SaLSA-RAG: State-and-Law Summary Aligned Retrieval-Augmented Generation for Conversational Legal Advice

Anonymous ACL submission

Abstract

Conversational legal advice must generate grounded answers under evolving multi-turn context, where the key challenge is to retrieve statutes that are legally applicable rather than merely topically similar. Standard retrieval-augmented generation typically relies on a single query view, which can surface lexically plausible yet inapplicable evidence. We propose SaLSA-RAG, a State-and-Law Summary Aligned framework for multi-turn legal consultation. At each turn, SaLSA-RAG builds (i) a history-aware retrieval query from the current utterance and user-only dialogue history, and (ii) a concise legal analysis state that captures parties, salient facts, procedural posture, and the sub-issue to resolve. A dense retriever retrieves candidate statutes, and SaLSA-Reranker aligns the query and induced state with applicability-oriented statute summaries to score and select evidence for generation. On the Chinese LexRAG benchmark, SaLSA-RAG improves downstream answer quality, raising micro keyword recall from 0.286 to 0.370 and the overall LLM-judge score from 5.13 to 5.63. It also improves retrieval quality: with the default dense encoder, test nDCG@10 increases from 0.1143 to 0.1798, and reaches 0.2109 with a stronger embedding model.

1 Introduction

Legal consultation is increasingly mediated by chat-based assistants, where users describe disputes across multiple turns and expect actionable, statute-grounded guidance. Retrieval-augmented generation (RAG) is a natural fit for this setting: retrieved legal evidence can constrain generation and reduce hallucinations while preserving the flexibility of large language models (Lewis et al., 2020; Guu et al., 2020; Izacard and Grave, 2021; Izacard et al., 2022; Borgeaud et al., 2022).

Despite this promise, evidence retrieval for multi-turn legal dialogues remains challenging.

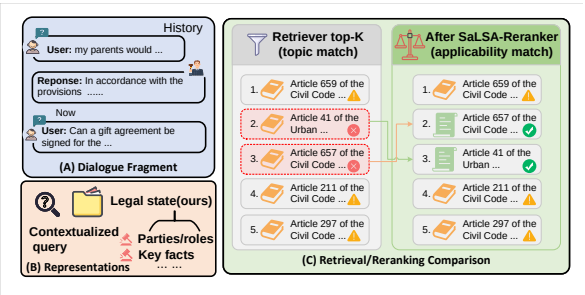

Figure 1: Illustrative example : SaLSA-Reranker uses an induced legal state to rerank the top-K statutes from topical matches toward legally applicable evidence.Figure text is translated from the original Chinese data; experiments use Chinese inputs.

First, user turns are context-dependent and often underspecified; resolving references and implicit constraints is necessary even before retrieval (Dalton et al., 2020; Elgohary et al., 2019). Second, legal relevance is primarily applicability-driven rather than topical: a statute may be lexically similar to a question yet inapplicable because its prerequisites are not met (e.g., missing party standing, procedural posture, or exception clauses). Standard bi-encoder retrievers (Karpukhin et al., 2020; Xiong et al., 2021) and even strong cross-encoder rerankers (Nogueira and Cho, 2019) may struggle to internalize such context- and precondition-dependent signals under limited supervision. Third, legal language differs from colloquial queries in terminology and structure, aggravating the mismatch between user descriptions and formal statutes (Chalkidis et al., 2020, 2022).

We present SaLSA-RAG (State-and-Law Summary Aligned RAG), a state-aware framework for multi-turn legal consultation. The key idea is to introduce a structured semantic view of the user's situation, namely a turn-level legal analysis state, and use it consistently for both retrieval and generation. At each dialogue turn, we (i) construct a history-aware turn representation from the current

utterance and preceding dialogue context, (ii) induce a concise legal analysis state that captures parties and relationships, key facts, procedural posture, and the sub-question to resolve, and (iii) retrieve a small candidate set of statutes with a first-stage retriever (BM25 or dense retrieval). While conversational query reformulation (often called "query rewrite") is a common strategy for improving first-stage retrieval, it is not required by SaLSA-RAG. In our main setting, we retrieve directly from the original turn with history; query reformulation is evaluated as an optional plug-in that can further improve candidate quality when available.

The core of SaLSA-RAG is SaLSA-Reranker, which targets applicability-aware evidence selection. We precompute an applicability-oriented summary for each statute, distilling its operative conditions and legal effects. SaLSA-Reranker then aligns three views, namely the dialogue turn representation, the induced legal analysis state, and the statute summary, to produce calibrated relevance scores over the retrieved candidate pool. The resulting top-k statutes provide both stronger evidence for downstream generation and an interpretable signal about why a statute is applicable under the current situation.

We make the following contributions:

- We propose SaLSA-RAG, a state-aware RAG framework for multi-turn legal consultation that explicitly models turn-level legal analysis states and applicability-oriented statute summaries.
- We introduce SaLSA-Reranker, a lightweight and interpretable applicability-aware reranker that aligns the dialogue state with statute summaries to improve evidence selection without changing the first-stage index or encoder.
- We show consistent gains in both retrieval and end-to-end answer quality on a Chinese conversational legal benchmark, and provide analyses on embedding backbones, feature design, and alternative fusion strategies.

2 Related Work

Retrieval-augmented generation. RAG grounds language model outputs with retrieved evidence (Lewis et al., 2020). Classic lines improve retrieval integration and reader robustness, including latent/differentiable retrieval and strong decoder-side fusion (Guu et al., 2020; Karpukhin et al., 2020; Izacard and Grave, 2021; Izacard et al., 2022; Borgeaud et al., 2022). Our focus is conversational legal advice, where evidence selection under evolving context is the main bottleneck.

Retrieval and reranking. BM25 remains a competitive sparse baseline (Robertson and Zaragoza, 2009). Dense bi-encoders learn semantic matching in a shared embedding space (Karpukhin et al., 2020; Xiong et al., 2021), while late-interaction and sparse-neural approaches offer alternative effectiveness–efficiency trade-offs (Khattab and Zaharia, 2020; Formal et al., 2021). Cross-encoder rerankers can be highly effective but costly (Nogueira and Cho, 2019). We compare to an LLM reranker and show that a lightweight, state-aware scorer can be competitive under the same candidate pools.

Conversational retrieval, rewriting, and fusion. Conversational search highlights the need to incorporate dialogue history (Dalton et al., 2020). Question rewriting is a common way to make turns self-contained (Elgohary et al., 2019), but it may still miss legally salient facets (e.g., preconditions, exceptions, procedural posture). We therefore introduce an induced legal analysis state as an additional view. We also relate to rank fusion and learning-to-rank, which combine signals via heuristics or supervised models (Cormack et al., 2009; Burges et al., 2005, 2006; Wu et al., 2010; Ke et al., 2017); our findings emphasize task-aligned state/summary signals over generic higher-capacity fusion.

