

EnSI-RAG: Entity-Structure-Indexed Retrieval-Augmented Generation for Long-Document Question Answering

Xuanyu Meng, Jiashuo Sun, Jash Rajesh Parekh, Jiawei Han

University of Illinois Urbana-Champaign
Urbana, Illinois, USA
{xuanyum2, jiashuo5, jashrp2, hanj}@illinois.edu

Abstract

Question answering (QA) over long, connected documents remains challenging because relevant evidence may span multiple entities and their relationships. Existing retrieval-augmented generation (RAG) methods typically index documents as raw chunks and retrieve them through embedding similarity. Their performance degrades when chunk boundaries separate entities from supporting evidence or when a question requires multi-hop reasoning across the corpus. We propose *EnSI-RAG* (**Entity-Structure-Indexed Retrieval-Augmented Generation**), a framework that constructs a query-independent, entity-centered index. Each record $\langle e, t, k, v \rangle$ represents an entity e , its type t , a semantic category $k \in \{\text{property, relation, aspect}\}$, and a value v , while retaining links to the original source passages. At query time, these records serve as retrieval handles, and an LLM synthesizes the retrieved passages into the final answer. This design separates evidence localization from answer synthesis while preserving traceable source evidence. Across Loong and Oolong, EnSI-RAG achieves an average accuracy of 78.24. Relative to the published baseline scores used as references, this is 6.62 points higher, suggesting its effectiveness across these settings. The code is available at <https://github.com/RamonMeng/EnSI-RAG>.

Introduction

Answering questions over long documents remains a fundamental challenge for large language models (LLMs). In many real-world applications, the information required to answer a question is rarely contained within a single passage. Financial analysts may need to compare values across multiple reports, legal researchers may need to classify or aggregate evidence from many judgments, and scientists may need to trace citations, methods, or findings across collections of papers. These tasks require systems not only to locate relevant evidence, but also to preserve entity identity, document structure, and relationships among dispersed pieces of information.

Recent advances in LLMs have made long-document question answering increasingly feasible. One prominent direction is to extend the context window, allowing models to consume longer inputs directly (Anthropic 2024; Gemini 2024). However, larger context windows alone do not guarantee reliable evidence utilization: models remain sensitive to the position of relevant information, exhibiting the well-known

“lost in the middle” phenomenon (Liu et al. 2024). Moreover, recent long-context benchmarks demonstrate that even frontier models continue to struggle with questions requiring evidence aggregation and multi-hop reasoning across long documents rather than localized information lookup (Wang et al. 2024; Bertsch et al. 2025; Bai et al. 2025). Another direction is retrieval-augmented generation (RAG), which retrieves external evidence before generation (Lewis et al. 2020; Guu et al. 2020). While RAG substantially reduces the amount of text processed by the LLM, it shifts the burden to the retriever, since answer quality is ultimately limited by the evidence it retrieves.

Most existing RAG systems treat documents as collections of fixed-length chunks and retrieve them using lexical or dense similarity. Although simple and scalable, fixed-size chunking is often poorly aligned with the semantic organization of long documents. Chunk boundaries may separate entities from their supporting evidence, detach table rows from their headers, or merge multiple unrelated topics into the same retrieval unit. Consequently, retrieval may return passages that are locally similar to the query but insufficient for answering it, while overlooking semantically essential evidence that shares little lexical similarity with the question.

Our key observation is that retrieval units should be defined semantically rather than mechanically. Instead of indexing arbitrary chunks, EnSI-RAG represents documents as *entity-centered passages*, where each passage focuses primarily on one entity together with the local context describing that entity. Such passages preserve semantically coherent evidence while reducing both semantic fragmentation and contamination, providing a more reliable foundation for retrieval. However, semantically coherent passages alone are insufficient. To retrieve them efficiently for arbitrary future questions, they must also be organized into a query-independent representation that exposes their underlying semantic structure.

Based on this observation, we propose **EnSI-RAG** (Entity-Structure-Indexed Retrieval-Augmented Generation), a query-independent framework for long-document question answering. EnSI-RAG first preprocesses a document collection by constructing entity-centered passages and extracting structured records describing the entities, their semantic types, and their associated properties, relations, and aspects. These records are organized into an entity-structure index that serves as a set of retrieval handles pointing back to the origi-

nal passages. At query time, the index guides retrieval toward the supporting passages, while the LLM remains responsible for integrating the retrieved evidence and synthesizing the final answer. Unlike fully structured approaches that perform reasoning over normalized databases, EnSI-RAG uses structure only to improve evidence localization; reasoning remains grounded in the original document text.

We evaluate EnSI-RAG on two complementary long-document question-answering benchmarks: Loong (Wang et al. 2024) and Oolong (Bertsch et al. 2025). Loong evaluates heterogeneous multi-document reasoning across financial, legal, and academic domains, while Oolong emphasizes large-scale information aggregation over long inputs. Across these different document structures and reasoning requirements, EnSI-RAG applies a unified retrieval framework while allowing domain-specific passage construction. Experimental results demonstrate that this design achieves competitive or superior performance compared with strong retrieval baselines. Our contributions are threefold. First, we introduce an entity-centered passage representation that defines retrieval units according to semantic coherence rather than fixed-length segmentation. Second, we propose a query-independent entity-structure index that enables passages to be retrieved through entities and their associated semantic information while preserving the original documents as evidence. Third, we develop a unified retrieval framework that combines structured evidence localization with LLM-based semantic synthesis, providing an effective middle ground between conventional chunk-based RAG and fully structured question answering systems.

Related Work

Offline corpus structuring. Building query-independent corpus representations of corpora long predates LLMs. Classical information retrieval constructs inverted indexes and term-weighted vectors offline (Salton and Buckley 1988; Robertson et al. 1994), while web-scale information extraction extends indexing to entities and relations (Etzioni et al. 2004, 2008; Dong et al. 2014). Recent methods use LLMs to construct hierarchical or recursive summaries (Sarathi et al. 2024; Chen et al. 2023), segment text into propositions or entity-centered spans (Chen et al. 2024; Jiang, Ma, and Chen 2024), or convert documents into tables, schemas, and knowledge graphs (Li et al. 2024; Liang et al. 2026; Wu, Zhang, and Li 2022). Such representations provide compact and uniform access to a corpus, but information discarded during structuring is unavailable at query time, and schemas designed for one question class may transfer poorly to others. EnSI-RAG instead uses structured records only as retrieval handles: they locate the original passages, which retain the evidence used for answer generation.

