SAGA: Structured Applicability-Guided Alignment for Conversational Legal Retrieval

Anonymous ACL submission

Abstract

Conversational legal retrieval requires going beyond broad topical similarity toward statutes that are better matched to the user's case context. Standard sparse and dense retrievers often struggle in this setting, because they may surface statutes that are semantically related but not well aligned with the specific factual conditions described in the dialogue. To address this problem, we propose SAGA, a training-free legal retrieval framework that injects structured applicability-oriented cues into dense retrieval. SAGA constructs a CAF from user-side dialogue context to summarize dispute issues, core facts, and missing facts, and uses an offline AAF to represent statute-side issue tags, prerequisites, legal effects, and evidence cues. It then combines a raw semantic retrieval stream with a structural applicability stream through late fusion, allowing broad semantic relevance and structured legal cues to complement each other. Experiments on two Chinese statute retrieval benchmarks, LexRAG and STARD, show that SAGA improves over strong sparse, dense, and LLM-enhanced retrieval baselines under a unified full-corpus retrieval protocol. In particular, SAGA yields statistically significant gains on the conversational LexRAG benchmark and more modest gains on STARD over the strongest dense baseline, improving nDCG@10 from 0.3504 to 0.3696 on LexRAG and from 0.5862 to 0.5940 on STARD. These results suggest that structured applicability-oriented cues provide useful retrieval signals beyond raw semantic matching alone, especially in conversational legal retrieval.

1 Introduction

Legal assistants and legal question-answering systems depend critically on accurate statutory retrieval. In realistic consultations, however, users rarely describe their cases in precise legal language. Instead, they reveal facts incrementally, often in colloquial, incomplete, and context-dependent forms. As a result, legal retrieval is harder than standard topical search: the system must identify statutes that are not only semantically related to the query, but also better matched to the user's factual situation.

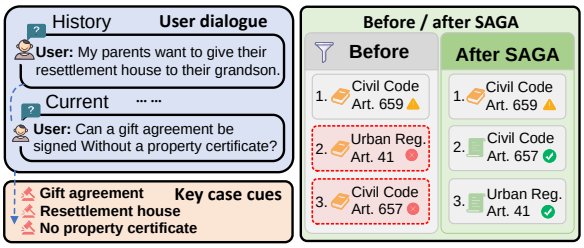

Figure 1: Illustrative example of conversational legal retrieval. Raw retrieval may return statutes that are topically related but not optimally ranked for the user's case context. By incorporating structured applicability-oriented cues, SAGA produces a more suitable ranking of statutory evidence. Figure text is translated from the original Chinese data; experiments use Chinese inputs.

This gap between semantic relevance and case compatibility is especially important in conversational legal retrieval. A statute may appear highly relevant because it shares surface-level vocabulary or broad topical similarity with the user's question, yet still be poorly matched to the actual dispute because its prerequisites are not satisfied, it governs a different legal subject, or crucial facts are still missing. The problem is further amplified by the mismatch between layperson expressions and formal statutory language. Consequently, strong sparse retrievers, dense encoders, and even LLM-enhanced retrieval methods can still return statutes that are topically plausible but not well aligned with the case context.

To address this problem, we propose **SAGA**, a training-free legal retrieval framework that injects structured applicability-oriented cues into dense retrieval. On the query side, SAGA constructs a

Case Applicability Frame (CAF) that organizes the user's legal situation into structured signals such as dispute issues, core facts, and missing facts. On the statute side, SAGA uses an offline Article Applicability Frame (AAF) that summarizes each statute in terms of issue tags, prerequisites, legal effects, and evidence cues. SAGA then combines a raw semantic retrieval stream with a structural applicability stream, allowing broad semantic relevance and structured legal cues to complement each other in a unified ranking framework.

SAGA is designed as a training-free method under a unified full-corpus retrieval setting. Rather than relying on raw semantic similarity alone, it directly ranks the statute library by integrating broad semantic relevance from raw text with structured applicability-oriented cues derived from CAF and AAF. This design is particularly suitable for legal retrieval: strong semantic encoders are useful but often insufficient, while purely structured representations can highlight case-side and statute-side conditions that are easy to overlook in raw text alone. By combining both signals, SAGA preserves the strengths of dense retrieval while reducing its tendency to over-rely on superficial semantic similarity.

We evaluate SAGA on two Chinese statute retrieval benchmarks with different characteristics: LexRAG, a multi-turn legal consultation benchmark, and STARD, a broader statute retrieval benchmark for layperson queries. Across both datasets, SAGA improves over strong sparse, dense, and LLM-enhanced retrieval baselines under the same full-corpus retrieval protocol, with clearer and statistically significant gains in the conversational LexRAG setting. This pattern suggests that structured applicability-oriented cues are especially helpful in conversational retrieval, where user intent and legally relevant case conditions are often only partially revealed in dialogue.

In summary, our main contributions are as follows:

- We propose **SAGA**, a training-free legal retrieval framework that bridges colloquial user descriptions and statute-side legal conditions through structured applicability-oriented representations.
- We introduce a dual-stream retrieval design that combines raw semantic retrieval with structural applicability cues within a unified full-corpus retrieval framework.
- We conduct experiments on two Chinese statute retrieval benchmarks, LexRAG and STARD, showing that SAGA improves over strong sparse, dense, and LLM-enhanced baselines, with clearer gains in the conversational LexRAG setting, while further analyses clarify which structured cues matter most and how the two retrieval streams should be balanced.

2 Related Work

Legal retrieval and general retrieval models. Accurate statute retrieval is a core component of legal assistants and legal reasoning systems. Existing retrieval pipelines commonly rely on sparse lexical matching such as BM25 (Robertson and Zaragoza, 2009), dense bi-encoder retrieval (Karpukhin et al., 2020; Xiong et al., 2020), late-interaction models (Khattab and Zaharia, 2020), or neural cross-encoder rerankers (Nogueira and Cho, 2019). These methods are effective at modeling topical or semantic relevance and therefore provide strong baselines for legal retrieval. However, legal retrieval differs from open-domain search in an important way: a statute may be topically related to a query while still being legally inapplicable. In practice, correct retrieval often depends on whether statutory prerequisites are satisfied, whether the statute governs the relevant legal subject, and whether crucial facts are present or still missing. This makes legal retrieval less a problem of broad semantic similarity alone, and more a problem of matching facts and legal conditions under domain-specific constraints.

Conversational retrieval, query reformulation, and fusion. Conversational retrieval must deal with omissions, references, and information revealed incrementally across turns (Dalton et al., 2021). A common strategy is to rewrite the current turn into a more self-contained query (Elgohary et al., 2019). While effective in many settings, rewriting typically flattens dialogue context into a single text sequence and may fail to preserve legally salient structure, such as dispute issues, procedural posture, or missing facts. More broadly, classical fusion and learning-to-rank methods (Cormack et al., 2009; Burges et al., 2005, 2006; Wu et al., 2010; Ke et al., 2017) provide general ways to combine retrieval signals, but they do not by themselves specify which legal signals should be made explicit. These limitations motivate retrieval designs that represent dialogue context in a more

structured, applicability-oriented form rather than relying on query rewriting or generic fusion alone. Recent LLM-based retrieval augmentation methods improve retrieval through query rewriting or hypothetical document generation, whereas SAGA instead introduces structured case-side and statute-side cues.

Legal NLP resources and evaluation settings. Legal-domain models and benchmarks have substantially expanded the scope of legal NLP research (Chalkidis et al., 2020, 2022). Recent resources increasingly highlight the gap between layperson descriptions and formal legal texts. For example, STARD (Su et al., 2024) studies real-life, non-professional legal queries for statute retrieval, while CitaLaw (Zhang et al., 2025a) emphasizes legal reasoning and citation-grounded support. These resources reveal an open challenge: retrieval must often bridge colloquial case descriptions and statute-side legal conditions, especially in multi-turn consultations where relevant facts emerge gradually. Our work addresses this challenge by introducing structured representations on both the case side and the statute side, enabling retrieval to incorporate applicability-oriented signals more explicitly.