Legal NLP and evaluation. Legal language motivates domain-specific models and benchmarks (Chalkidis et al., 2020, 2022). To assess end-to-end assistance, we report both grounding-oriented keyword metrics and LLM-judge evaluation, following common practice in LLM evaluation (Zheng et al., 2023; Liu et al., 2023).

3 Method

We propose SaLSA-RAG, a retrieval-augmented generation approach for multi-turn legal consultation. At each dialogue turn, the system retrieves relevant statutory articles as evidence and then generates a grounded response. SaLSA-RAG is built around two intermediate representations designed for legal applicability: (i) a turn-level legal analysis state that summarizes the legally salient situation described by the user, and (ii) an applicability-

oriented summary for each statute that highlights operative conditions and legal effects. These representations are shared across retrieval and generation: they are used to refine the evidence set from a first-stage retriever and to provide structured grounding for the final answer.

3.1 Task Formulation

We model a consultation as a sequence of user–assistant turns

$$\mathcal{C} = \{(u_1, a_1), \ldots, (u_T, a_T)\},$$

where u_t is the user utterance and a_t is the assistant response at turn t. The knowledge base is a fixed collection of statutory articles

$$\mathcal{L} = \{\ell_1, \ldots, \ell_N\},$$

where each ℓ_i includes an identifier (e.g., statute name and article number) and the full legal text.

We define retrieval and generation at the level of individual turns. For a conversation index c and turn index t, we denote the turn by (c, t), with dialogue history

$$H_{c,t} = \{(u_1, a_1), \ldots, (u_{t-1}, a_{t-1})\}$$

and the current utterance u_t. In the supervised setting, a subset $\mathcal{L}^*_{c,t} \subseteq \mathcal{L}$ is annotated as relevant statutory support for this turn.

At runtime, the system must (i) retrieve a small set of relevant statutes for each (c, t) and (ii) generate a grounded answer conditioned on the conversation and these statutes. The retrieval component outputs a ranked list $\pi_{c,t}$ over $\mathcal{L}$, and the generation component produces an answer $\hat{a}_{c,t}$ conditioned on $H_{c,t}$, u_t, and the top-ranked statutes from $\pi_{c,t}$. This section describes the modeling of these components; concrete model choices and evaluation metrics are provided in Section 4.

Turn query representation. User turns are often colloquial and context-dependent, so we construct a turn-level query view for retrieval. We denote this query view as $q_{c,t}$. In our main setting, $q_{c,t}$ is derived directly from the current utterance together with dialogue history (e.g., by formatting or concatenating key context from $(H_{c,t}, u_t)$), without requiring any explicit query rewriting. Optionally, $q_{c,t}$ can be replaced by a rewritten standalone question produced by an instruction-tuned language model conditioned on $(H_{c,t}, u_t)$, which resolves coreferences and fills omitted context. We treat this query rewriting as a plug-in to the first-stage retrieval and evaluate it separately in experiments, while the core SaLSA-RAG components (state induction, statute summarization, and SaLSA-Reranker) remain unchanged.

3.2 Framework Overview

Given a conversation c and the t-th user turn, our goal is to retrieve a small set of statutory articles that are applicable to the user-described situation under the evolving dialogue context, and then generate a grounded answer. Figure 2 summarizes the pipeline.

(1) Turn query view. We first construct a turn-level query view $q_{c,t}$ from the current utterance and dialogue history. In the main setting, $q_{c,t}$ is formed without explicit rewriting, using the available conversational context to resolve underspecification. Optionally, $q_{c,t}$ can be replaced by a rewritten standalone question produced by an instruction-tuned language model conditioned on $(H_{c,t}, u_t)$; we treat this rewriting as a plug-in and analyze it separately in experiments.

(2) Legal state induction. In parallel, we induce a concise legal analysis state $s_{c,t}$ that abstracts the legally salient situation at this turn, such as parties and relationships, disputed issues, key facts, procedural posture, and the sub-question to resolve. This state is not a paraphrase of the utterance. Instead, it organizes the case into structured legal facets that are useful for assessing statutory applicability, complementing the surface-form query view $q_{c,t}$.

(3) Candidate retrieval. A first-stage dense retriever takes $q_{c,t}$ as input and retrieves a candidate set $\mathcal{A}_{c,t}$ of top-K statutory articles from the law library.

(4) SaLSA-Reranker. We then apply SaLSA-Reranker (Section 3.5) to rerank $\mathcal{A}_{c,t}$ by aligning the turn query view $q_{c,t}$ and the induced state $s_{c,t}$ with an applicability-oriented summary of each candidate statute. This alignment provides an explicit signal for filtering candidates that are topically similar but legally inapplicable.

(5) Answer generation. Finally, we pass the top-ranked articles (together with their summaries) to a generator to produce the response grounded in statutory evidence.

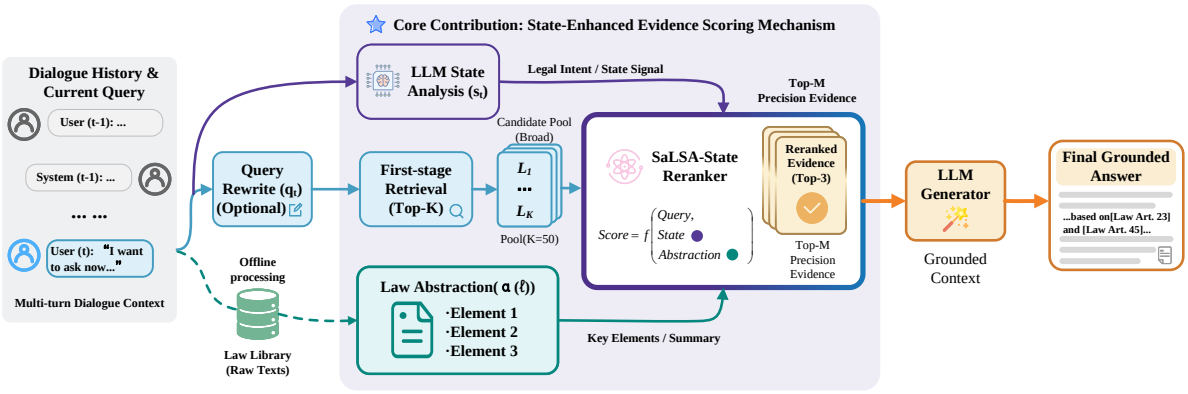

Figure 2: SaLSA-RAG pipeline. For each turn, we induce a legal state and retrieve top-K candidates, then SaLSA-Reranker aligns the query/state with statute summaries for reranking before generation.

3.3 Conversation State Representation

A turn utterance u_t (and its turn query view $q_{c,t}$) often provides only a partial description of the underlying case. In practice, legal consultation depends on a compact internal case model: who the parties are and how they relate, what the disputed issue is, which key facts matter, what the procedural posture is, and what sub-question this turn is trying to resolve. Much of this information is implicit or scattered across the dialogue history.