Retrieval-augmented generation. Larger LLM context windows do not fully resolve long-document QA as context size is not the binding constraint (Du et al. 2025). Models exhibit a lost in the middle effect, retrieving evidence unevenly across input positions (Liu et al. 2024), leading to struggles with aggregation-intensive questions (Wang et al. 2024; Bertsch et al. 2025; Bai et al. 2025; Yen et al. 2025). RAG grounds

generation in retrieved evidence (Lewis et al. 2020; Guu et al. 2020; Izacard et al. 2023), with adaptive, iterative, and RL-trained retrieval policies improving this process (Asai et al. 2024; Trivedi et al. 2023; Jiang et al. 2023; Jin et al. 2025). However, these methods still rely on near-perfect retriever recall. The dominant approach, ranking fixed-size chunks by lexical or dense similarity (Karpukhin et al. 2020; Izacard et al. 2022; Khattab and Zaharia 2020), is poorly suited to long documents. For example, chunk boundaries may sever coherent discussions, detach table rows from their headers, or merge unrelated entities into a single retrieval unit, yielding context that is locally similar yet incomplete or omitting evidence with little lexical overlap (Gong et al. 2020; Wang et al. 2025; Shin et al. 2025). Graph-based RAG addresses part of this issue by building entity and relation structures (Edge et al. 2024; Jiménez Gutiérrez et al. 2024; Parekh, Jiang, and Han 2025; Guo et al. 2024), but is primarily developed and evaluated on short-passage benchmarks with well-canonicalized entities. These assumptions often break down in long, heterogeneous, and domain-specific documents.

EnSI-RAG

Figure 1 presents the overall workflow of EnSI-RAG. The framework consists of an offline preprocessing stage followed by an online question-answering stage. During offline preprocessing, EnSI-RAG performs **Passage Construction** to convert long documents into semantically contained, entity-centered passages, followed by **Information Extraction** to derive structured records describing the entities appearing in each passage together with their semantic types, properties, relations, and aspects. These records are then transformed during **Index Entry Building** into a query-independent entity-structure index, while the original passages are preserved in a passage store. During online question answering, the **Retrieval** stage first analyzes the user query to derive structured retrieval requirements and then uses the entity-structure index to locate the corresponding supporting passages. Finally, the retrieved original passages are provided to the **Generation** stage, where an LLM synthesizes the final answer directly from grounded source evidence. This separation between offline indexing and online reasoning allows EnSI-RAG to reuse the same indexed corpus for arbitrary future queries while preserving the original passages as the basis for answer generation. The following subsections describe these stages in detail.

Passage Construction

Given a collection of long documents $\mathcal{D} = \{d_1, \dots, d_n\}$, EnSI-RAG converts each document into semantically self-contained passages. Unlike a fixed-size chunk, which is determined by a token budget, character length, or sliding window, a passage is organized around a main entity and retains sufficient local context to identify its type and extract its properties, relations, and aspects. Formally,

$$\begin{aligned} \mathcal{P}_d &= \{p_1, \dots, p_m\}, \\ \mathcal{P} &= \bigcup_{d \in \mathcal{D}} \mathcal{P}_d. \end{aligned} \tag{1}$$

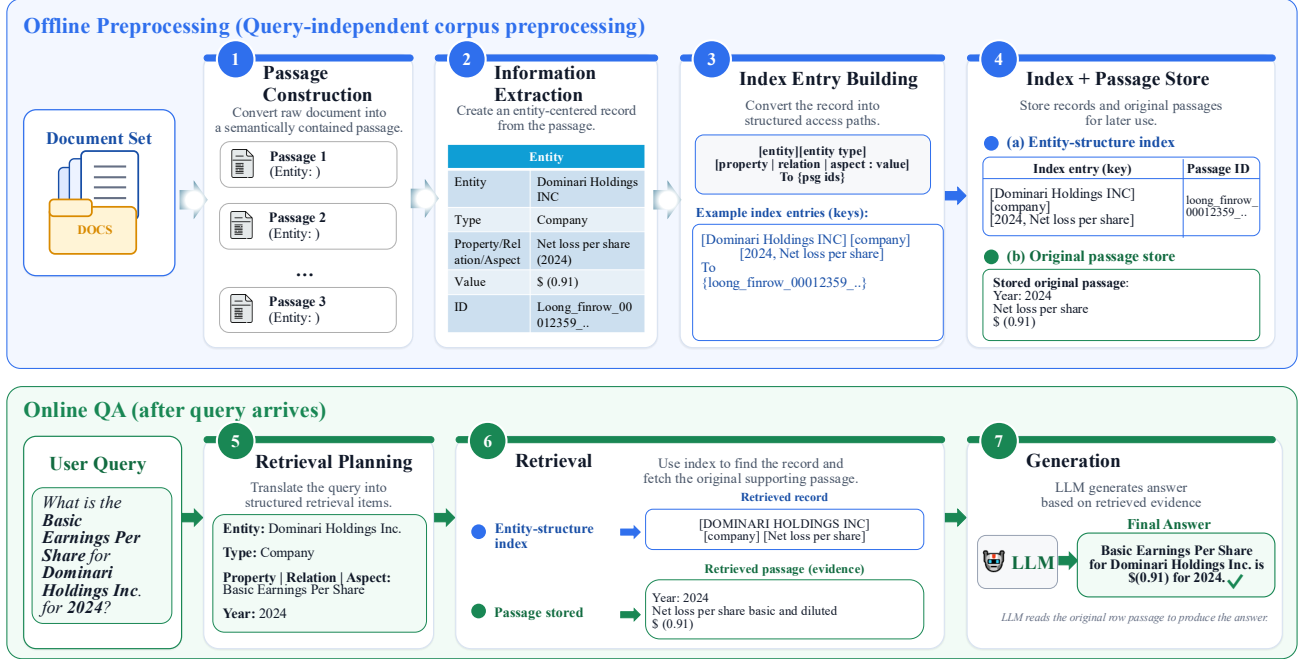


Figure 1: End-to-end EnSI-RAG workflow on a financial QA example. The upper pipeline shows query-independent offline preprocessing: a Dominari Holdings filing is converted into a self-contained financial row passage, an entity-centered record, and entity-structure index entries. The lower pipeline shows online QA: after the query arrives, EnSI-RAG plans retrieval, retrieves the matched record and original supporting passage, and uses an LLM to generate the final answer from the passage evidence

Each passage has a stable identifier psg_id and preserves its source document, available source offsets, and lightweight structural context, such as titles, headers, reporting periods, or local citations.