3 Method

We propose **SAGA**, a training-free legal retrieval framework that incorporates structured applicability-oriented cues into dense retrieval. Its goal is to move beyond broad semantic matching by representing both the query side and the statute side in a more structured form. On the query side, SAGA constructs a CAF to summarize the user's case context. On the statute side, it uses an AAF to represent statute-side issue tags, prerequisites, legal effects, and evidence cues. By combining raw semantic retrieval with structured applicability-oriented retrieval, SAGA encourages ranking toward statutes that are better matched to the case context.

3.1 Task Formulation

We model a legal consultation as a sequence of conversational 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 consists of a fixed set of statutory articles $\mathcal{L} = \{\ell_1, \ldots, \ell_N\}$.

Our objective is to retrieve legally relevant statutes for each user turn. To prevent information leakage, retrieval input is restricted to user-side utterances. Let $H^u_{c,t} = \{u_1, \ldots, u_{t-1}\}$ denote the user-only history before turn t. Given the current user turn u_t and the preceding user-side context $H^u_{c,t}$, the system outputs a ranked list $\pi_{c,t}$ over the full statute set $\mathcal{L}$. During training and evaluation, a subset of statutes $\mathcal{L}^*_{c,t} \subseteq \mathcal{L}$ is annotated as the ground-truth support for turn (c, t). The goal is to rank the true relevant statutes in $\mathcal{L}^*_{c,t}$ as high as possible. For single-turn benchmarks, this formulation reduces to the special case where $H^u_{c,t}$ is empty.

3.2 Framework Overview

Figure 2 illustrates the overall SAGA framework. Given a legal query context, SAGA operates through three components.

Raw semantic retrieval. The first component captures broad semantic relatedness between the user query and the statute text. For conversational settings, the retrieval input is constructed only from user-side dialogue context, excluding assistant responses. A dense encoder maps the query and each statute into a shared embedding space, producing a raw semantic score that reflects overall textual compatibility.

Structural framing. In parallel, SAGA converts both sides of the retrieval problem into structured applicability-oriented representations. On the query side, it constructs $\mathrm{CAF}_{c,t}$ from user-side dialogue context to summarize dispute issues, core facts, and missing facts. On the statute side, each article $\ell \in \mathcal{L}$ is associated with a pre-computed $\mathrm{AAF}(\ell)$ that summarizes issue tags, prerequisites, legal effects, and evidence cues. These structured views make case-side and statute-side conditions more explicit than raw semantic matching alone.

Dual-stream fusion. Finally, SAGA combines the raw semantic signal and the structural signal through a zero-shot late-fusion design. The raw stream preserves flexible semantic matching from the original text, while the structured stream emphasizes case-side and statute-side applicability cues. Their fusion yields the final ranking score $S_{\text{final}}(c, t, \ell)$, which promotes statutes that are both semantically relevant and better aligned with the user's case context.

3.3 Case Applicability Frame (CAF)

In legal consultations, users rarely present a complete and well-structured case description in a single utterance. Instead, legally relevant information is often distributed across turns and expressed in

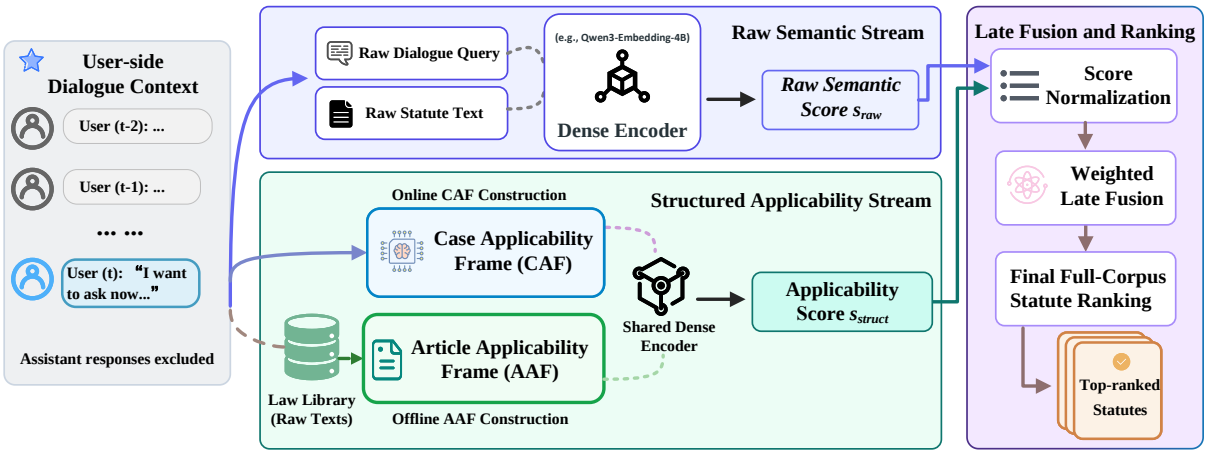

Figure 2: Overview of the SAGA framework. Under a unified full-corpus retrieval setting, SAGA combines two complementary retrieval streams. The raw semantic stream captures broad semantic relatedness between the user query and statute text, while the structural applicability stream captures structured case-side and statute-side cues through the dynamically generated CAF and the offline AAF. Their late fusion produces the final statute ranking.

colloquial, incomplete, or ambiguous language. Directly retrieving from raw dialogue may therefore overemphasize surface wording while overlooking the case elements that matter for statute retrieval.

To address this problem, SAGA constructs a CAF for each query context, as illustrated in Figure 3a. The CAF converts the current user turn together with the preceding user-side context into a structured case representation. In SAGA, the CAF represents the query side with three applicability cues: *dispute issues*, *core facts*, and *missing facts*. These cues capture what the user is asking about, what is already established in the dialogue, and what key information is still absent for legal evaluation.

By design, the CAF extractor is descriptive rather than normative. It does not cite statutes, assign legal responsibility, or make legal conclusions. Instead, it provides a structured summary of the user's factual situation and leaves the final ranking decision to the retrieval model.

3.4 Article Applicability Frame (AAF)

Raw statutory text is formal, compact, and often logically dense. A single article may intertwine issue scope, applicability conditions, legal consequences, and evidential cues in a highly compressed form. As a result, matching informal user dialogue directly against raw statutory text may capture topical similarity without adequately reflecting statute-side applicability structure.

To make this structure more explicit, SAGA constructs an AAF for each statute in the law library, as shown in Figure 3b. The AAF reorganizes each

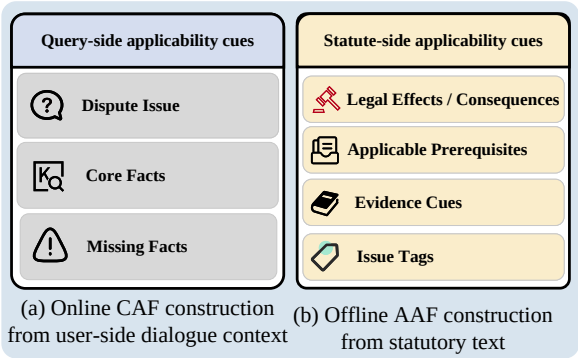

Figure 3: Structured applicability cues used in SAGA. (a) Query-side cues are constructed online from user-side dialogue context and include dispute issues, core facts, and missing facts. (b) Statute-side cues are constructed offline from statutory text and include issue tags, prerequisites, legal effects, and evidence cues.

article into a standardized statute-side representation that is easier to compare with the case-side CAF. In SAGA, the AAF represents the statute side with four applicability cues: *issue tags*, *prerequisites*, *legal effects*, and *evidence cues*. These cues summarize what the statute is about, when it applies, what consequences it triggers, and what kinds of factual support are typically relevant.

Unlike the CAF, which is generated dynamically from user-side dialogue context, the AAF is derived solely from statutory text and is therefore case-independent. This makes it reusable across consultations and allows the statute library to be indexed in a structured applicability-oriented form.

3.5 Hybrid Semantic-Structural Alignment

Dense retrieval models are effective at capturing broad semantic relatedness, but legal retrieval often requires more than semantic similarity alone. A statute may appear relevant at the textual level while still be poorly matched to the case context because its prerequisites are not satisfied. Conversely, purely structured representations make legal constraints more explicit, but may lose some of the softer contextual cues present in the original dialogue. SAGA therefore combines both views through a dual-stream late-fusion design.