SaLSA-RAG therefore associates each turn (c, t) with a concise legal analysis state $s_{c,t}$ in natural language. The state is produced by applying a fixed prompting template to the dialogue context $(H_{c,t}, u_t)$. The template instructs a language model to distill the situation into a small set of legal facets (e.g., parties and relationships, dispute focus, key facts, procedural stage, plausible legal domains, and the current sub-question) and to output a short, structured analysis. We write this as

$$s_{c,t} = \Phi_{\text{state}}(H_{c,t}, u_t),$$

where Φ_{state} denotes the prompting operator and the underlying language model.

The resulting $s_{c,t}$ is typically a few sentences long: it is more explicit than the raw utterance about roles, disputes, and legally salient constraints, while remaining much shorter than the full history $H_{c,t}$. In our pipeline, $s_{c,t}$ serves two purposes. First, it provides an additional textual view for evidence selection by SaLSA-Reranker, complementing the query view $q_{c,t}$. Second, it supplies a compact, case-oriented description that can be reused during answer generation.

To enable scoring, we embed the state text using the same sentence encoder as the dense retriever and treat it as an additional view in SaLSA-Reranker (Section 3.5).

3.4 Law Article Abstraction

Statutory articles are written in formal and compact language, often containing nested conditions, exceptions, and cross-references. Directly matching user-side text against raw provisions can be noisy, because surface overlap does not reliably reflect legal applicability. SaLSA-RAG addresses this mismatch by constructing an applicability-oriented abstraction for each statute.

Given an article $\ell \in \mathcal{L}$ with its official name and full text, we apply a prompting template that asks a language model to summarize the provision into a compact representation. The template focuses on information that supports applicability assessment, such as operative conditions, core legal effects, and typical scenarios of invocation. The resulting fields are concatenated into a short paragraph g_ℓ:

$$g_\ell = \Phi_{\text{law}}(\ell),$$

where Φ_{law} denotes the law-summary prompting operator.

Summaries are generated offline for all statutes that appear in any candidate pool and cached for reuse. Compared to raw statutes, g_ℓ removes syntactic detail and foregrounds applicability cues, which makes it a better target for alignment with both the turn query view $q_{c,t}$ and the induced legal analysis state $s_{c,t}$.

3.5 SaLSA-Reranker: State-Aware Evidence Retrieval

SaLSA-RAG adopts a two-stage evidence retrieval design. The first stage builds a small candidate pool

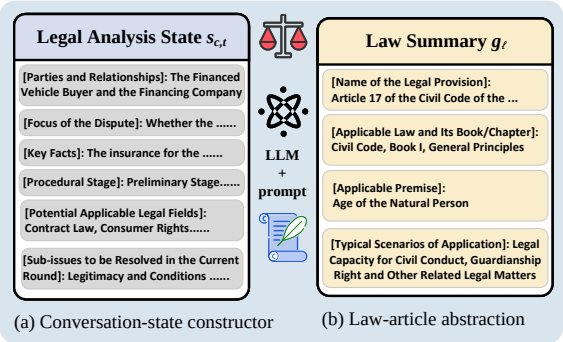

Figure 3: Component illustrations. (a) Conversation-state constructor: a language model distills multi-turn context into a structured legal analysis state. (b) Law-article abstraction: a language model summarizes raw statutes into a compact, applicability-oriented representation.

using dense retrieval; the second stage, SaLSA-Reranker, reranks these candidates by aligning user-side semantics with applicability-oriented statute summaries.

Stage-1 candidate retrieval. Given a turn (c, t), we obtain a retrieval query view $q_{c,t}$ (derived from the current utterance and its dialogue history; Section 3.2). A dense retriever uses $q_{c,t}$ and the raw statute texts to retrieve a fixed-size candidate set $\mathcal{C}_{c,t} \subset \mathcal{L}$ along with dense similarity scores. SaLSA-RAG keeps the first-stage encoder and index fixed, and focuses on refining $\mathcal{C}_{c,t}$ using the additional semantic views introduced in Sections 3.3 and 3.4.

Multi-view embeddings. We embed the turn query view, the induced conversation state, and the law summary into a shared vector space with a single sentence encoder $E(\cdot)$. For each (c, t) and each candidate article $\ell \in \mathcal{C}_{c,t}$, we compute ℓ_2-normalized embeddings:

$$\mathbf{q}_{c,t} = E(q_{c,t}), \qquad \mathbf{s}_{c,t} = E(s_{c,t}), \qquad \mathbf{z}_\ell = E(g_\ell),$$

where g_ℓ is the applicability-oriented summary of ℓ (Section 3.4). In addition, the first stage provides a dense similarity score $s_{\text{dense}}(q_{c,t}, \ell)$ computed from $E(q_{c,t})$ and the raw statute text.

Feature construction. Within each candidate set $\mathcal{C}_{c,t}$, we normalize dense scores by min–max scaling so that the lowest-scoring candidate maps to 0 and the highest-scoring candidate maps to 1. We denote the resulting score by $\widehat{s}_{c,t,\ell}$ and use it as the first reranking feature.

We then compute three cosine similarities from the multi-view embeddings:

$$\text{sim}_{q,\ell} = \cos(\mathbf{q}_{c,t}, \mathbf{z}_\ell),$$

$$\text{sim}_{s,\ell} = \cos(\mathbf{s}_{c,t}, \mathbf{z}_\ell),$$

$$\text{sim}_{q,s} = \cos(\mathbf{q}_{c,t}, \mathbf{s}_{c,t}).$$

Here, $\text{sim}_{q,\ell}$ measures alignment between the user-facing query view and the statute summary, while $\text{sim}_{s,\ell}$ measures alignment between the induced legal state and the statute summary. Finally, $\text{sim}_{q,s}$ is article-independent and captures the consistency between the turn query view and the induced state. A low $\text{sim}_{q,s}$ indicates that the induced state emphasizes facets that are weakly supported by the user-side description at this turn, which can make state-based matching less reliable.

Each candidate $\ell \in \mathcal{C}_{c,t}$ is represented by a compact four-dimensional feature vector:

$$\mathbf{x}_{c,t,\ell} = \left[\widehat{s}_{c,t,\ell},\ \text{sim}_{q,\ell},\ \text{sim}_{s,\ell},\ \text{sim}_{q,s}\right]^\top.$$

This design keeps reranking lightweight and interpretable: each feature corresponds to either dense retrieval evidence or an explicit alignment between user-side and statute-side representations.

Supervised evidence scorer. We train a supervised scorer over the fixed candidate sets $\mathcal{C}_{c,t}$. For each training turn (c, t) and candidate $\ell \in \mathcal{C}_{c,t}$, we attach a binary label $y_{c,t,\ell} \in \{0, 1\}$:

$$y_{c,t,\ell} = \begin{cases} 1, & \text{if } \ell \in \mathcal{L}^*_{c,t}, \\ 0, & \text{otherwise.} \end{cases}$$

Following standard practice for candidate-set reranking, we drop training turns whose candidate pools contain no relevant statute.