Passage construction seeks semantic containment: a passage should be able to keep an entity with its supporting evidence, yet focused enough to avoid mixing unrelated entities or facts. Rather than imposing a globally fixed passage schema, EnSI-RAG instantiates passage design on the fly during corpus preprocessing according to the requirements of the task and the structural organization of the documents. Passage boundaries may therefore vary across tasks, domains, and document types, while remaining independent of any individual query. Regardless of the constructor used, every passage conforms to the same downstream interface—a semantically contained text unit centered on one entity and linked to its source by a unique psg_id .

This objective balances evidence coverage for the main entity against the inclusion of unrelated entities or topics. These are semantic criteria rather than hard length constraints, so passage sizes may vary with the context needed to make the evidence self-contained.

Compared with arbitrary chunking, this design reduces semantic fragmentation because an entity and its evidence remain together whenever possible. It also reduces semantic

contamination because unrelated facts are less likely to share a retrieval unit. The resulting context makes the extraction of entity types, properties, relations, and aspects more reliable. Most importantly, each passage preserves the original text and its provenance, allowing later stages to retrieve grounded evidence rather than an extraction-only surrogate and to defer final semantic integration to the LLM.

Information Extraction

EnSI-RAG next extracts entity-centered information from each passage. Rather than answering a particular question or normalizing the corpus into a rigid database, this stage creates reusable retrieval handles through which passages can be addressed by their entity structure.

For each $p \in \mathcal{P}$, the extractor produces

$$\mathcal{R}_p = \{r_1, \dots, r_k\}, \quad r = \langle e, t, \mathcal{M}, psg_id \rangle, \quad (2)$$

where e is an entity mention or canonical name, t is its type, and $\mathcal{M}$ contains its passage-supported semantic fields. Each field is

$$m = \langle c, k, v \rangle, \quad c \in \{property, relation, aspect\}, \quad (3)$$

where k is the field name and v its value. A property records an attribute value, a relation records a target entity, and an

aspect indicates the kind of information expressed about the entity. For example, $\text{birth-loc} = \text{Memphis}$ is a property of a singer, whereas $\text{cites} = \text{paper } B$ is a relation of *paper A*. Birthplace and financial metrics are properties, citations and affiliations are relations, and method or legal reasoning are aspects.

These categories describe distinct retrieval cues. Properties connect an entity to an attribute value, relations connect it to another entity, and aspects expose passages that discuss a particular facet even when no single normalized value is appropriate. A passage may yield multiple records and each record may contain multiple supported fields, while the common psg_id preserves their connection to the same source evidence.

Extraction follows two principles. First, every indexed field must be supported by local passage evidence. Second, extraction is retrieval-oriented rather than answer-oriented: a record need not encode every detail required by future questions, but should provide reliable access to the original evidence. This differs from fully structured systems, in which an extraction error directly enters the reasoning state. EnSI-RAG uses records primarily as passage-addressing keys and generates answers from the retrieved source passages; consequently, even a partial record remains useful when it leads to the correct evidence.

This separation is essential: extracted values identify where information is located but are not treated as its complete representation. Details unnecessary for indexing remain in the passage and are available during generation. Conversely, unsupported information is excluded even if it appears plausible from background knowledge, keeping every access path grounded in the corpus.

Index Entry Building

The extracted records form an entity-structure index $\mathcal{I}$ that maps each tuple (e, t, c, k, v) to the identifiers of passages whose records contain the corresponding entity, type, and semantic field. An entry is written compactly as

$$[e][t][c : k = v] \rightarrow \{\text{psg_id}\}. \quad (4)$$

Every field in a record contributes its psg_id to the corresponding set. This rule covers all three field categories and naturally aggregates evidence across passages and documents. For instance,

$$\begin{aligned} &[\text{Franklin}][\text{Singer/Person}] \\ &[\text{property} : \text{birth-loc} = \text{Memphis}] \\ &\rightarrow \{\text{psg_id}\}. \end{aligned}$$

The output is deliberately a set rather than a single identifier. The same entity-structure key may be supported by passages in which a fact is repeated, refined, contradicted, or presented from different perspectives. Retaining all such passages avoids prematurely collapsing evidence into one canonical passage or normalized value.

Relations that are useful in both directions may also receive inverse entries. For example, a *paper A cites paper B* entry may be accompanied by an entry from *paper B* to *paper A* under *cited-by*. The same principle applies to authorship,

aliases, location containment, and organizational affiliation, enabling retrieval from either side of a relation.

Set-valued indexing supports different reasoning needs. A narrow lookup can select the most relevant passage; aggregation or comparison can combine several passages; and ambiguous or multi-hop questions can explore alternative evidence paths. The index is therefore a passage-addressing layer rather than a lossy database of final answers.

The retained alternatives matter because repeated evidence is not necessarily redundant. Separate passages may supply complementary context, use different terminology, qualify a value, or reveal a disagreement that should be resolved during generation. Keeping these passages available lets retrieval select evidence according to the current question rather than fixing a single preferred context during indexing.

Each entry retains both symbolic fields and a textual representation. The former enables constrained matching over entities, types, categories, field names, and values, while the latter supports semantic matching across different surface forms. Together they provide precise yet flexible access to the source corpus.

Retrieval

Given a question q , EnSI-RAG derives a retrieval plan

$$H(q) = \{h_1, \dots, h_m\}, \quad (5)$$

where each hop h_i specifies a retrieval requirement over some combination of entity, entity type, property, relation, aspect, value, or target entity. Direct questions may require only one hop; in multi-hop questions, later requirements may depend on entities or values discovered earlier.

Accordingly, $H(q)$ is a plan rather than a requirement to decompose every question. When one structured requirement is sufficient, EnSI-RAG performs a single retrieval step. When dependencies exist, the result of hop h_i supplies an intermediate entity or value used to instantiate h_{i+1} . This preserves the original question’s reasoning chain while expressing each retrieval step in the same entity-centered vocabulary as the index.

Each hop is matched against an index key $x = (e, t, c, k, v)$. Their compatibility is modeled as

$$\text{score}(h, x) = \sum_{a \in \{e, t, c, k, v\}} \lambda_a s_a(h, x), \quad (6)$$

where a ranges over entity, entity type, field category, field name, and value or target; s_a measures compatibility on field a ; and λ_a controls its contribution. This formulation accommodates both symbolic and semantic matching without tying the framework to a particular implementation.

The component form makes partial requirements possible. A hop may strongly constrain an entity and relation while leaving the target unknown, or it may seek an aspect for an entity of a particular type. Only fields expressed or implied by the retrieval requirement need to contribute to its match. The ranked keys therefore act as structured hypotheses about which index entries provide the required evidence.

