule's number. Note that you must output **exactly one** rule number for removal. The output should be a pure numeric form only—do not include the rule name, specific content, rewriting format, or any other information. Emphasize again that you must strictly follow the format in the **template** for segmented output, and you are strictly prohibited from providing any other content or making any additional explanations.

Template:

Analysis:

{{A brief explanation of why a certain rewriting rule must be removed.}}

A rule number that should be removed with priority:

{{1/2/3/.....}}

B Prompts of Query Rewriting

In this work, we employ Qwen3-4B-Thinking as the rewriting model for all experiments. We use following prompts for query rewriting.

The original prompt of query rewriting

你是一名法律专家，你的最终目标是利用检索器从历史案件数据库中搜集与**待决案件**相似的案件，即待决案件的“类案”。

现在，请你结合下列**修改规则**，基于**待决案件基本事实**进行修改。修改后的文本将作为原有检索查询语的增强，以确保待决案件的类案出现在检索结果中。

修改规则:

#{rules}

待决案件基本事实:

#{query}

要求:

你需要严格遵循**修改规则**，以**待决案件基本事实**为基础，将其修改为更适合作为检索查询语的文本。已知，数据库中存储的是历史案件的全文，包括基本事实、裁判理由、判决结果等内容。检索器采用的是BM25方法。

请注意，你无需重新输出一遍待决案件基本事实的原文。

请你在思考后，针对每一条修改规则，按照该规则的改写策略，并严格遵循规则后附的格式要求，直接输出相应的增强查询语，不要做额外解释。

在输出时，请你以“修改后的检索查询语：”作为总体的开头。之后，每条规则对应的增强查询语依次以数字标号1、2、3……作为起始。请不要提供“修改说明”之类的冗余解释。

The translated prompt of query rewriting

You are a legal expert, and your ultimate goal is to use a retriever to collect cases from a historical case database that are similar to the **pending case**.

Now, please revise the query in accordance with the following rules, based on the basic facts of the pending case. The rewritten query will be used as an enhancement of the original retrieval query, in order to ensure that cases similar to the pending case appear in the retrieval results.

Rewriting Rules:

`#{rules}`

Facts of the Pending Case:

`#{query}`

Instructions:

You need to strictly follow the modification rules and, based on the basic facts of the pending case, modify it into text that is more suitable for use as a retrieval query. It is known that the database stores the full text of historical cases, including basic facts, reasoning of the judgment, judgment results, and other content. The retriever adopts the BM25 method.

Please note that you do not need to reproduce the original text of the basic facts of the pending case.

After thinking carefully, for each rule, rewrite according to the rewriting strategy of that rule, and strictly follow the format requirements attached to the rule, directly output the corresponding enhanced query, without providing any additional explanation.

In the output, begin with the heading “Revised Retrieval Queries:”. Then, present the enhanced query corresponding to each rule sequentially, starting with numerical labels 1, 2, 3, and so on. Do not include any

redundant explanations such as “modification notes.”

C More Implementation Details

Following previous work (Li et al., 2024), we employ Pyserini Lucene (Lin et al., 2021) to implement BM25 retrieval. The judicial document of a case typically comprises three core components: the facts, the legal bases (i.e., statutes and legal reasons), and the judicial decision results.

To align real-world practice of judicial decision, we use the full texts of documents to construct the index, while using only the section of facts as the query during retrieval.

During self-evolution and for all query writing experiments, we do not adjust the hyper-parameters of LLMs for generation. We adopt the default temperatures, top-p, top-k, and other hyper-parameters.

D Case Study for Experiment Planning

We study how the agent plan the next experiment based on historical results via reasoning. Figure 8 shows a case generated by gpt-oss-120b when selecting a rule combination to validate.

We find the agent can explicitly mention the rule combinations which have been validated at previous steps. From the analysis provided by LLM, we can identify two reference combinations, the subset {2, 3} and the subset {1, 2, 3}. The agent finally chooses to examine the subset {1, 2, 3, 6}, we thus denote the mentioned combination with minimum edit distance (i.e., the subset {1, 2, 3}) as the *anchor* experiment for planning.

The original outputs:

****分析**:**

当前实验中，规则组合[2, 3]已取得最高的Recall@1K (79.4509)，且加入规则1后(组合[1, 2, 3])进一步提升至79.6803，显示规则1、2、3的协同增益明显。规则6单独效果一般，但有潜力在已有高效组合上提供额外的检索线索，尚未与[1, 2, 3]组合实验。为验证是否能够进一步突破当前最高召回，计划在已证明有效的[1, 2, 3]基础上加入规则6，进行一次实验。

****下一步实验需使用的规则编号列表**:**

1,2,3,6

The translated outputs:

****Analysis**:**

In the existing experiments, the rule combination [2, 3] achieves one of the highest Recall@1K (79.4509). After incorporating Rule 1 (i.e., combination [1, 2, 3]), the performance is further improved to 79.6803, indicating a clear synergistic gain among Rules 1, 2, and 3. Rule 6 alone shows moderate effectiveness, but it has the potential to provide additional retrieval cues on top of existing high-performing combinations and has not yet been evaluated together with [1, 2, 3]. To verify whether the current best recall can be further improved, we plan to conduct an experiment by adding Rule 6 to the validated combination [1, 2, 3].

****List of rule numbers needed for the next step experiment**:**

1,2,3,6

Figure 8: An case of the reasoning process for experiment planning, with gpt-oss-120b as the core LLM.