SaLSA-Reranker is parameterized as a logistic regression model:

$$f_\theta(\mathbf{x}_{c,t,\ell}) = \sigma(\mathbf{w}^\top \mathbf{x}_{c,t,\ell} + b),$$

where $\sigma(\cdot)$ is the sigmoid function and $\theta = (\mathbf{w}, b)$ are learned parameters. We minimize a regularized logistic loss with class weighting to address the imbalance between relevant and non-relevant candidates. The resulting linear model is efficient to train, robust on small feature sets, and directly interpretable via feature weights.

At test time, the dense retriever produces $\mathcal{C}_{c,t}$ and dense scores. For each candidate $\ell \in \mathcal{C}_{c,t}$, we compute the feature vector $\mathbf{x}_{c,t,\ell}$ using the cached

state $s_{c,t}$ and law summary g_ℓ, apply f_θ, and sort candidates by the predicted score. The top-ranked statutes are then passed to the generator as evidence.

3.6 State-Guided Answer Generation

The final component of SaLSA-RAG is a generator that produces natural-language legal advice at each turn conditioned on the dialogue context and retrieved statutory evidence. Given a turn (c, t), we take the top-ranked articles under the reranked list and build an evidence context by concatenating each article's name and its applicability-oriented summary g_ℓ. When needed, we optionally append short excerpts from the raw statute text to preserve exact legal phrasing.

The generator is prompted with four inputs: (i) the dialogue history $H_{c,t}$ and current user utterance u_t, (ii) the query view $q_{c,t}$, (iii) the induced legal analysis state $s_{c,t}$, and (iv) the evidence context constructed from the retrieved articles. The prompt instructs the model to (a) ground its advice in the provided statutes, (b) explain the reasoning in plain language, and (c) cite the relevant provisions when stating legal conclusions.

The induced state $s_{c,t}$ plays a complementary role across retrieval and generation. In reranking, it provides a structured characterization of the turn that is aligned against statute summaries to better select applicable evidence. In generation, it serves as a compact case description that organizes the answer around the key parties, facts, procedural posture, and the specific sub-question at the current turn. By reusing the same state representation in both stages, SaLSA-RAG couples state-aware evidence selection with state-guided response construction for multi-turn legal RAG.

4 Experiments

4.1 Experimental Setup

Dataset and evaluation. We evaluate on LEXRAG (Li et al., 2025), a Chinese multi-turn legal consultation benchmark with turn-level relevance judgments over supporting statutory articles. We follow the official train/dev/test split and the evaluation protocol in LexRAG. Retrieval is evaluated at the turn level using Recall@k (R@k), nDCG@k (N@k), and MRR. For end-to-end RAG, we report (i) keyword-based grounding metrics (keyword accuracy/recall and hit rate, following LexRAG), and (ii) LLM-as-a-judge scores with dimensions for overall quality, factual/legal accuracy, and user-need satisfaction. Unless stated otherwise, we aggregate metrics over all turns in the split.

History-aware inputs and query variants. Unless stated otherwise, the retrieval input for turn (c, t) concatenates the current user utterance with user-only history, i.e., $\langle u_t;\, u_1, \ldots, u_{t-1} \rangle$ (assistant responses are excluded). Our main setting uses this history-aware input directly (NoRewrite). We additionally report a QueryRewrite variant that rewrites $\langle u_t;\, u_1, \ldots, u_{t-1} \rangle$ into a standalone question; this is included to test complementarity rather than being required by the framework.

Conversation states and law summaries. For each turn, we induce a short legal analysis state capturing legally salient facets (e.g., parties/relations, key facts, dispute focus, procedural status, and the sub-question). For each statute, we construct an applicability-oriented summary distilling key preconditions and legal effects. States and summaries are generated offline using a fixed prompting template with gpt-4o-mini and cached for reuse; the state induction uses only user-side conversation text and does not access gold statutes.

Candidate pools and rerankers. We consider BM25 and dense retrieval as first-stage candidate pools and rerank the top-K=50 candidates for controlled comparison. BM25 uses BM25Okapi with jieba[1] tokenization over statute texts. Dense retrieval encodes queries and statutes with bge-base-zh by default (and Qwen embedding variants when specified) and retrieves by cosine similarity over ℓ_2-normalized embeddings. We compare an off-the-shelf neural reranker (Qwen3-Reranker-4B) with our SaLSA-Reranker, which is trained on the train split and reranks the same fixed candidate set using a lightweight feature-based scorer over multi-view alignments.

Answer generation and metrics. For end-to-end RAG, we provide the generator with the conversation context and the top retrieved statutes as evidence; SaLSA-RAG additionally supplies the induced state as a compact planning scaffold. We use gpt-4o-mini as the generator (temperature 0) and follow the LexRAG protocol for keyword-based grounding metrics and LLM-as-a-judge evaluation.

[1] https://github.com/fxsjy/jieba

Model	Dev						Test					
	R@1	R@3	R@5	R@10	N@10	MRR	R@1	R@3	R@5	R@10	N@10	MRR
BM25 candidate pool												
BM25	0.0606	0.1088	0.1287	0.1776	0.1147	0.1056	0.0615	0.0968	0.1373	0.1918	0.1179	0.1025
+ Qwen3-Reranker-4B	0.0616	0.1400	0.1653	**0.2290**	0.1402	0.1229	0.0631	0.1329	0.1623	0.2154	0.1350	0.1233
+ SaLSA-Reranker (ours)	0.0859	0.1478	0.1786	0.2037	0.1465	0.1375	0.0898	0.1565	0.1765	0.2173	0.1520†	0.1421†
+ SaLSA-Reranker (ours, Qwen-4B)	**0.0992**	0.1483	0.1760	0.2095	0.1544†	0.1466†	0.0961	0.1597	0.1995	**0.2483**	0.1693†‡	0.1536†‡
+ SaLSA-Reranker (ours, Qwen-8B)	**0.0992**	**0.1626**	**0.1838**	0.2122	**0.1563**†	**0.1473**†	**0.1087**	**0.1723**	**0.2116**	0.2325	**0.1732**†‡	**0.1660**†‡
Dense candidate pool												
Dense	0.0441	0.0862	0.1201	0.1726	0.1007	0.0914	0.0461	0.1017	0.1495	0.1978	0.1143	0.1012
+ Qwen3-Reranker-4B	0.0760	0.1526	0.2095	0.2639	0.1617	0.1418	0.0657	0.1518	0.1990	0.2741	0.1624	0.1412
+ SaLSA-Reranker (ours)	0.0791	0.1581	0.2009	0.2588	0.1637	0.1420	0.0929	0.1723	0.2152	0.2771	0.1798	0.1588
+ SaLSA-Reranker (ours, Qwen-4B)	0.1109	0.1797	0.2239	0.2677	0.1873†	0.1713†‡	**0.1118**	0.1925	0.2460	0.2939	0.1995†‡	0.1784†‡
+ SaLSA-Reranker (ours, Qwen-8B)	**0.1304**	**0.2105**	**0.2358**	**0.2728**	**0.2018**†‡	**0.1889**†‡	0.1108	**0.2257**	**0.2624**	**0.3064**	**0.2109**†‡	**0.1890**†‡
Dense + QueryRewrite	0.0760	0.1366	0.1633	0.2112	0.1415	0.1330	0.0493	0.1101	0.1384	0.2013	0.1174	0.1017
+ Qwen3-Reranker-4B	0.0698	0.1386	0.1674	0.2418	0.1487	0.1292	0.0430	0.1101	0.1331	0.1929	0.1118	0.0969
+ SaLSA-Reranker (ours)	0.1016	0.1947	0.2352	0.2882	0.1938	0.1742	0.0964	0.1761	0.2075	0.2762	0.1793	0.1579
+ SaLSA-Reranker (ours, Qwen-8B)	**0.1311**	**0.2296**	**0.2584**	**0.3005**	**0.2169**	**0.2017**	**0.1122**	**0.2075**	**0.2516**	**0.2872**	**0.1997**	**0.1784**