Because every key maps to a passage set, the candidates for a hop and the accumulated evidence for the question are

$$\begin{aligned} C_h &= \bigcup_{x \in \text{TopKeys}(h)} \mathcal{I}(x), \\ C_q &= \bigcup_{h \in H(q)} C_h. \end{aligned} \quad (7)$$

Duplicate identifiers are removed, after which the passages may be reranked before generation.

Key ranking and passage ranking serve different roles. The former identifies compatible entity structures; the latter selects the most useful source contexts among all passages reached through those structures. Their separation is particularly useful when a highly compatible key has several supporting passages of different relevance to the question.

Unlike ordinary RAG, EnSI-RAG does not rely solely on matching the full question against anonymous raw chunks. It uses structured, entity-centered keys to constrain retrieval while returning the original evidence passages. In multi-hop reasoning, one hop can retrieve a relation and expose an intermediate entity that becomes the query entity of the next hop. Retrieval thus follows the corpus’s entity structure rather than a single embedding search.

The set-valued index preserves alternative supporting contexts within this structured search space. It can supply multiple passages under one key, compare evidence associated with different keys, and maintain competing paths for ambiguous questions. EnSI-RAG therefore indexes structured access paths while leaving evidence integration and final answer generation to the LLM.

Generation

After retrieval, EnSI-RAG generates the final answer directly from the retrieved original passages. Given a user question q and the retrieved passage set C_q , the generation stage simply provides them to an LLM:

$$\hat{y} = \text{LLM}(q, C_q). \quad (8)$$

Unlike the preceding stages, this step performs no additional indexing or symbolic reasoning. The LLM is responsible only for synthesizing the final answer from the retrieved evidence. Because retrieval has already localized the relevant supporting passages, the generation model can focus on semantic understanding and evidence integration rather than searching through the document collection.

This design cleanly separates evidence localization from answer synthesis. EnSI-RAG uses the entity-structure index to retrieve the original supporting passages, while the LLM operates directly on those passages to produce the final answer. Consequently, the retrieved passages remain the source of evidence, and the structured records are used only as retrieval handles rather than as a reasoning substrate.

Experiments

Experimental Setup

We evaluate EnSI-RAG on two open-source long-document question-answering benchmarks: Loong (Wang et al. 2024)

Benchmark	Task Type	# Questions
Loong	Retrieval, Aggregation	177
Oolong	Classification, Aggregation	192

Table 1: Benchmarks used in our experiments. The number of questions corresponds to the evaluated subset of each benchmark.

and Oolong (Bertsch et al. 2025). Loong emphasizes heterogeneous multi-document reasoning across financial, legal, and academic domains, whereas Oolong focuses on large-scale information aggregation over long inputs. Together, they test whether EnSI-RAG can localize and integrate evidence across different document structures and reasoning requirements. The benchmark details are shown in Table 1.

We follow the official evaluation protocol of each benchmark. Loong is evaluated using average accuracy. For Oolong, non-numeric questions use an LLM-based judge, whereas numeric aggregation questions use the official deviation-based metric.

We compare EnSI-RAG with RAG (Lewis et al. 2020), LongRAG (Jiang, Ma, and Chen 2024), GraphRAG (Edge et al. 2024), BaseModel, DocETL (Shankar et al. 2024), Chain-of-Agent (Zhang et al. 2024), RLM (Zhang, Kraska, and Khattab 2025), and SLIDERS (Joshi et al. 2026). All baseline scores on Loong and Oolong are taken from SLIDERS (Joshi et al. 2026). EnSI-RAG is evaluated on our sampled Loong subset and on the Oolong subset obtained by filtering for the required document length and selecting the first 192 eligible questions in dataset order, following the SLIDERS configuration. Because our Loong evaluation subset may differ from that used for the published baselines, its baseline scores are included as reference rather than as controlled head-to-head comparisons.

Main Results

Table 2 compares EnSI-RAG with the baseline results reported by SLIDERS (Joshi et al. 2026). The Oolong results use the same deterministic benchmark slice, whereas the Loong results may use different sampled subsets and are therefore included as reference rather than as a controlled head-to-head comparison. Each EnSI-RAG result is obtained from one complete run.

Ablation Study

We study three design choices in EnSI-RAG: passage granularity, property/aspect granularity, and retrieval depth. For passage granularity and retrieval depth, we report results on Financial questions from Loong. We evaluate property/aspect granularity on Loong questions from the Financial, Paper, and Legal domains. Within each comparison, we use the same questions and vary only the factor under study. Following the evaluation protocol of Loong (Wang et al. 2024), an evaluator assesses each answer from 1 to 100 according to accuracy, hallucination, and completeness. We report average accuracy (Acc.).

Models	LLMs	Oolong	Loong	Avg.
RAG	Qwen3-4B + GPT-4.1	11.32	54.35	32.84
LongRAG	Qwen3-4B + GPT-4.1	22.00	59.10	40.55
GraphRAG	Qwen3-4B + GPT-4.1	22.00	61.28	41.64
BaseModel	GPT-4.1	45.56	76.74	61.15
BaseModel	Qwen3.5-122B-A10B	24.89	74.78	49.84
DocETL	GPT-4.1	49.00	75.03	62.02
Chain-of-Agent	GPT-5 + GPT-5-mini	17.11	54.46	35.79
RLM	GPT-5 + GPT-5-mini	51.42	72.64	62.03
SLIDERS	GPT-4.1 + GPT-4.1-mini	64.67	78.57	71.62
EnSI-RAG	GPT-4.1 + GPT-4.1-mini	71.84	84.64	78.24

Table 2: Accuracy on the two benchmarks. All baseline scores are taken from SLIDERS (Joshi et al. 2026), while EnSI-RAG results are obtained from our experiments. Higher is better.

Pipeline Stage	Input Tokens	Latency (s)	LLM Calls
<i>Offline preprocessing</i>			
Passage Construction and Information Extraction	157,031	468.51	32.35
Index Entry Building	0	0.05	0.00
<i>Online question answering</i>			
Retrieval	656	19.01	1.00
Generation	5,460	6.73	1.00
Overall	163,146	494.30	34.35

Table 3: Average efficiency per query under the default EnSI-RAG configuration. Passage Construction, Information Extraction, and Index Entry Building are offline stages whose costs are amortized over the evaluated queries. Retrieval and Generation are online stages. In the current implementation, LLM-based information extraction is integrated into Passage Construction.