Raw semantic stream. The first stream captures broad text-level relatedness between the user query and the statute text. Let $q_{c,t}$ denote the retrieval input for turn (c, t) and let ℓ denote a statute. Using a frozen dense encoder $E(\cdot)$, we compute the cosine similarity

$$s_{\text{raw}}(c, t, \ell) = \frac{E(q_{c,t}) \cdot E(\ell)}{\|E(q_{c,t})\| \, \|E(\ell)\|}.$$

This score reflects how well the raw query text and the raw statute text align in the embedding space.

Structural applicability stream. The second stream compares the query-side CAF and the statute-side AAF more explicitly. Although CAF and AAF are produced as JSON objects, the structured stream does not encode raw JSON syntax directly. Instead, we convert each structured representation into a field-labeled natural-language string by concatenating the relevant fields in a fixed order. For list-valued fields, extracted items are joined with delimiters, while empty fields are omitted. Concretely, the query-side string is formed from dispute issues, core facts, and missing facts, and the statute-side string is formed from issue tags, prerequisites, legal effects, and evidence cues. This produces a textual representation that preserves field identity while remaining compatible with the same dense encoder used in the raw stream.

The structural score is then computed as

$$s_{\text{struct}}(c, t, \ell) = \frac{E(\text{CAF}_{c,t}) \cdot E(\text{AAF}(\ell))}{\|E(\text{CAF}_{c,t})\| \, \|E(\text{AAF}(\ell))\|}.$$

This design avoids brittle manually engineered slot-matching rules while still allowing the encoder to capture structured relationships between case-side conditions and statute-side cues.

Late fusion. Finally, SAGA combines the two streams into a unified ranking score. Because the raw and structural streams may follow different score distributions, we first apply per-query Min–Max normalization to obtain $\tilde{s}_{\text{raw}}$ and $\tilde{s}_{\text{struct}}$. The final score is computed as

$$S_{\text{final}} = \alpha \cdot \tilde{s}_{\text{raw}} + (1 - \alpha) \cdot \tilde{s}_{\text{struct}},$$

where $\alpha \in [0, 1]$ controls the balance between broad semantic relevance and structured applicability-oriented cues. This design allows SAGA to preserve the flexibility of dense semantic retrieval while incorporating structured case-side and statute-side information into the final ranking.

4 Experiments

4.1 Experimental Setup

Datasets and evaluation. We evaluate SAGA on two Chinese statute retrieval benchmarks. LexRAG (Li et al., 2025) is a multi-turn legal consultation dataset with turn-level relevance judgments over supporting statutory articles. We follow its official train/dev/test split and report final results on the test set. STARD (Su et al., 2024) is a Chinese statute retrieval benchmark with question–statute relevance annotations. Following its released train/dev split, we use the training split for selection and report final results on the official dev set. We report Hit@k (H@k), Recall@k (R@k), nDCG@k (N@k), and Mean Reciprocal Rank (MRR). In the main table, we emphasize N@10 and MRR as the primary ranking-quality metrics, and additionally report H@50 and R@50 to reflect medium-depth coverage.

Unified full-corpus retrieval. All methods in the main comparison directly rank the entire statute library rather than reranking a separately retrieved candidate pool. For dense methods, we encode the full corpus and perform exact nearest-neighbor search with FAISS IndexFlatIP (Johnson et al., 2019) over ℓ_2-normalized embeddings. On LexRAG, baselines without structured cues use the current user turn as the retrieval query, and assistant responses are excluded from all retrieval inputs. History-aware and other multi-view fusion variants are analyzed separately in Section 4.3. On STARD, each input question is used directly as the retrieval query.

Structured representations. For LexRAG, we construct a CAF from user-side dialogue con-

Method	LexRAG (Test)				STARD (Official Dev)			
	N@10	MRR	H@50	R@50	N@10	MRR	H@50	R@50
BM25	0.1173	0.1026	0.3543	0.3349	0.3157	0.3368	0.6364	0.5515
BGE-large-zh-v1.5	0.1842	0.1625	0.5052	0.4822	0.3779	0.4008	0.7890	0.6997
Qwen3-Embedding-0.6B	0.2801	0.2567	0.5954	0.5741	0.5292	0.5512	0.8831	0.8075
Qwen3-Embedding-4B	0.3504	0.3246	0.6646	0.6466	0.5862	0.6037	0.9416	0.8692
HyDE + Qwen3-Embedding-4B	0.3173	0.2863	0.6667	0.6445	0.5698	0.5905	0.9318	0.8674
SAGA (ours)	**0.3696**†	**0.3410**	**0.7086**†	**0.6916**†	**0.5940**	**0.6190**	**0.9481**	**0.8781**

Table 1: Main results under a unified full-corpus retrieval protocol. We report N@10 and MRR as the primary ranking-quality metrics, together with H@50 and R@50 as medium-depth coverage metrics. For SAGA, the fusion weight α is tuned on the LexRAG development split and the STARD training split, yielding α=0.6 for LexRAG and α=0.8 for STARD. HyDE uses gpt-4o-mini to generate hypothetical documents and Qwen3-Embedding-4B as the retrieval backbone. † indicates a statistically significant improvement over Qwen3-Embedding-4B under a paired sign-flip randomization test ($p < 0.05$). Full significance results are reported in Appendix K.

text up to the current turn, excluding assistant responses. For both LexRAG and STARD, each statute is paired with an offline AAF extracted once from the statute text. CAF and AAF are generated with fixed prompts using gpt-4o-mini and cached for reuse. In SAGA, the query-side structured representation uses dispute issues, core facts, and missing facts, while the statute-side structured representation uses issue tags, prerequisites, legal effects, and evidence cues. The analyses in Section 4.3 examine these components as diagnostic studies of the method rather than as selection on the final reporting splits.

Baselines and implementation. We compare SAGA against BM25, two general dense retrievers (BGE-large-zh-v1.5 (Xiao et al., 2024) and Qwen3-Embedding-0.6B), a stronger dense backbone (Qwen3-Embedding-4B), and an LLM-enhanced baseline, HyDE (Gao et al., 2023), implemented on top of Qwen3-Embedding-4B (Zhang et al., 2025b). For HyDE, hypothetical documents are generated with gpt-4o-mini. SAGA combines a raw dense retrieval stream with a structured applicability-oriented stream derived from CAF and AAF. The fusion weight α is selected on the LexRAG development split and the STARD training split, yielding α=0.6 for LexRAG and α=0.8 for STARD; sensitivity curves are provided in Appendix H.

4.2 Main Results

Table 1 reports the main results under a unified full-corpus retrieval protocol on LexRAG and STARD. Overall, SAGA achieves the best performance among the baselines in the main table on both benchmarks, with clearer gains on LexRAG. This suggests that structured applicability-oriented cues are especially helpful in multi-turn legal retrieval, where the current question is often underspecified unless interpreted together with the user's broader case context.

Comparison with strong baselines. Among the baselines, Qwen3-Embedding-4B is the strongest raw dense retriever, while HyDE + Qwen3-Embedding-4B is the strongest LLM-enhanced expansion baseline. On LexRAG, SAGA improves over Qwen3-Embedding-4B from 0.3504 to 0.3696 on N@10 and from 0.3246 to 0.3410 on MRR. On STARD, the corresponding gains are from 0.5862 to 0.5940 on N@10 and from 0.6037 to 0.6190 on MRR. SAGA also remains above HyDE + Qwen3-Embedding-4B on both datasets, indicating that generic query expansion alone is not sufficient to account for the observed gains. Taken together, these results suggest that a strong dense retriever can still benefit from structured case-side and statute-side cues beyond raw semantic matching alone.