Table 1: Retrieval performance on LexRAG (Li et al., 2025) with user-only conversational history. Unless specified, all rerankers operate on the top-K=50 candidates from the corresponding first-stage retriever. The default dense encoder is bge-base-zh; encoder variants are indicated in the model name (Qwen-4B/8B). Qwen3-Reranker-4B is an off-the-shelf neural reranker baseline. N@10 denotes nDCG@10. Superscripts on Dev and Test N@10 and MRR indicate statistical significance using paired permutation tests with bootstrap (20,000 permutations) on the common query set: † denotes improvement over the corresponding first-stage retriever (BM25 or Dense), and ‡ denotes improvement over Qwen3-Reranker-4B (p<0.05). Best result within each candidate-pool group is highlighted in bold.

4.2 Main Results

4.2.1 Retrieval Results

Table 1 reports retrieval quality on LexRAG with user-only conversational history under three candidate-pool settings.

SaLSA-Reranker improves evidence selection across pools. Across both BM25 and dense candidate pools, SaLSA-Reranker consistently improves N@10 and MRR over the corresponding first-stage retrievers. The gains are most pronounced in ranking quality at the top of the list, which is critical for downstream legal RAG. Qualitatively, this matches our motivation: conversational legal relevance depends on applicability conditions, and the induced legal state together with statute summaries provides complementary signals beyond surface lexical or query-only semantic matching. As a result, SaLSA-Reranker better suppresses statutes that look plausible by topic but fail to satisfy key preconditions.

Stronger encoders help, but do not replace state-aware alignment; rewriting is optional. Scaling the embedding encoder generally improves results, yet SaLSA-Reranker continues to provide additional gains on top of stronger representations. This suggests that the benefit mainly comes from task-aligned, applicability-oriented features rather than merely increasing fusion capacity. Moreover, our main setting does not rely on query rewriting: SaLSA-Reranker remains effective in the NoRewrite setting, while QueryRewrite can be added as a complementary option when available to further strengthen candidate quality.

Statistical significance. We verify that the main improvements in N@10 and MRR are statistically significant using paired permutation tests with bootstrap confidence intervals; full results and test details are provided in Appendix D.

4.2.2 Generation Results

Table 2 reports end-to-end answer quality on the Test split. Across both BM25 and dense candidate pools, SaLSA-RAG improves keyword-based evidence coverage and achieves higher GPT-5-mini judge scores than the corresponding baselines. This is consistent with the retrieval gains in Table 1, suggesting that evidence selection is a key driver of generation quality in multi-turn legal RAG. We also include results with an alternative judge (GPT-4o-mini) in Appendix F.

4.3 Ablation Studies

Encoder scaling. We further vary the embedding backbone used for SaLSA features under a fixed setting; larger encoders consistently improve retrieval, and Qwen3-Embedding-8B performs best (Appendix Table 9).

Model	Keyword metrics			LLM-judge		
	Acc	Rec	Hit	Score	Acc	Need
BM25 candidate pool						
BM25 baseline	0.306	0.306	0.680	5.25	4.12	4.98
+ Qwen3-Reranker-4B	0.290	0.290	0.657	5.20	4.06	4.93
+ SaLSA(ours)	0.365	0.359	0.729	5.69	4.52	5.50
+ SaLSA(ours,Q-4B)	**0.380**	**0.372**	**0.737**	**5.74**	**4.55**	**5.54**
+ SaLSA(ours,Q-8B)	0.372	0.366	0.729	5.69	4.54	5.50
Dense candidate pool						
Dense baseline	0.289	0.286	0.647	5.13	3.99	4.86
+ Qwen3-Reranker-4B	0.317	0.322	0.694	5.57	4.32	5.34
+ SaLSA(ours)	0.381	0.370	0.741	5.63	4.45	5.46
+ SaLSA(ours,Q-4B)	0.389	0.373	0.743	5.73	4.54	5.49
+ SaLSA(ours,Q-8B)	**0.396**	**0.380**	**0.747**	**5.83**	**4.69**	**5.62**

Table 2: End-to-end generation quality on Test. Keyword metrics measure evidence coverage, and LLM-judge scores are computed by GPT-5-mini under the same rubric.

Ablation on feature sources. We study which semantic views contribute to SaLSA-Reranker. In the no-rewrite setting, let q be the history-aware turn query, s the induced legal state, and $sum(\ell)$ the applicability-oriented summary of a candidate statute ℓ. All variants include the normalized first-stage dense score and optionally add cosine-similarity features between $\mathrm{Emb}(q)$, $\mathrm{Emb}(s)$, and $\mathrm{Emb}(sum(\ell))$: query-summary ($qsum$), state-summary ($ssum$), and query-state (qs). The full model uses $\{dense, qsum, ssum, qs\}$. We ablate (i) without conversational state (removing state-related signals, leaving $\{dense, qsum\}$), and (ii) without query-summary alignment (removing direct query evidence, leaving $\{dense, ssum\}$).

Table 3 shows that both views are beneficial and complementary. Removing the conversational state degrades nDCG@10 from 0.1800 to 0.1608 and MRR from 0.1612 to 0.1411, indicating that state cues help promote legally applicable statutes to the top ranks. Removing query-summary alignment also hurts performance (nDCG@10 0.1715; MRR 0.1536), but remains stronger than removing the state, suggesting that state-summary matching provides a robust applicability signal even when the surface query is underspecified. Overall, combining query evidence with state-derived applicability signals yields the best head ranking and the highest Recall@10 (0.2650).