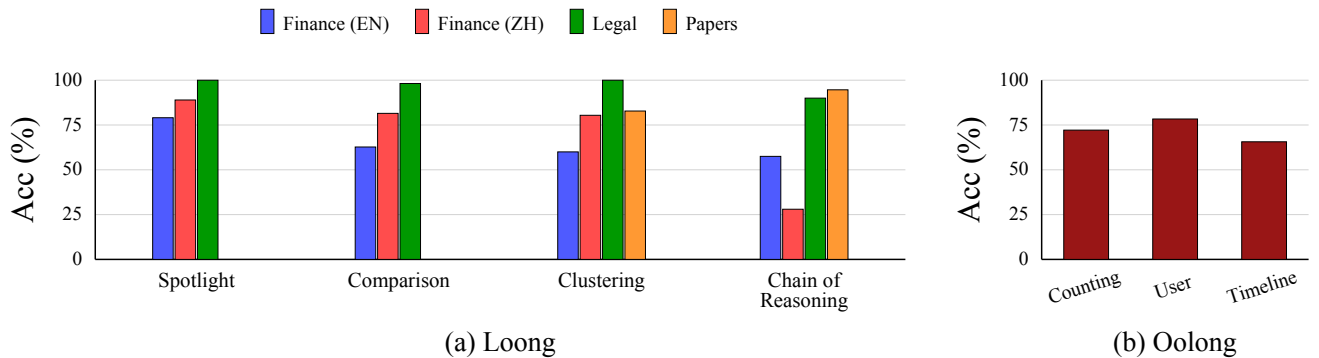


Figure 2: Accuracy breakdown by task type on Loong and Oolong. Missing bars denote absent domain-task combinations. Higher is better.

Passage granularity. Representing each financial table row as an individual passage substantially outperforms representing the entire table as one passage. Row-level passages achieve 100.00, compared with 91.50 for table-level passages. The 8.50-point gap indicates that compact evidence units reduce irrelevant context within a retrieved passage and provide a more precise interface between retrieval and generation. We therefore select row-level passages for Financial documents.

Property, relation, and aspect granularity. We compare labels extracted verbatim from the source with labels mapped to higher-level concepts. This transformation applies to property and aspect names; directional relation labels are preserved to avoid changing graph semantics. Coarse labels improve English Financial from 92.00 to 93.50 and Paper from 88.00 to 91.50, suggesting that semantic abstraction reduces lexical sparsity in these domains. In contrast, Legal decreases from 71.25 to 68.75. Legal queries often depend on

Passage granularity	Accuracy (%) $\uparrow$
Row-level	100.00
Table-level	91.50

Table 4: Passage-granularity ablation on English Financial questions from Loong. Higher is better.

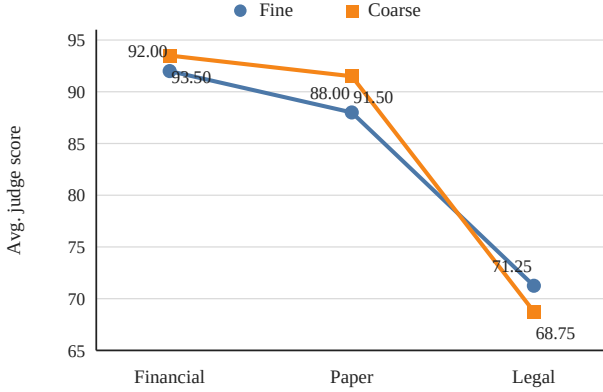


Figure 3: Accuracy under different property, relation, and aspect granularities across Loong domains. Coarse labels perform better on Financial and Paper, whereas fine-grained labels perform better on Legal. Higher accuracy is better.

fine distinctions among case attributes, and collapsing these distinctions can make retrieved evidence less discriminative. Thus, the optimal granularity is domain dependent: coarse labels are selected for English Financial and Paper, whereas fine labels are retained for Legal.

Retrieval depth. Top-5 achieves the highest English Financial accuracy of 94.00, outperforming Top-15 at 92.50 and Top-12 at 92.00. Retrieving more evidence therefore provides no benefit on this subset, suggesting that concentrated evidence reduces distraction. Nevertheless, all main results use Top-12 to maintain a common configuration and avoid benchmark-specific tuning.

Discussion. Overall, the ablations favor row-level passages and domain-dependent property granularity, while showing that shallower retrieval may benefit the evaluated English Financial subset.

Efficiency

We evaluate the average per-query efficiency of each EnSI-RAG stage using LLM input tokens, wall-clock latency, and the number of LLM calls. Passage Construction, Information Extraction, and Index Entry Building are query-independent offline stages, so their costs are amortized over the evaluated queries. In the current implementation, passage construction and LLM-based information extraction are performed jointly.

As shown in Table 3, Passage Construction and Information Extraction account for 157,031 input tokens, 468.51 seconds, and 32.35 LLM calls per query after amortization. Index Entry Building adds no input tokens or LLM calls and

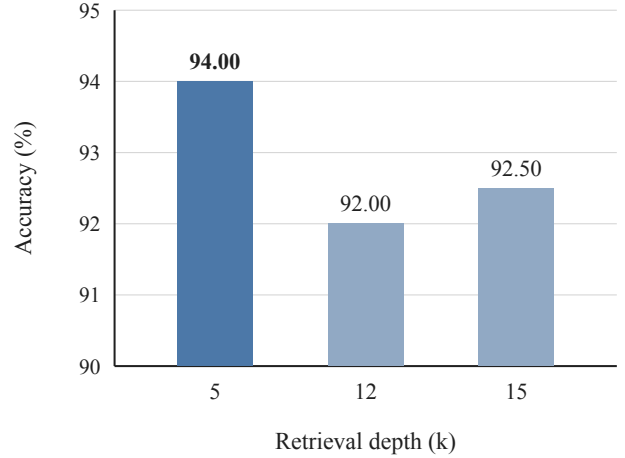


Figure 4: Retrieval-depth ablation on English Financial questions from Loong. Top-5 achieves the highest accuracy; retrieving additional passages provides no improvement in this setting. Higher accuracy is better.

requires 0.05 seconds. The online stages are substantially lighter: Retrieval uses 656 input tokens, 19.01 seconds, and 1.00 LLM call, while Generation uses 5,460 input tokens, 6.73 seconds, and 1.00 LLM call. Overall, EnSI-RAG uses 163,146 input tokens, 494.30 seconds, and 34.35 LLM calls per query. This profile reflects the system’s central design: expensive corpus processing is performed offline and reused, while online inference is limited to structured retrieval and answer generation over a small set of original passages.