Coverage and cross-dataset pattern. The gains of SAGA are not limited to top-ranked ordering. It also achieves the best H@50 and R@50 on both datasets. On LexRAG, H@50 and R@50 improve from 0.6646 and 0.6466 with Qwen3-Embedding-4B to 0.7086 and 0.6916, while on STARD they improve from 0.9416 and 0.8692 to 0.9481 and 0.8781. This indicates that SAGA improves not only the very top of the ranking but also medium-depth coverage of relevant statutes. The larger gains on LexRAG are consistent with the conversational nature of the benchmark: multi-turn legal

Structured Variant	LexRAG Dev N@10
Structured stream used by SAGA	**0.2651**
w/o issue/fact–prerequisite	0.1352
w/o missing/evidence	0.2530

Table 2: Field-group ablation on the structured stream used in SAGA, reported on LexRAG Dev. We remove one structured cue group at a time to examine its contribution. This analysis is diagnostic and is not used for model selection on the final reporting split.

consultation often requires recovering case conditions that are only partially expressed in the raw query, whereas the raw dense baseline on STARD is already very strong.

Statistical significance. We further verify the main gains with paired sign-flip randomization tests over per-query scores. On LexRAG, SAGA significantly outperforms Qwen3-Embedding-4B on N@10, H@50, and R@50, and significantly outperforms HyDE + Qwen3-Embedding-4B on all four reported metrics. On STARD, the gain over HyDE + Qwen3-Embedding-4B remains significant on N@10, while the improvements over Qwen3-Embedding-4B are smaller and not statistically significant. Detailed results are reported in Appendix K.

While these results establish the overall effectiveness of SAGA, they do not yet reveal which structured cues matter most, whether the gains could be explained by simpler alternatives such as conversational history or generic reformulation, or how the two retrieval streams should be balanced. We investigate these questions next.

4.3 Ablation and Analysis

To better understand where the gains of SAGA come from, we analyze two aspects of the method: which structured cue groups matter most, and whether the gains can be explained by simpler alternatives such as added dialogue history, generic LLM reformulation, or single-side structuring.

Field-group ablation. Table 2 reports field-group ablations on the structured stream used by SAGA. This analysis should be read as a diagnostic study of the current method rather than a model-selection procedure. On LexRAG Dev, removing the issue/fact–prerequisite group causes a much larger drop than removing the missing/evidence group, indicating that explicit alignment between case-side dispute issues and core facts and statute-side issue tags and prerequisites is the dominant source of structured retrieval gains. The missing/evidence group is also useful, but its contribution is clearly secondary. Overall, these results suggest that the gains of SAGA rely primarily on the issue/fact–prerequisite cues, while the missing/evidence cues provide a smaller but still useful contribution.

Variant	N@10	MRR
Context-only controls		
Current-turn query	0.3504	0.3246
History-aware query	0.3481	0.3211
Current + History RRF	0.3503	0.3244
Generic LLM reformulation / expansion		
Query rewriting	0.2419	0.2151
Summary generation	0.2705	0.2347
HyDE expansion	0.3173	0.2863
Applicability-oriented variants		
CAF + statute text	0.3428	0.3105
Query text + AAF	0.3025	0.2772
SAGA	**0.3696**	**0.3410**

Table 3: LexRAG test results for context-only controls, generic LLM reformulation/expansion, and applicability-oriented variants. The first two groups retrieve against raw statute text. HyDE represents a generic query-expansion baseline, while CAF + statute text isolates query-side applicability-oriented structuring. The best results are obtained by SAGA, which combines the raw semantic stream with two-sided CAF/AAF alignment.

History, generic LLM reformulation/expansion, and applicability-oriented variants. We further evaluate whether the gains of SAGA can be explained by simpler alternatives such as adding more dialogue history, generic LLM reformulation, generic query expansion, or single-side structuring. As shown in Table 3, replacing the current-turn query with a history-aware query or fusing current-turn and history-aware retrieval with RRF brings almost no improvement over the raw dense baseline. This suggests that the gain of SAGA cannot be explained by added dialogue history alone. We also compare against three generic LLM-based query-side baselines, namely query rewriting, natural-language summary generation, and HyDE expansion. All three remain below the raw dense baseline, indicating that simply rewriting, summarizing, or hypothetically expanding the dialogue is not sufficient for applicability-aware legal retrieval. Among the applicability-oriented variants, CAF + statute text is a strong baseline and

clearly outperforms HyDE, showing that query-side applicability-oriented structuring is more effective than generic query expansion in this setting. However, SAGA still achieves clearly higher results on both N@10 and MRR. By contrast, Query text + AAF is weaker than CAF + statute text, suggesting that query-side case structuring is more important than statute-side abstraction alone in conversational settings. Taken together, these results suggest that the gains of SAGA are better explained by combining query-side and statute-side structured cues than by added context, generic LLM reformulation, generic query expansion, or single-side structuring alone.

Preliminary applicability-oriented manual analysis. To obtain a more direct view of the ranking behavior encouraged by SAGA, we conduct a preliminary manual analysis on 100 sampled LexRAG test queries. Compared with Qwen3-Embedding-4B, SAGA increases Applicable Precision@10 from 33.7% to 37.1% and reduces the proportion of clearly irrelevant statutes in the top-10 results from 25.5% to 19.1%. At the same time, the rate of topically related but inapplicable statutes does not decrease, rising slightly from 36.1% to 38.3%, indicating that SAGA does not fully eliminate near-miss retrievals that remain semantically related but ultimately fail to satisfy all applicability conditions. A query-level comparison shows a similar pattern: SAGA achieves a higher Applicable Precision@10 on 37 of the 100 sampled queries, compared with 26 losses and 37 ties. Moreover, among the documents newly introduced by SAGA into the top-10, applicable statutes are more frequent (130 vs. 96), while clearly irrelevant statutes are much less frequent (87 vs. 151), than among the documents uniquely retained by the baseline. Taken together, these findings suggest that SAGA shifts ranking toward more applicable and less clearly irrelevant statutes, rather than functioning as a strict applicability verifier. Figure 4 visualizes the overall label composition of the top-10 results. Detailed annotation settings and full results are reported in Appendix G.

Practical overhead. SAGA is a training-free framework, so its main additional online cost comes from CAF generation rather than model training. In our latency study on 100 sampled LexRAG queries, online CAF generation with gpt-4o-mini takes 5.09 seconds per query on average, with a p50 of 4.84 seconds and a p95 of 7.15 seconds. The corresponding average token cost is only $0.000212 per query, indicating that the main practical overhead of SAGA lies in latency rather than monetary cost. Since AAFs are constructed offline and cached for reuse, this overhead is incurred only on the query side. Detailed latency and token-cost results are reported in Appendix E.

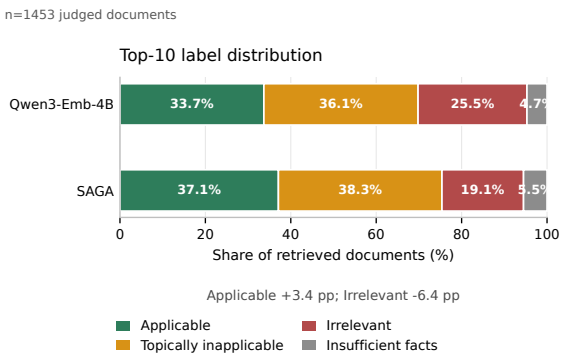

Figure 4: Applicability-oriented manual analysis on 100 sampled LexRAG test queries. Compared with Qwen3-Embedding-4B, SAGA increases the share of applicable statutes and reduces the share of clearly irrelevant statutes in the top-10 results, although topically related but inapplicable near-miss results still remain.

Additional sensitivity results for the fusion weight are provided in Appendix H.

5 Conclusion

We presented SAGA, a training-free legal retrieval framework that incorporates structured applicability-oriented cues into dense retrieval. By combining a raw semantic retrieval stream with a structural stream derived from CAF and AAF, SAGA helps bridge the gap between colloquial case descriptions and formal statutory language. Under a unified full-corpus retrieval setting, experiments on LexRAG and STARD show that SAGA yields its clearest gains in conversational legal retrieval. In particular, SAGA achieves significant improvements over strong baselines on LexRAG, while providing more modest gains on STARD. Taken together with our additional analyses, these results suggest that structured case-side and statute-side cues provide useful retrieval signals beyond raw semantic matching alone, especially in multi-turn legal consultations where the user's case context is only partially revealed in dialogue.