Fusion and learning-to-rank alternatives. Our reranker uses a linear scorer over a small set of aligned features to obtain stable and interpretable calibration under limited supervision. To test whether the gains mainly come from using a stronger generic fusion operator, we replace our linear scorer with common alternatives under the same setting (Dense+QueryRewrite; Qwen3-Embedding-8B; top-K=50): reciprocal rank fusion (RRF), a multi-layer perceptron (MLP) fuser, and a LightGBM ranker. Table 4 shows that while more complex models can be competitive, none surpass SaLSA-RAG. This suggests that the key benefit comes from task-aligned state and law-summary signals coupled with a well-calibrated, low-variance scorer, rather than from increasing fusion model complexity.

Method	R@10	N@10	MRR
Dense (baseline)	0.1726	0.1007	0.0914
SaLSA-Reranker (full)	**0.2650**	**0.1800**	**0.1612**
without conversational state	0.2461	0.1608	0.1411
without query-summary alignment	0.2543	0.1715	0.1536

Table 3: Feature-source ablation on Dev (Dense retrieval; no-rewrite with history; top-K=50). N@10 denotes nDCG@10.

Method	R@3	R@5	N@10	MRR
RRF fusion	0.1027	0.1593	0.1131	0.0908
MLP fusion	0.1887	0.2264	0.1782	0.1565
LightGBM ranker	0.1866	0.2212	0.1889	0.1690
SaLSA-RAG (ours)	**0.2075**	**0.2516**	**0.1997**	**0.1784**

Table 4: Fusion and LTR alternatives on Test (Dense+QueryRewrite; Qwen3-Embedding-8B; top-K=50). N@10 denotes nDCG@10.

5 Conclusion

We proposed SaLSA-RAG, a state- and law-summary aligned framework for conversational legal advice. Its core component, SaLSA-Reranker, improves evidence selection by aligning the turn-level legal state and user query with applicability-oriented statute summaries for lightweight reranking over a fixed candidate pool. Experiments on LexRAG show consistent retrieval gains under both BM25 and dense pools, which translate into improved end-to-end answer quality under keyword-based grounding and LLM-based judging. Future work will explore richer state representations and more efficient reranking/summarization for robust deployment in longer and harder legal dialogues.

Limitations

Our framework relies on instruction-tuned LLMs to induce conversation states and to produce statute summaries. While this design is modular, the quality of these intermediate representations may vary with the underlying model and prompting choices, and stronger models can further improve robustness. In addition, generating states and summaries introduces extra inference cost compared to pure retrieval pipelines, although summaries can be precomputed and reused. Finally, our experiments focus on a Chinese statutory corpus and one benchmark; broader validation across jurisdictions, statute formats, and longer real-world dialogues is a natural next step.

References

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, and 1 others. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of ICML*.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of ICML*.

Christopher J. C. Burges, Robert Ragno, and Quoc V. Le. 2006. Learning to rank with nonsmooth cost functions. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of EMNLP*.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2022. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of ACL*.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of SIGIR*.

Jeff Dalton, Chenyan Xiong, Jamie Callan, and 1 others. 2020. Overview of the TREC 2020 conversational assistance track (CAsT). In *Proceedings of TREC*.

Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can you unpack that? learning to rewrite questions-in-context. In *Proceedings of EMNLP-IJCNLP*.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse lexical and expansion model for first stage ranking. *arXiv preprint arXiv:2107.05720*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. In *Proceedings of ICML*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of ICLR*.

Gautier Izacard and 1 others. 2022. ATLAS: Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of SIGIR*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and Sebastian Riedel. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Haitao Li, Yifan Chen, Yiran Hu, Qingyao Ai, Junjie Chen, Xiaoyu Yang, Jianhui Yang, Yueyue Wu, Zeyang Liu, and Yiqun Liu. 2025. Lexrag: Benchmarking retrieval-augmented generation in multi-turn legal consultation conversation. *arXiv preprint arXiv:2502.20640*.

Yang Liu and 1 others. 2023. G-eval: Nlg evaluation using GPT-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. In *arXiv preprint arXiv:1901.04085*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Qiang Wu, Christopher J. C. Burges, Krysta M. Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270.

Lee Xiong, Chenyan Xiong, Ye Li, Karthik Tang, Jialin Liu, Paul Bennett, Javed Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proceedings of ICLR*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Suyu Zhuang, Zihao Wu, Yong Zhuang, Zhiqing Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging LLMs by LLMs: Benchmarking LLM-as-a-judge with MT-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

A Implementation details (supplementary)

Retrieval. BM25 uses jieba tokenization with BM25Okapi. Dense retrieval uses bge-base-zh and ranks by cosine similarity (inner product over ℓ_2-normalized embeddings); statute embeddings are precomputed and query embeddings are computed per turn.

Reranking. Qwen3-Reranker-4B reranks query–statute pairs with max length 512 and batch size 8. SaLSA-Reranker is trained on the train split with top-K=50 candidates from the corresponding first-stage retriever. Features include the normalized first-stage score and cosine similarities between embeddings of the query, induced state, and statute summary. We use logistic regression (C=1.0, `max_iter`=1000, `class_weight=balanced`).

Generation and judging. For generation, the evidence context consists of the top-N statutes (N=5), where each statute is represented by its title and an applicability-oriented summary; when budget allows, we additionally append a truncated excerpt of the original statute text to preserve precise legal wording. We use gpt-4o-mini as the generator with temperature 0. Unless noted, gpt-4o-mini is also used for state induction and statute summarization; artifacts are cached. LLM-judge evaluation uses GPT-5-mini; we report averages over valid (parsable) judge records.

Table 5 summarizes omitted hyperparameters.

Component	Setting / Hyperparameter
BM25 (pool)	Tokenizer: jieba; implementation: BM25Okapi (`rank_bm25`).
Dense retrieval (pool)	Encoder: bge-base-zh; similarity: cosine over ℓ_2-normalized embeddings (equiv. to inner product).
Neural baseline	Model: Qwen3-Reranker-4B; max length: 512; batch size: 8.
SaLSA-Reranker	Candidates: top-K from first-stage retrieval (K=50 unless specified). Features: normalized first-stage score + cosine similarities between {query/state/statute summary} embeddings. Learner: Logistic Regression (C=1.0, `class_weight=balanced`).
Evidence context	Top-N statutes (N=5): statute title + applicability summary; optionally a truncated excerpt of the original text if budget allows.
Intermediate artifacts	State induction and statute summarization: gpt-4o-mini; generated offline and cached.
Generator	Model: gpt-4o-mini; temperature: 0.
LLM judge	Model: GPT-5-mini; aggregation: turn-level scores averaged over the Test set (invalid parse entries are skipped).

Table 5: Implementation details and hyperparameters used in our experiments (supplementary).