The ablation experiments are run locally using Qwen3-32B on four NVIDIA RTX A6000 GPUs. The main EnSI-RAG experiments use GPT-4.1-mini and GPT-4.1 through the OpenAI API, for which provider-side hardware details are unavailable.

Conclusion

We presented EnSI-RAG, an entity-structure-indexed retrieval-augmented generation framework for long-document question answering. Instead of indexing documents as arbitrary fixed-size chunks, EnSI-RAG constructs semantically contained passages and builds a query-independent index of the form $[entity][entity_type][property \mid relation \mid aspect : value] \rightarrow \{psg_id\}$. This index provides structured access paths from future questions to sets of original supporting passages, improving evidence localization while preserving the LLM’s role as the final semantic integrator. Across financial, legal, academic, and aggregation-heavy QA settings, EnSI-RAG is designed to provide a flexible middle ground between unstructured chunk-based RAG and fully structured database-style reasoning. More broadly, our results suggest that long-document QA benefits from separating evidence localization from answer synthesis: the index guides the system toward relevant passages, and the LLM integrates those passages into the final answer.

References

- Anthropic. 2024. The Claude 3.5 Sonnet Model.
- Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; and Hajishirzi, H. 2024. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *International Conference on Learning Representations*.
- Bai, Y.; Tu, S.; Zhang, J.; Peng, H.; Wang, X.; Lv, X.; Cao, S.; Xu, J.; Hou, L.; Dong, Y.; Tang, J.; and Li, J. 2025. Long-Bench v2: Towards Deeper Understanding and Reasoning on Realistic Long-Context Multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Bertsch, A.; Pratapa, A.; Mitamura, T.; Neubig, G.; and Gormley, M. R. 2025. Oolong: Evaluating Long Context Reasoning and Aggregation Capabilities. *arXiv preprint arXiv:2511.02817*.
- Chen, H.; Pasunuru, R.; Weston, J.; and Celikyilmaz, A. 2023. Walking Down the Memory Maze: Beyond Context Limit through Interactive Reading. *arXiv preprint arXiv:2310.05029*.
- Chen, T.; Wang, H.; Chen, S.; Yu, W.; Ma, K.; Zhao, X.; Zhang, H.; and Yu, D. 2024. Dense X Retrieval: What Retrieval Granularity Should We Use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 15159–15177.
- Dong, X. L.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmman, T.; Sun, S.; and Zhang, W. 2014. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 601–610.
- Du, Y.; Tian, M.; Ronanki, S.; Rongali, S.; Bodapati, S.; Galstyan, A.; Wells, A.; Schwartz, R.; Huerta, E. A.; and Peng, H. 2025. Context length alone hurts LLM performance despite perfect retrieval. *arXiv preprint arXiv:2510.05381*.
- Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; and Larson, J. 2024. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv preprint arXiv:2404.16130*.
- Etzioni, O.; Banko, M.; Soderland, S.; and Weld, D. S. 2008. Open Information Extraction from the Web. *Communications of the ACM*, 51(12): 68–74.
- Etzioni, O.; Cafarella, M.; Downey, D.; Kok, S.; Popescu, A.-M.; Shaked, T.; Soderland, S.; Weld, D. S.; and Yates, A. 2004. Web-Scale Information Extraction in KnowItAll. In *Proceedings of the 13th International Conference on World Wide Web*, 100–110.
- Gemini. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Gong, H.; Shen, Y.; Yu, D.; Chen, J.; and Yu, D. 2020. Recurrent chunking mechanisms for long-text machine reading comprehension. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, 6751–6761.
- Guo, Z.; Xia, L.; Yu, Y.; Ao, T.; and Huang, C. 2024. LightRAG: Simple and Fast Retrieval-Augmented Generation. *arXiv preprint arXiv:2410.05779*.
- Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; and Chang, M.-W. 2020. REALM: Retrieval-Augmented Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 3929–3938.
- Izacard, G.; Caron, M.; Hosseini, L.; Riedel, S.; Bojanowski, P.; Joulin, A.; and Grave, E. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research*.
- Izacard, G.; Lewis, P.; Lomeli, M.; Hosseini, L.; Petroni, F.; Schick, T.; Dwivedi-Yu, J.; Joulin, A.; Riedel, S.; and Grave, E. 2023. Atlas: Few-Shot Learning with Retrieval-Augmented Language Models. *Journal of Machine Learning Research*, 24(251): 1–43.
- Jiang, Z.; Ma, X.; and Chen, W. 2024. LongRAG: Enhancing Retrieval-Augmented Generation with Long-Context LLMs. *arXiv preprint arXiv:2406.15319*.
- Jiang, Z.; Xu, F. F.; Gao, L.; Sun, Z.; Liu, Q.; Dwivedi-Yu, J.; Yang, Y.; Callan, J.; and Neubig, G. 2023. Active Retrieval Augmented Generation. *arXiv preprint arXiv:2305.06983*.
- Jiménez Gutiérrez, B.; Shu, Y.; Gu, Y.; Yasunaga, M.; and Su, Y. 2024. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. *arXiv preprint arXiv:2405.14831*.
- Jin, B.; Zeng, H.; Yue, Z.; Wang, D.; Zamani, H.; and Han, J. 2025. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. *arXiv preprint arXiv:2503.09516*.
- Joshi, H.; Shethia, P.; Dao, J.; and Lam, M. S. 2026. Contexts Are Never Long Enough: Structured Reasoning for Scalable Question Answering over Long Document Sets. *arXiv preprint arXiv:2604.22294*.
- Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 6769–6781.
- Khattab, O.; and Zaharia, M. 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*, 39–48.
- Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, 9459–9474.
- Li, Z.; Chen, X.; Yu, H.; Lin, H.; Lu, Y.; Tang, Q.; Huang, F.; Han, X.; Sun, L.; and Li, Y. 2024. StructRAG: Boosting Knowledge-Intensive Reasoning of LLMs via Inference-Time Hybrid Information Structurization. *arXiv preprint arXiv:2410.08815*.
- Liang, Z.; Lin, X.; Zhang, Z.; Luo, Y.; Wang, H.; and Tang, N. 2026. Long-Document QA with Chain-of-Structured-Thought and Fine-Tuned SLMs. *arXiv preprint arXiv:2603.29232*.

- Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12: 157–173.
- Parekh, J. R.; Jiang, P.; and Han, J. 2025. Structure-Augmented Reasoning Generation. *arXiv preprint arXiv:2506.08364*.
- Robertson, S. E.; Walker, S.; Jones, S.; Hancock-Beaulieu, M. M.; and Gatford, M. 1994. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*, volume 500-225 of *NIST Special Publication*, 109–126. National Institute of Standards and Technology.
- Salton, G.; and Buckley, C. 1988. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24(5): 513–523.
- Sarathi, P.; Abdullah, S.; Tuli, A.; Khanna, S.; Goldie, A.; and Manning, C. D. 2024. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval. In *International Conference on Learning Representations*.
- Shankar, S.; Chambers, T.; Shah, T.; Parameswaran, A. G.; and Wu, E. 2024. DocETL: Agentic Query Rewriting and Evaluation for Complex Document Processing. *arXiv preprint arXiv:2410.12189*.
- Shin, J.; Park, C.; Park, J.; Seo, J.; and Lim, H.-S. 2025. MultiDocFusion: Hierarchical and Multimodal Chunking Pipeline for Enhanced RAG on Long Industrial Documents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 20996–21015.
- Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 2023. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 10014–10037.
- Wang, M.; Chen, L.; Fu, C.; Liao, S.; Zhang, X.; Wu, B.; Yu, H.; Xu, N.; Zhang, L.; Luo, R.; Li, Y.; Yang, M.; Huang, F.; and Li, Y. 2024. Leave No Document Behind: Benchmarking Long-Context LLMs with Extended Multi-Document QA. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.
- Wang, Z.; Gao, C.; Xiao, C.; Huang, Y.; Si, S.; Luo, K.; Bai, Y.; Li, W.; Duan, T.; Lv, C.; et al. 2025. Document segmentation matters for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, 8063–8075.
- Wu, X.; Zhang, J.; and Li, H. 2022. Text-to-Table: A New Way of Information Extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2518–2533.
- Yen, H.; Gao, T.; Hou, M.; Ding, K.; Fleischer, D.; Izsak, P.; Wasserblat, M.; and Chen, D. 2025. HELMET: How to Evaluate Long-Context Models Effectively and Thoroughly. In *International Conference on Learning Representations*.
- Zhang, A. L.; Kraska, T.; and Khattab, O. 2025. Recursive Language Models. *arXiv preprint arXiv:2512.24601*.
- Zhang, Y.; Sun, R.; Chen, Y.; Pfister, T.; Zhang, R.; and Arik, S. Ö. 2024. Chain of Agents: Large Language Models Collaborating on Long-Context Tasks. In *Advances in Neural Information Processing Systems*.

Appendix: Experimental Details

Benchmark	Primary Focus	Description
Loong (Wang et al. 2024)	Multi-document question answering	A realistic long-context, multi-document question-answering benchmark spanning academic papers, legal documents, and financial reports. It includes four reasoning categories: Spotlight Locating, Comparison, Clustering, and Chain of Reasoning. These tasks require models to locate and integrate evidence across multiple related documents rather than retrieve an answer from a single context.
Oolong (Bertsch et al. 2025)	Long-context aggregation	Evaluates long-context reasoning and information aggregation. Relevant information is distributed throughout long inputs, requiring a model to identify pertinent segments, reason locally over them, and aggregate the intermediate results into a final answer. The benchmark contains synthetic aggregation tasks (OOLONG-SYNTH) and real-world aggregation tasks (OOLONG-REAL). All Oolong results reported in the main paper are evaluated exclusively on Oolong-Synth; we do not evaluate on Oolong-Real.

Table A1: Detailed descriptions of the benchmarks used in our experiments. Evaluated subset sizes and task types are reported in the main paper.

Benchmark	Evaluator or Metric	Protocol
Loong	Official benchmark evaluation framework	We use the official Loong evaluation framework, which is also adopted by SLIDERS, and report the average accuracy across the evaluated questions.
Oolong	LLM-as-a-judge and official deviation-based metric	We follow the official Oolong protocol. Non-numeric questions are evaluated using an LLM-as-a-judge, whereas numeric aggregation questions use Oolong’s official deviation-based metric, which assigns higher scores to predictions with smaller numerical errors.

Table A2: Evaluation protocols used for the two benchmarks.
We follow each benchmark’s official protocol.

Baseline	Category	Description
RAG (Lewis et al. 2020)	Conventional retrieval	Retrieves semantically similar fixed-length document chunks and provides them to an LLM for answer generation. It represents the standard retrieval-based baseline. Because its retrieval units are determined by fixed chunk boundaries, related evidence may be separated and unrelated topics may be placed in the same chunk.
LongRAG (Jiang, Ma, and Chen 2024)	Larger-granularity retrieval	Extends conventional RAG by retrieving substantially larger textual units. The increased retrieval granularity can reduce semantic fragmentation and preserve more context, but may also introduce more irrelevant information into the generation context.
GraphRAG (Edge et al. 2024)	Graph-guided retrieval	Constructs a graph representation of the document collection in which entities and their relations form the retrieval structure. It performs graph-guided retrieval over connected entities before generating the final answer.
BaseModel	Direct long-context inference	Directly performs long-context inference without external retrieval or preprocessing. The entire available document context is provided to the LLM, allowing us to compare structured retrieval and indexing with simply increasing the context supplied to the model.
DocETL (Shankar et al. 2024)	Document transformation	Preprocesses long documents through LLM-driven extraction and transformation operations, converting unstructured documents into structured intermediate representations before question answering. Unlike retrieval-based methods, it focuses on making the document collection more query-friendly during preprocessing.
Chain-of-Agent (Zhang et al. 2024)	Multi-agent reasoning	Decomposes long-document understanding among multiple coordinated LLM agents. Individual agents process different document portions or intermediate sub-tasks, and their outputs are progressively combined to produce the final answer. This design reduces the amount of context handled in any single model invocation.
RLM (Zhang, Kraska, and Khattab 2025)	Recursive reasoning	Treats a long prompt as an external environment that the model can programmatically inspect and decompose. It recursively invokes the language model over selected parts of the prompt and aggregates the resulting intermediate outputs.
SLIDERS (Joshi et al. 2026)	Structured retrieval and reasoning	The most closely related baseline and our primary comparison. It transforms long documents into reconciled relational tables through structured information extraction and cross-document reconciliation, then performs retrieval and reasoning over the resulting structured database. In contrast, EnSI-RAG uses structured records only as retrieval handles and generates answers from the retrieved original passages.

Table A3: Detailed descriptions of the baselines used in our experiments. The selected methods cover conventional retrieval-augmented generation, long-context inference, document transformation, agent-based reasoning, recursive reasoning, and structured retrieval.