Limitations

SAGA has two main limitations. First, the structural stream depends on the quality of CAF and AAF extraction. Although the late-fusion design reduces over-reliance on structured representations alone, extraction errors or omissions from the underlying LLM can still affect retrieval quality. Our manual audits suggest that LLM-based structuring is generally reliable and rarely hallucinatory, but it is not always perfect in capturing all legally relevant details. In particular, some cases still exhibit incomplete prerequisite extraction or less precise representation of legal effects, which may weaken retrieval in difficult examples. Our preliminary applicability-oriented manual analysis further suggests that the current framework is more effective at reducing clearly irrelevant statutes than at fully resolving fine-grained applicability errors among semantically related candidates. Second, our experiments focus on two Chinese statute retrieval benchmarks. Further validation is still needed across other languages, legal domains, and legal traditions, especially in settings where precedent and case law play a more central role.

References

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 the 22nd international conference on Machine learning*, pages 89–96.

Christopher Burges, Robert Ragno, and Quoc Le. 2006. Learning to rank with nonsmooth cost functions. *Advances in neural information processing systems*, 19.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 2898–2904.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. Lexglue: A benchmark dataset for legal language understanding in english. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2021. Cast 2020: The conversational assistance track overview. In *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 the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5918–5924.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE transactions on big data*, 7(3):535–547.

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 the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 6769–6781.

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. *Advances in neural information processing systems*, 30.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Haitao Li, Yifan Chen, Hu YiRan, 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. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3606–3615.

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

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Information Retrieval*, 3(4):333–389.

Weihang Su, Yiran Hu, Anzhe Xie, Qingyao Ai, Quezi Bing, Ning Zheng, Yun Liu, Weixing Shen, and Yiqun Liu. 2024. Stard: A chinese statute retrieval dataset derived from real-life queries by non-professionals. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10658–10671.

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

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Kepu Zhang, Weijie Yu, Sunhao Dai, and Jun Xu. 2025a. Citalaw: Enhancing llm with citations in legal domain. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11183–11196.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A Implementation Details

Retrieval setting. All retrieval experiments in the main paper are conducted under a unified full-corpus retrieval protocol. Assistant responses are excluded from retrieval inputs. For LexRAG, the raw retrieval query is the current user turn, and the corresponding structured query is constructed from the user-side dialogue context up to the current turn. For STARD, each question is used directly as the raw retrieval query, and the corresponding structured query is derived from its generated case-side representation.

Dense retrieval and indexing. For dense retrieval methods, we encode the full statute library and perform exact nearest-neighbor search with FAISS IndexFlatIP over ℓ_2-normalized embeddings. The raw dense baselines rank statutes by cosine similarity between the raw query text and the raw statute text. For the structured stream in SAGA, CAF and AAF are not encoded as raw JSON strings. Instead, the structured representations are linearized into field-labeled text in a fixed order, with list-valued fields concatenated by delimiters and empty fields omitted. The resulting linearized texts are then encoded with the same dense backbone and ranked by cosine similarity.

CAF and AAF construction. CAF and AAF are generated with gpt-4o-mini and cached for reuse. CAF is extracted only from user-side inputs and does not access gold statutes or gold answers. Although the extraction prompts produce JSON outputs, the structured retrieval stream uses their linearized text form rather than raw JSON syntax. In the current formulation of SAGA, the case-side structured text is formed from dispute issues, core facts, and missing facts, while the statute-side structured text is formed from issue tags, prerequisites, legal effects, and evidence cues.

Baselines. The main comparison includes one sparse baseline (BM25), two general dense baselines (BGE-large-zh-v1.5 and Qwen3-Embedding-0.6B), one stronger dense backbone (Qwen3-Embedding-4B), and one LLM-enhanced retrieval baseline (HyDE + Qwen3-Embedding-4B). HyDE uses hypothetical documents generated by gpt-4o-mini.

SAGA implementation. SAGA is a training-free dual-stream retrieval framework. Its raw semantic stream computes cosine similarity between the raw query and the raw statute text using Qwen3-Embedding-4B. Its structured stream computes cosine similarity between the linearized CAF and AAF representations using the same encoder. The two streams are combined by per-query Min–Max normalization followed by weighted late fusion. The fusion weight α is selected on validation data, yielding $\alpha = 0.6$ on LexRAG and $\alpha = 0.8$ on STARD.

Table 4 provides a compact summary of the key implementation choices used in the main experiments. We use the same dense retrieval inference settings across all Qwen-based runs, including fixed maximum input length, batch size, and GPU precision; the exact runtime configuration and software versions will be released with the code.

Component	Setting
Retrieval protocol	Unified full-corpus retrieval; assistant responses excluded from retrieval inputs.
Dense backbone	Qwen3-Embedding-4B for both the raw semantic stream and the structured stream; exact search with FAISS IndexFlatIP over ℓ_2-normalized embeddings.
CAF / AAF construction	Generated with gpt-4o-mini and cached for reuse. The prompts produce structured JSON outputs, which are converted into field-labeled text before encoding.
Structured stream encoding	CAF and AAF are not encoded as raw JSON strings. Instead, the structured representations are linearized in a fixed field order, with list-valued fields concatenated by delimiters and empty fields omitted. In SAGA, the case-side text is formed from dispute issues, core facts, and missing facts, while the statute-side text is formed from issue tags, prerequisites, legal effects, and evidence cues.
Baselines	BM25, BGE-large-zh-v1.5, Qwen3-Embedding-0.6B, Qwen3-Embedding-4B, and HyDE + Qwen3-Embedding-4B.
Fusion	Per-query Min–Max normalization followed by weighted late fusion; $\alpha = 0.6$ on LexRAG and $\alpha = 0.8$ on STARD.

Table 4: Key implementation details used in the main experiments.

B Qrels and Evaluation Coverage

All retrieval metrics in the main text are computed on the normalized qrels files used in our pipeline. For LexRAG, the effective numbers of evaluated queries are 3817 for Train, 487 for Dev, and 477 for Test. For STARD, evaluation follows the released train/dev split, and the official dev set is used for final reporting.

For significance testing, per-query scores are computed on the intersection of qids shared by the compared runs and the corresponding qrels split. In practice, coverage differences among the compared full-corpus systems are negligible, so this common-qid restriction has minimal impact on the reported results.

C Manual Inspection of CAF Quality

To assess the reliability of the query-side structured representation, we manually inspect 100 sampled compact CAFs from LexRAG. The results indicate that the extracted CAFs are generally reliable. For faithfulness, 87% of samples are rated as perfect and the remaining 13% as good, with no bad cases. Hallucination is rare, with a hallucination-free rate of 99%. For completeness and retrieval usefulness, the perfect rates are lower (59% and 55%, respectively), but the combined perfect-or-good rates remain high. This suggests that the main limitation of CAF extraction is not factual fabrication or unusable output, but that some cases are only partially complete or partially optimized for retrieval.

Metric	Perfect	Good	Bad
Faithfulness	87	13	0
Completeness	59	38	3
Usefulness	55	42	3
Hallucination-free		99%	
Hallucination rate		1%	

Table 5: Manual evaluation of 100 sampled compact CAFs from LexRAG. Faithfulness is high and hallucination is rare, while incompleteness is more common than outright factual fabrication.

D Manual Inspection of AAF Quality

To assess the reliability of the statute-side structured representation, we manually inspect 100 sampled compact AAFs from the law library. The results indicate that the extracted AAFs are generally reliable and closely grounded in the original statute text. For faithfulness, 81% of samples are rated as perfect and the remaining 19% as good, with no bad cases. Hallucination is also relatively rare, with a hallucination-free rate of 97%. For prerequisite completeness and legal-effect correctness, the perfect rates are lower (32% and 40%, respectively), but the combined perfect-or-good rates remain high. These results suggest that the main limitation of AAF extraction is not that the outputs are unusable, but that some statute-side representations are not fully complete or fully precise in difficult cases, especially when statutory prerequisites are implicit or legal effects are densely expressed.

Taken together, the CAF and AAF audits suggest that LLM-based structuring in SAGA is generally

Metric	Perfect	Good	Bad
Faithfulness	81	19	0
Prerequisite completeness	32	63	5
Legal-effect correctness	40	54	6
Hallucination-free		97%	
Hallucination rate		3%	

Table 6: Manual evaluation of 100 sampled compact AAFs from the law library. Faithfulness is generally high and hallucination is relatively rare, while the more common weakness lies in incomplete prerequisite extraction and partial loss of legal-effect precision.

reliable and rarely hallucinatory, while its remaining weakness lies mainly in producing outputs that are sometimes good rather than fully perfect in coverage or precision.