Split	#Q (before)	#Q (after)	#Pos (before)	#Pos (mapped)	#Pos (unmapped)
Train	3817	3780	4394	4332	62
Dev	487	485	565	563	2
Test	477	469	542	531	11

Table 6: Coverage of qrels normalization against the indexed statute library. Unmappable positive judgments are skipped.

B Further Analysis

Turn-wise analysis. Figure 4 shows nDCG@10 as a function of the turn index. Dense retrieval performs poorly on early turns where the user request is often underspecified, and its quality varies substantially across turns. SaLSA-Reranker improves retrieval quality consistently across all turns, and the margin becomes larger in later turns. This trend aligns with our motivation: as the dialogue progresses, the information need increasingly depends on prior context and on applicability conditions, which are captured more explicitly by the induced legal state and leveraged by the reranker.

Candidate size sensitivity. Figure 5 evaluates retrieval quality as we vary the candidate size K. SaLSA-Reranker improves steadily as K increases, suggesting that state and summary alignment can promote relevant statutes that are retrieved but not highly ranked by the first-stage dense retriever. The gains taper off beyond K=100, indicating that moderate candidate sizes can provide a good balance between effectiveness and efficiency in practice.

Pool	Split	Comparison (A → B)	ΔN@10	ΔMRR	ΔR@10	Sig.
Dense	Test	Dense → SaLSA (Qwen-4B)	+0.085	+0.077	+0.096	‡
Dense	Test	Qwen3-Reranker-4B → SaLSA (Qwen-4B)	+0.037	+0.037	+0.020	∗
Dense	Dev	Dense → SaLSA (Qwen-4B)	+0.087	+0.080	–	‡
BM25	Test	BM25 → SaLSA (default)	+0.034	+0.040	+0.026	†
BM25	Test	BM25 → SaLSA (Qwen-4B)	+0.051	+0.051	+0.056	‡
BM25	Test	BM25+Qwen3-Reranker-4B → SaLSA (Qwen-4B)	+0.034	+0.030	+0.033	∗

Table 7: Paired significance tests for retrieval metrics. Δ denotes the absolute gain (B−A) on nDCG@10 (N@10), MRR, and Recall@10. All tests use two-sided paired permutation tests (20,000 permutations) on common query turns, with bootstrap 95% confidence intervals computed for the mean difference. Sig.: ‡ $p < 10^{-4}$, † $p < 10^{-3}$, ∗ $p < 0.05$. "–" indicates the metric was not computed for that comparison.

C Qrels–Corpus normalization and coverage

Some relevance judgments in LexRAG refer to statute identifiers that cannot be mapped to our indexed statute library (e.g., unmatched titles or non-indexed items). During preprocessing, we normalize each qrels document id to a unique statute-library id; any positive judgment that fails normalization is skipped. This filtering is deterministic and affects only a small fraction of the training labels.

We report #Q as the number of turns that have at least one positive label; "after" counts turns with at least one mappable positive statute id. All retrieval metrics and significance tests are computed on the normalized subset ("after") to avoid undefined labels.

In our experiments, retrieval evaluation and significance testing are conducted on the set of queries shared by the compared runs (common qids), ensuring a fair and consistent comparison even when some qrels entries are unmappable.

D Significance Testing for Retrieval Metrics

We conduct paired significance tests on per-turn retrieval metrics to assess whether SaLSA-Reranker yields reliable improvements over strong baselines. For each comparison, we run a two-sided paired permutation test with 20,000 permutations, and report bootstrap 95% confidence intervals for the mean difference. All tests are performed on the set of common query turns shared by the two runs (common qids).

Model	Keyword metrics			LLM-judge		
	Acc	Rec	Hit	Score	Acc	Need
BM25 candidate pool						
BM25 baseline	0.306	0.306	0.680	4.10	**5.10**	5.11
+ Qwen3-Reranker-4B	0.290	0.290	0.657	5.03	4.02	5.08
+ SaLSA(ours)	0.365	0.359	0.729	**5.31**	4.37	**5.28**
+ SaLSA(ours,Q-4B)	**0.380**	**0.372**	**0.737**	5.28	4.32	5.24
+ SaLSA(ours,Q-8B)	0.372	0.366	0.729	5.28	4.38	5.24
Dense candidate pool						
Dense baseline	0.289	0.286	0.647	5.04	4.25	5.14
+ SaLSA(ours)	0.381	0.370	0.741	**5.41**	**4.78**	**5.46**
+ SaLSA(ours,Q-4B)	0.389	0.373	0.743	5.26	4.39	5.26
+ SaLSA(ours,Q-8B)	**0.396**	**0.380**	**0.747**	5.32	4.45	5.30

Table 8: Generation quality on Test with GPT-4o-mini as the judge (same rubric).

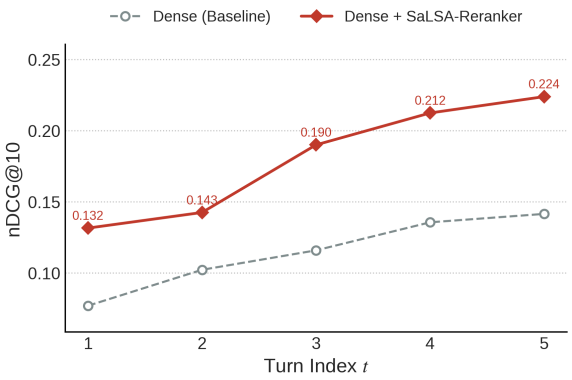

Figure 4: Turn-wise nDCG@10 on the Test split (dense candidate pool). SaLSA-Reranker consistently outperforms the dense baseline, with larger gains in later turns.

Encoder	R@10	N@10	MRR
bge-base-zh	0.2762	0.1793	0.1579
Qwen3-Embedding-0.6B	0.2600	0.1729	0.1547
Qwen3-Embedding-4B	0.2830	0.1904	0.1686
Qwen3-Embedding-8B	**0.2872**	**0.1997**	**0.1784**

Table 9: Embedding backbone ablation on Test under the Dense+QueryRewrite setting with top-K=50 candidates. N@10 denotes nDCG@10.

E Additional Encoder Ablations

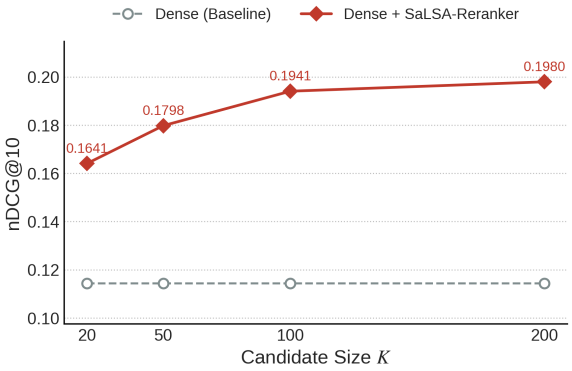

Figure 5: Candidate-size sensitivity on the Test split (dense candidate pool). nDCG@10 improves as K increases and saturates beyond K=100 for SaLSA-Reranker.

F Additional judge results (GPT-4o-mini)

We additionally report judge scores using GPT-4o-mini to show that the overall trends do not hinge on a single judge model.

G Case Study 7: Resettlement Housing Gifting (Chinese Excerpts + English Analysis)

Translation note. This case study is **originally in Chinese** and is reported in Chinese for fidelity. To facilitate presentation in an English paper, **we provide an English translation table** (Table 10) for key terms and queries. **The Chinese text is the primary source of truth**; the translations are for readability only.

T1 (S1): Can the house be gifted before registration?

User query (CN): 您好:我的父母想把他们的安置房送给他们的孙子，就是现在没有房产证。
Gold: 《民法典》209条 (registration as a key requirement).

Baseline

- **Retrieval:** *[Miss]* Miss 209.
- **Citation/Reasoning:** Relies on **[Alt-Support]** 659-like "registration needed" language but does not align with the core "no-certificate/no-registration" legal boundary.

Ours

- **Retrieval:** *[Miss]* Miss 209, but hits **[Alt-Support]** 659-style procedural support.
- **Citation/Reasoning:** More actionable by explicitly stating "title transfer requires registration," but still misses the Gold anchor (209).

Chinese term (verbatim)	English translation (used in this appendix)
安置房/ 拆迁安置房	Resettlement housing (relocation unit)
房产证/ 不动产登记	Property certificate / real-estate registration
赠与（合同）	Gift (donation) contract
过户/ 变更登记	Title transfer / registration change
合同有效vs. 物权变动	Contract validity vs. property-right transfer
撤销（赠与撤销）	Revocation of a gift
税费（税务问题）	Taxes/fees (tax-law domain shift)

Table 10: Term glossary (Chinese → English) for Case Study 7.

State	Trigger (CN)	Sub-problem to resolve (EN)	Gold
S1	无证赠与意图	Whether gifting can produce *property-right transfer* before registration	209
S2	询问先签协议	Contract may be signed, but does not imply title transfer (contract vs. registration)	657
S3	办证后落地	Execution requirements: gift contract + registration change (procedural closure)	657/667
S4	税费域切换	Tax-law domain shift: emphasize locality and conservative guidance	Tax-law mainline
S5	撤销问题	Pre-transfer arbitrary revocation vs. post-transfer statutory revocation	666

Table 11: State decomposition for multi-turn legal RAG analysis in Case Study 7.

T2 (S2): Can we sign a gift agreement without a certificate?

User query (CN): 安置房没有房产证的情况下可以签订赠与协议吗
Gold: 《民法典》657条 (definition of a gift contract).

Baseline

- **Retrieval:** **[Match]** Hit 657.
- **Citation/Reasoning:** Over-weights **[Alt-Support]** 659 and risks conflating *contract validity* with *title transfer*.

Ours (Better grounding)

- **Retrieval:** **[Match]** Hit 657 (high-ranked), plus **[Alt-Support]** 659.
- **Citation/Reasoning:** Uses **[Match]** 657 to support "the agreement can be signed," and **[Alt-Support]** 659 to clarify "registration is needed for title transfer" (valid contract ≠ property-right transfer).

T3 (S3): How to complete gifting after the certificate is issued?

User query (CN): 房产证办理后如何进行赠与
Gold: 657/667.

Baseline

- **Retrieval:** **[Match]** Hit 657; *[Miss]* Miss 667.
- **Citation/Reasoning:** Mostly procedural advice; Gold alignment remains incomplete (*[Miss]* 667).

Ours

- **Retrieval:** **[Match]** Hit 657; *[Miss]* Miss 667.
- **Citation/Reasoning:** States a clear two-step requirement: *(i) gift contract + (ii) registration change*, supported by **[Match]** 657 and **[Alt-Support]** 659-style registration language.

T4 (S4): Taxes/fees for gifting (Tax-law domain shift)

User query (CN): 赠与房产是否需要缴纳税费
Gold: Tax-law mainline (and "local policy differs").

Baseline (failure mode)

- **Retrieval:** *[Miss]* Does not align with tax-law mainline.
- **Citation/Reasoning:** **[Hallucination/Irrelevant]** Mixes holding tax (property tax) with **[Hallucination/Irrelevant]** transfer-related taxes/fees, leading to misleading guidance.

Ours (closer, but imperfect)

- **Retrieval:** *[Miss]* Misses key tax-law anchors.
- **Citation/Reasoning:** Points to income-tax-like obligations but shows **[OOR]** out-of-retrieval tax citation; should emphasize "consult local tax authority" more conservatively.

T5 (S5): Can the gift be revoked?

User query (CN): 安置房赠与后是否可以撤销
Gold: 《民法典》666条.

Baseline

- **Retrieval:** *[Miss]* Miss 666.
- **Citation/Reasoning:** Vague revocation discussion; lacks a clear pre-/post-transfer distinction.

Ours

- **Retrieval:** **[Match]** Hit 658 (pre-transfer revocation) and **[Match]** 663 (statutory revocation); **[Alt-Support]** 666 retrieved but not used.
- **Citation/Reasoning:** Clearly separates *before title transfer* vs. *after title transfer* conditions, producing a more executable legal explanation.

Conclusion. Across T2–T3–T5, our system is more consistent in state-aware legal structuring: it repeatedly separates (i) contract formation from (ii) registration-based title transfer, and explicitly distinguishes revocation rules before vs. after title transfer. The key weakness is the tax-law domain shift (T4), where retrieval coverage is insufficient and the model risks out-of-retrieval citations; this motivates stronger tax-law indexing and/or a more conservative "local policy + consult tax authority" response policy.

H Artifacts to be released

To facilitate reproducibility, we will release the following artifacts upon publication (or earlier if permitted by the venue policy):

- **Code.** Training and inference code for SaLSA-Reranker and the end-to-end SaLSA-RAG pipeline, including preprocessing, feature construction, and evaluation scripts.
- **Prompts and templates.** The prompts used for legal state induction, statute summarization, and answer generation, as well as the LLM-judge instruction templates and parsing code.
- **Configurations.** YAML/JSON configuration files specifying model checkpoints, hyperparameters, decoding parameters, and retrieval/reranking settings.
- **Run scripts.** Shell scripts to reproduce all main tables, significance tests, and ablations, together with example commands and expected output formats.
- **Intermediate artifacts.** Cached conversational states and statute summaries (when redistribution is allowed), or alternatively, scripts to regenerate them from the released prompts.
- **Intermediate artifacts.** Scripts to regenerate conversational states and statute summaries from the released prompts (and optional cached files when redistribution is allowed).

We will also provide a short README describing the environment setup, data preparation steps, and an end-to-end reproduction checklist for Tables 1 and 2.