E Latency and Token Cost of Online CAF Generation

To assess the main additional online overhead introduced by SAGA, we measure the latency and token cost of CAF generation on 100 sampled LexRAG queries using gpt-4o-mini. The average latency is 5.09 seconds per query, with a p50 of 4.84 seconds and a p95 of 7.15 seconds. The average token usage is 791.92 input tokens and 144.47 output tokens, corresponding to 936.39 total tokens per query. Under the ChatAnywhere pricing used in our experiments, this yields an average cost of $0.000212 per query. These results suggest that the main practical overhead of SAGA lies in online latency rather than token cost.

Metric	Value
Avg latency / query	5.09 s
P50 latency	4.84 s
P95 latency	7.15 s
Avg input tokens	791.92
Avg output tokens	144.47
Avg total tokens	936.39
Avg cost / query	$0.000212

Table 7: Latency and token-cost analysis for online CAF generation with gpt-4o-mini on 100 sampled LexRAG queries.

F Additional Analysis of Conversational History and Simple Fusion

Is conversational history sufficient? A natural question is whether the gain of SAGA on LexRAG simply comes from using more dialogue context. To test this, we compare the raw dense retriever with a history-aware version and with a simple RRF fusion of current-turn and history-aware runs. As shown in Table 8, neither replacing the current-turn query with a history-aware query nor fusing the two views yields a meaningful improvement over the raw dense baseline. This suggests that the advantage of SAGA does not come merely from adding more dialogue history, but from introducing a structured applicability-oriented view.

Does naive fusion suffice? We also examine whether simple multi-view fusion can already account for the observed gains. The answer is largely negative. On LexRAG, neither Current+History RRF nor raw+HyDE fusion produces a meaningful gain over the raw dense retriever, indicating that naive semantic-view fusion is insufficient in the conversational setting. On STARD, raw+HyDE fusion is a particularly strong competitor, showing that query expansion is highly effective for single-turn statute retrieval. However, broader or less targeted fusion is not uniformly reliable. In contrast, SAGA uses a more targeted structured view derived from applicability cues, which provides a clearer and more stable improvement signal than simply adding more semantic retrieval views.

Variant	N@10	MRR
Current-turn dense retrieval	0.3504	0.3246
History-aware dense retrieval	0.3481	0.3211
Current + History RRF	0.3503	0.3244
SAGA	**0.3696**	**0.3410**

Table 8: Does conversational history alone explain the gain on LexRAG? Simply replacing the current-turn query with a history-aware query, or fusing the two with RRF, brings almost no improvement over the raw dense retriever.

G Applicability-Oriented Manual Analysis

To obtain a more direct view of the ranking behavior encouraged by SAGA, we conduct a preliminary manual analysis on 100 sampled queries from the LexRAG test set. For each query, we inspect the union of the baseline top-10 results from Qwen3-Embedding-4B and the top-10 results from SAGA, and annotate each retrieved statute using four labels: *Applicable*, *Topically related but inapplicable*, *Irrelevant*, and *Insufficient facts*.

This analysis is exploratory and author-reviewed. The annotations were carried out by two graduate-student reviewers, with GPT-5.5 used only as an auxiliary reference during the review process rather than as the final decision maker. Where disagreements arose, the final label was determined by discussion between the two reviewers. The goal of this analysis is not to establish a formal legal evaluation benchmark, but to provide additional evidence about the ranking tendencies encouraged by SAGA.

Table 9 shows that SAGA improves Applicable Precision@10 from 33.7% to 37.1% and reduces the proportion of clearly irrelevant statutes from 25.5% to 19.1%. At the same time, the proportion of topically related but inapplicable statutes does not decrease, rising slightly from 36.1% to 38.3%. This suggests that the main effect of SAGA is to replace many clearly irrelevant retrievals with statutes that are more often applicable, rather than to fully resolve fine-grained applicability errors.

A query-level comparison shows a similar pattern: SAGA achieves a higher Applicable Precision@10 than the baseline on 37 of the 100 sampled queries, compared with 26 losses and 37 ties. Moreover, among the documents newly introduced by SAGA into the top-10, applicable statutes are more frequent than among the documents uniquely retained by the baseline (130 vs. 96), while clearly irrelevant statutes are much less frequent (87 vs. 151). Overall, this exploratory analysis is consistent with the main-text claim that SAGA shifts ranking toward more applicable and less clearly irrelevant statutes, while also suggesting that further work is needed to better suppress topically related but ultimately inapplicable statutes.

Metric	Baseline	SAGA
Applicable Precision@10	33.7%	**37.1%**
Topically inapplicable Rate@10	**36.1%**	38.3%
Irrelevant Rate@10	25.5%	**19.1%**
Insufficient Facts Rate@10	4.7%	5.5%
Applicable Precision wins / ties / losses	37 / 37 / 26	
Lower inapplicable rate wins / ties / losses	38 / 18 / 44	

Table 9: Preliminary applicability-oriented manual analysis on 100 sampled LexRAG test queries. Compared with Qwen3-Embedding-4B, SAGA increases the share of applicable statutes in the top-10 and reduces clearly irrelevant statutes, although it does not reduce the rate of topically related but inapplicable statutes.

Label	SAGA-only	Baseline-only
Applicable	**130**	96
Topically inapplicable	206	184
Irrelevant	**87**	151
Insufficient facts	30	22

Table 10: Label distribution of changed top-10 documents in the applicability-oriented manual analysis. SAGA introduces more applicable statutes and fewer clearly irrelevant statutes than the baseline, but also retains many topically related yet inapplicable near-miss results.

H Fusion-Weight Sensitivity

We examine the sensitivity of SAGA to the fusion weight α in

$$S_{\text{final}} = \alpha \cdot \tilde{s}_{\text{raw}} + (1 - \alpha) \cdot \tilde{s}_{\text{struct}},$$

where α controls the contribution of the raw semantic stream and $(1 - \alpha)$ controls the contribution of the structured stream. Because SAGA is a zero-shot, training-free method, this analysis is used only for weight selection and does not involve model training. Accordingly, α is tuned on a held-out selection split rather than on the final reporting split: LexRAG uses the development split, while STARD uses the training split and reserves the official dev split for final evaluation.

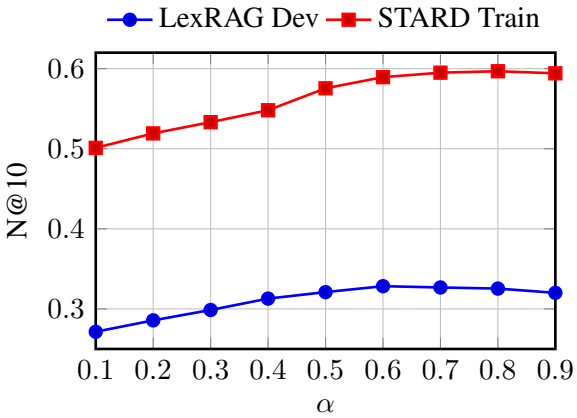

Figure 5: Sensitivity of the fusion weight α in the zero-shot fusion score $S_{\text{final}} = \alpha\tilde{s}_{\text{raw}} + (1-\alpha)\tilde{s}_{\text{struct}}$, where α weights the raw semantic stream and $(1-\alpha)$ weights the structured stream. The best selection results are obtained at $\alpha = 0.6$ on LexRAG Dev (0.6/0.4) and $\alpha = 0.8$ on STARD Train (0.8/0.2).

As shown in Figure 5, the best selection result is obtained at $\alpha = 0.6$ on LexRAG, corresponding to a 0.6/0.4 balance between the raw and structured streams, and at $\alpha = 0.8$ on STARD, corresponding to a 0.8/0.2 balance. The curves remain relatively stable around their optima, suggesting that SAGA is not highly brittle to small changes in the fusion weight.

I Prompt Templates

This appendix reports the original prompt templates used in our experiments. For readability, we provide both English summaries and Chinese versions. The Chinese prompts are the primary source of truth, while the English versions are provided for presentation only.

For transparency, we report the extraction and query-generation prompts as they were used in our pipeline. These prompts define the JSON outputs or text generations used by the corresponding retrieval variants described in the main text.

I.1 English Prompt Templates

Template 1: CAF Extraction Prompt

Note: This is the original extraction prompt used to generate CAF outputs in our pipeline.
System Prompt:
You are a legal consultation structure extractor. Your task is to extract a Case Applicability Frame (CAF), i.e., a structured representation of the user's case situation, based only on the user's current query and dialogue history. Do not answer the legal question.
User Prompt Template:
[Input Information]
Case Type: {case_type}
History: {history_queries_block}
Current: {current_query}
[Extraction Requirements]

1. Extract strictly from the user input. Do not use external knowledge.
2. Do not provide legal advice, article names, or judgments.
3. Put important but currently unspecified information into missing_facts.
4. If uncertain, prefer [] or "unknown".
5. Output a JSON object only.

[JSON Output Template]

```
{
  "case_type": "",
  "parties_roles": [],
  "legal_relations": [],
  "core_facts": [],
  "dispute_issues": [],
  "remedy_goals": [],
  "procedural_stage": "",
 "contract_or_condition_clauses": [],
  "exception_signals": [],
  "missing_facts": []
}
```

Template 2: AAF Extraction Prompt

Note: This is the original extraction prompt used to generate AAF outputs in our pipeline.
System Prompt:
You are a statute structure extractor. Your task is to extract an Article Applicability Frame (AAF), i.e., a structured representation of how a statute applies, based only on the provided statute text.
User Prompt Template:
[Input Information]
Name: {article_name}
Content: {article_content}
[Extraction Requirements]

1. Rely solely on the provided statute text.

2. Do not infer legal content that is not supported by the text.

3. Use [] for unmentioned exceptions or procedural stages.

4. Output a JSON object only.

[JSON Output Template]

```
{
  "article_id": "",
  "law_domain": "",
  "target_subject": [],
  "governed_relations": [],
  "prerequisites": [],
  "exceptions": [],
  "procedural_scope": [],
  "legal_effects": [],
  "issue_tags": [],
  "evidence_cues": []
}
```

Template 3: Query Rewriting Prompt

System Prompt:
You are a legal query rewriter. Your task is not to answer the question, but to rewrite a multi-turn legal consultation into a single standalone query for statute retrieval.
User Prompt Template:
[Input Information]
Case Type: {case_type}
History: {history_queries_block}
Current: {current_query}
[Rewriting Requirements]

1. Rewrite strictly based on the user input. Do not add facts that are not explicitly stated.

2. Do not provide legal advice, article names, or explanations.

3. Preserve key legally relevant facts, dispute points, and relations whenever possible.

4. Output only one rewritten Chinese retrieval query, without quotation marks, numbering, or extra text.

Template 4: Summary Generation Prompt

System Prompt:
You are a legal retrieval summarizer. Your task is not to answer the question and not to output structured fields, but to compress a multi-turn legal consultation into a single natural-language summary query for statute retrieval.
User Prompt Template:
[Input Information]
Case Type: {case_type}
History: {history_queries_block}
Current: {current_query}
[Summarization Requirements]

1. Summarize strictly based on the user input. Do not add facts that are not explicitly stated.

2. Do not provide legal advice, article names, or responsibility judgments.

3. Preserve key legally relevant information whenever possible, including party relations, core facts, dispute points, and known conditions.

4. Output a single concise natural-language Chinese retrieval summary only.

5. Do not use JSON, numbering, bullet points, or schema-style field names.

Template 5: HyDE Prompt

System Prompt:
You are a legal retrieval assistant. Given a user's legal question, generate a concise and credible hypothetical document that resembles relevant legal text or a legal analysis summary, in order to support downstream statute retrieval. Do not output explanations or prefaces.
User Prompt Template:
Please generate a hypothetical document that would help retrieve relevant statutes for the following legal question.
Requirements:

1. Use Chinese.

2. The content should stay close to legal concepts, legal relations, elements of liability, responsibility, or remedies.

3. Do not fabricate specific statute numbers.

4. Keep it concise, around 120–220 Chinese characters.

Question:
{query}

I.2 Chinese Prompt Templates

模板一：CAF 抽取Prompt

Note: This is the original extraction prompt used to generate CAF outputs in our pipeline.
System Prompt:
你是一个法律咨询结构抽取器。你的任务是根据用户当前问题及历史提问，抽取一个“案件适用框架（Case Applicability Frame, CAF）”，即用于表示用户案件情况的结构化表示。不要回答法律问题。
User Prompt Template:
【输入信息】
案件类型: {case_type}
历史提问: {history_queries_block}
当前提问: {current_query}
【抽取要求】

1. 仅依据用户输入抽取，不要使用外部知识。
2. 不要输出法律建议、法条名称或责任判断。
3. 对重要但尚未明确的信息，放入missing_facts。
4. 不确定时优先输出[] 或“未知”。
5. 仅输出JSON 对象。

【输出**JSON** 模板】

```
{
  "case_type": "",
  "parties_roles": [],
  "legal_relations": [],
  "core_facts": [],
  "dispute_issues": [],
  "remedy_goals": [],
  "procedural_stage": "",
 "contract_or_condition_clauses": [],
  "exception_signals": [],
  "missing_facts": []
}
```

模板二：AAF 抽取Prompt

Note: This is the original extraction prompt used to generate AAF outputs in our pipeline.
System Prompt:
你是一个法条结构抽取器。你的任务是仅根据给定法条正文，抽取一个“法条适用框架（Article Applicability Frame, AAF）”，即用于表示法条适用条件和法律效果的结构化表示。
User Prompt Template:
【输入信息】
法条名称: {article_name}
法条正文: {article_content}
【抽取要求】

1. 仅依据给定法条正文抽取。
2. 不要加入原文未支持的法律内容。
3. 对未明确出现的例外或程序范围输出[]。
4. 仅输出JSON 对象。

【输出**JSON** 模板】

```
{
  "article_id": "",
  "law_domain": "",
  "target_subject": [],
  "governed_relations": [],
  "prerequisites": [],
  "exceptions": [],
  "procedural_scope": [],
  "legal_effects": [],
  "issue_tags": [],
  "evidence_cues": []
}
```

模板三：Query Rewriting Prompt

System Prompt:
你是一个法律检索查询重写器。你的任务不是回答问题，而是将多轮法律咨询内容重写为一条适合法条检索的单轮查询。
User Prompt Template:
【输入信息】
案件类型: {case_type}
历史提问: {history_queries_block}
当前提问: {current_query}
【重写要求】

1. 仅依据用户输入重写，不要加入输入中未明确出现的事实。

2. 不要输出法律建议、法条名称或解释性文字。
3. 尽量保留与法律适用相关的关键事实、争议点和关系。
4. 仅输出一条中文检索查询，不要输出引号、编号或额外说明。

模板四：Summary Generation Prompt

System Prompt:
你是一个法律检索摘要器。你的任务不是回答问题，也不是输出结构化字段，而是将多轮法律咨询内容压缩为一条适合法条检索的自然语言摘要查询。
User Prompt Template:
【输入信息】
案件类型: {case_type}
历史提问: {history_queries_block}
当前提问: {current_query}
【摘要要求】

1. 仅依据用户输入进行摘要，不要添加未明确出现的事实。
2. 不要输出法律建议、法条名称或责任判断。
3. 尽量保留与法条适用相关的关键信息，包括当事关系、核心事实、争议点和已知条件。
4. 仅输出一条简洁、自然的中文检索摘要。
5. 不要使用JSON、编号、项目符号或schema 风格字段名。

模板五：HyDE Prompt

System Prompt:
你是一个法律检索辅助器。请根据用户问题，生成一段简洁、可信、像相关法律条文或法律分析摘要的“假设性文档”，用于帮助后续检索相关法条。不要输出解释，不要说“以下是”，直接输出正文。
User Prompt Template:
请为下面这个法律问题生成一段有助于检索相关法条的假设性文档。
要求:

1. 使用中文。
2. 内容应贴近法律概念、法律关系、构成要件、责任或救济方式。
3. 不要编造具体法条编号。
4. 保持简洁，控制在120–220 字左右。

问题:
{query}

J Case Study: Tax Rule Specialization in Conversational Statute Retrieval

Case motivation. This case study illustrates a setting where the user gradually narrows a tax-law consultation from general turnover-tax questions to a highly abbreviated follow-up about mixed business activities. The key question is whether retrieval can preserve the previously established VAT context when the current turn becomes underspecified.

Turn	Query focus	Gold	Role
T1	Business-tax standard	VAT Reg. Art. 2, 11	establish VAT domain
T2	VAT follow-up	VAT Reg. Art. 4	harder follow-up
T3	Tax rate	VAT Reg. Art. 2	easy anchor
T4	Mixed business activities	VAT Reg. Art. 3	key specialization

Table 11: Turn decomposition for the tax-rule specialization case.

T1: Establishing the VAT context

User query (CN): 您好，营业税是怎么收取的，他是以什么来做标准。
Gold: 《中华人民共和国增值税暂行条例》第二条；《中华人民共和国增值税暂行条例》第十一条

Raw dense retrieval

- **Retrieval:** Hits both gold statutes in the top-10, with Art. 11 at rank 3 and Art. 2 at rank 8.
- **Interpretation:** The raw dense retriever already enters the VAT domain, but the two gold anchors are relatively dispersed in the ranking.

HyDE retrieval

- **Retrieval:** Hits only one gold statute in the top-10, with Art. 2 at rank 1 and Art. 11 pushed down to rank 34.
- **Interpretation:** HyDE sharpens one tax anchor, but reduces overall gold coverage.

SAGA retrieval

- **Retrieval:** Hits both gold statutes in the top-5, with Art. 2 at rank 3 and Art. 11 at rank 4.

- **Interpretation:** SAGA preserves the VAT context while retrieving a more complete pair of statute-side anchors, covering both the tax-object/scope side and the tax-treatment side more tightly than the other systems.

T2: A harder follow-up about VAT calculation

User query (CN): 那增值税呢?
Gold: 《中华人民共和国增值税暂行条例》第四条

Raw dense retrieval

- **Retrieval:** Misses the gold in the top-10; Art. 4 appears at rank 14.

HyDE retrieval

- **Retrieval:** Also misses the gold in the top-10; Art. 4 falls to rank 36.

SAGA retrieval

- **Retrieval:** Still misses the gold in the top-10; Art. 4 appears at rank 24.
- **Interpretation:** This turn is not a strength of SAGA. All three systems remain inside the VAT domain, but none precisely resolves the follow-up to the tax-calculation rule in Art. 4.

T3: Easy tax-rate anchor

User query (CN): 那税率呢?
Gold: 《中华人民共和国增值税暂行条例》第二条

Raw dense retrieval / HyDE / SAGA

- **Retrieval:** All three systems hit the gold at rank 1.
- **Interpretation:** This is an easy turn, but it is useful analytically because it shows that the conversation remains centered on VAT. T4 should therefore be interpreted as a more specialized follow-up within the same tax-law context rather than as a new topic.

T4: Mixed-business specialization (core improvement)

User query (CN): 那我这经营的很多很杂，该怎么办?
Gold: 《中华人民共和国增值税暂行条例》第三条

Raw dense retrieval

- **Retrieval:** Misses the gold in the top-10; Art. 3 falls to rank 77.
- **Interpretation:** The raw dense baseline drifts away from the established VAT context and instead retrieves enterprise-registration, market-entity, and company-law articles triggered by the vague phrase "经营很多很杂".

HyDE retrieval

- **Retrieval:** Misses the gold in the top-10.
- **Interpretation:** Generic query expansion does not recover the intended tax-law specialization and remains dominated by company-law and registration-oriented evidence.

SAGA retrieval

- **Retrieval:** Hits the gold at rank 3, turning a raw-rank-77 / HyDE-miss case into a top-5 success.
- **Interpretation:** SAGA preserves the previously established VAT context and interprets the abbreviated follow-up as a mixed-business tax issue. As a result, it retrieves Art. 3, which is the correct statute-side anchor for the compatibility boundary under mixed or concurrent business activities.

Conclusion. This case illustrates both the strength and the remaining limits of SAGA in conversational statute retrieval. On the easier early turns, all systems can enter the correct tax-law domain, although SAGA provides more complete gold coverage in T1. The clearest benefit appears in T4: when the user shifts to the highly abbreviated query "经营的很多很杂", raw dense retrieval and HyDE both drift toward enterprise-registration and company-law evidence, whereas SAGA preserves the VAT context and promotes the gold statute from raw rank 77 / HyDE miss to rank 3. At the same time, T2 shows that SAGA does not solve all difficult follow-up turns, especially when the dialogue requires more precise recovery of a tax-calculation rule.

K Statistical Significance Tests

To verify that the main gains of SAGA are not due to random variation, we conduct paired sign-flip randomization tests over per-query scores on the common qid intersection of the compared runs. We report results against two strong baselines: the strongest raw dense baseline, Qwen3-Embedding-4B, and the LLM-enhanced retrieval baseline HyDE + Qwen3-Embedding-4B. Across datasets, the strongest evidence appears on LexRAG, where SAGA achieves statistically significant improvements over Qwen3-Embedding-4B on N@10, H@50, and R@50, and over HyDE + Qwen3-Embedding-4B on all four reported metrics. On STARD, the gains are smaller overall: the comparison against Qwen3-Embedding-4B is not statistically significant, while the improvement over HyDE + Qwen3-Embedding-4B remains significant on N@10. These results are consistent with the main-text observation that SAGA yields its clearest and most robust gains in the conversational LexRAG setting, while providing more modest improvements on STARD.

Dataset	Comparison	Metric	Delta	p-value	95% CI
LexRAG	SAGA vs. Qwen3-Embedding-4B	N@10	+0.0192	0.0170	[+0.0033, +0.0351]
		MRR	+0.0164	0.0731	[-0.0017, +0.0346]
		H@50	+0.0440	0.0003	[+0.0210, +0.0671]
		R@50	+0.0451	0.0002	[+0.0231, +0.0692]
LexRAG	SAGA vs. HyDE+Qwen3-Embedding-4B	N@10	+0.0523	<0.0001	[+0.0295, +0.0753]
		MRR	+0.0546	<0.0001	[+0.0302, +0.0798]
		H@50	+0.0419	0.0209	[+0.0084, +0.0755]
		R@50	+0.0472	0.0058	[+0.0140, +0.0814]
STARD	SAGA vs. Qwen3-Embedding-4B	N@10	+0.0079	0.2017	[-0.0040, +0.0200]
		MRR	+0.0153	0.0637	[-0.0006, +0.0316]
		H@50	+0.0065	0.6224	[-0.0065, +0.0195]
		R@50	+0.0089	0.2304	[-0.0043, +0.0233]
STARD	SAGA vs. HyDE+Qwen3-Embedding-4B	N@10	+0.0242	0.0470	[+0.0006, +0.0476]
		MRR	+0.0284	0.0863	[-0.0041, +0.0601]
		H@50	+0.0162	0.1244	[+0.0000, +0.0325]
		R@50	+0.0107	0.2619	[-0.0068, +0.0296]

Table 12: Paired sign-flip randomization tests over per-query scores on the common qid intersection of the compared runs. Delta denotes SAGA minus the comparison system. Confidence intervals are estimated with bootstrap resampling.

L Artifacts to be Released

To support reproducibility, we will release the following artifacts upon publication, subject to venue policy and licensing constraints:

- **Code** for the full SAGA pipeline, including preprocessing, CAF/AAF construction, dual-stream retrieval, fusion, and evaluation.
- **Prompts and parsing scripts** for CAF extraction, AAF extraction, query rewriting, summary generation, and HyDE generation.
- **Configuration files** specifying model checkpoints, retrieval settings, and fusion parameters.
- **Run scripts** for reproducing the main tables, ablations, and significance tests.
- **Intermediate outputs** for cached CAF/AAF structures where redistribution is allowed, together with regeneration scripts.

A detailed README will describe environment setup, data preparation, and an end-to-end reproduction checklist for Table 1.