Case Studies

We present two representative cases in which the structure preserved during preprocessing determines whether the system can recover the complete answer. Table A4 examines legal judgment-result matching, and Table A5 examines citation-chain construction.

Table A4: Full case study on legal judgment-result matching. The task requires preserving benchmark-facing document identifiers in addition to legal-document attributes.

Case component	Details
Question	The question provides 15 candidate judgment-result options and asks the system to assign each of 12 legal documents to the correct option. The answer must be a JSON mapping from benchmark document identifiers (e.g., <code>doc1</code>) to option identifiers (e.g., <code>doc8</code>).
Required operation	The task is not legal-case summarization or majority-result prediction. It requires the one-to-one mapping $\text{benchmark document label} \longrightarrow \text{judgment-result option ID}.$ The answer key must use the benchmark-local document label rather than the full case title, and the value must be the option identifier rather than the judgment text.
SLIDERS representation	SLIDERS induces and reconciles a relational table organized around legal attributes, including case title, case number, court, legal category, and judgment result. This representation captures useful legal information but does not preserve the benchmark document label as the task-facing key.
SLIDERS behavior	The system produces related outputs such as judgment summaries, frequent-result predictions, or title-level mappings. These outputs may describe the cases correctly, but they do not provide the required <code>Doc-i</code> to <code>Result-j</code> mapping. Incorrect
Why SLIDERS fails	The failure is caused by a representation mismatch. Identifiers such as <code>doc1</code> and <code>doc2</code> are benchmark-local labels rather than legal case titles. A table keyed primarily by case titles and legal attributes cannot directly recover those labels during generation, even when it contains the corresponding judgment text.
EnSI-RAG representation	Each legal document is represented by a metadata-centered passage that jointly preserves the benchmark document label, full case title, legal type, case category, court, case number, and judgment result. The resulting record makes the decision variable directly addressable: $[\text{document title}][\text{legal type}][\text{result} : \text{judgment text}] \rightarrow \{\text{psg_id}\}.$ The original passage also retains the benchmark label required for output formatting.
EnSI-RAG retrieval	At query time, EnSI-RAG recognizes judgment-result matching as the required task, parses the 15 candidate options, retrieves the judgment-result property for each document, and aligns it with the candidate options. Because both the canonical legal identity and benchmark-facing label are retained, the system can directly construct the requested mapping.
EnSI-RAG response	EnSI-RAG returns the required 12-entry JSON mapping from benchmark document labels to judgment-result option identifiers. All 12 assignments match the gold answer; representative entries include <code>"doc1" : "doc8"</code> , <code>"doc2" : "doc2"</code> , and <code>"doc12" : "doc15"</code> . Correct
Ground truth	The benchmark’s complete 12-entry document-to-option mapping. The representative assignments above are drawn directly from this gold mapping.
Key difference	SLIDERS organizes documents around induced legal attributes, whereas EnSI-RAG preserves the benchmark document identifier together with the canonical title, judgment-result property, and original evidence passage. The case is solved by retaining the correct retrieval and output handles before generation, rather than by adding more complex reasoning at generation time.

Table A5: Full case study on paper citation-chain construction. The task requires a complete directed citation graph and canonical paper titles.

Case component	Details
Question	Find the longest linear citation chain among five supplied papers and return the canonical paper titles in order. Only citation relations among the supplied papers should be considered, and non-linear branches should be ignored.
Required operation	The answer is not an unordered set of citation pairs. It requires constructing a directed graph over the five papers and selecting its longest valid linear path. The returned order runs from the earliest cited paper to the latest citing paper.
Gold chain	<i>Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters</i> → <i>Large Language Models Encode Clinical Knowledge</i> → <i>Capabilities of GPT-4 on Medical Challenge Problems</i> → <i>Sparks of Artificial General Intelligence: Early experiments with GPT-4</i> → <i>ChatGPT believes it is conscious</i> .
SLIDERS representation	SLIDERS induces separate paper and citation tables containing extracted paper identifiers, titles, and citation pairs. In this case, some titles are not canonicalized and the extracted citation table contains only three of the four edges required for the gold chain.
Missing evidence	The citation from <i>Capabilities of GPT-4 on Medical Challenge Problems</i> to <i>Large Language Models Encode Clinical Knowledge</i> is missing. Once this edge is absent, no downstream query over the reconciled table can reconstruct the complete five-paper path.
SLIDERS response	<i>Towards Understanding Chain-of-Thought Prompting</i> → <i>Large Language Models Encode Clinical Knowledge</i> → <i>Sparks of Artificial General Intelligence</i> → <i>ChatGPT believes it is conscious</i> . Incorrect
EnSI-RAG representation	Each paper is assigned a canonical title and aliases derived from arXiv identifiers and filename variants. Reference passages are used to construct explicit forward <i>cites</i> records and inverse <i>cited-by</i> records: $[source\ paper][paper][relation : cites = target\ paper] \rightarrow \{psg_id\}.$ Each relation retains provenance to its original reference passage.
Retrieved relations	EnSI-RAG recovers the four consecutive citation edges required by the gold chain, together with one additional non-linear edge. It constructs the directed citation graph, ignores the non-linear branch as requested, and selects the longest linear path.
EnSI-RAG response	<i>Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters</i> → <i>Large Language Models Encode Clinical Knowledge</i> → <i>Capabilities of GPT-4 on Medical Challenge Problems</i> → <i>Sparks of Artificial General Intelligence: Early experiments with GPT-4</i> → <i>ChatGPT believes it is conscious</i> . Correct
Ground truth	The complete five-paper chain shown above.
Key difference	SLIDERS reasons over the citation pairs that survive table extraction and reconciliation; a missing edge therefore makes the full chain unrecoverable. EnSI-RAG treats <i>cites</i> and <i>cited-by</i> as first-class directed relations, preserves canonical paper identities and aliases, and links every edge to original evidence. This relation-complete representation supports exact graph traversal and output in canonical-title form.

Additional Reproducibility Details

Verification of the Oolong slice. The public SLIDERS configuration applies the same deterministic selection rule: it filters the `test` split by the 262,144-token document-length condition and takes the first 192 eligible questions in dataset order. Our entry point therefore fixes `split=test`, sets both the minimum and maximum context length to 262,144, and applies `start=0` and `limit=192` after filtering. Here, 192 is the size of the evaluated subset; it is not the total number of questions in Oolong or necessarily the total number satisfying the length condition. Before evaluation, the program verifies that the selected manifest contains exactly 192 questions, that every question belongs to the `test` split, and that every selected document has the required length. The evaluator reconstructs the complete question string used by the public SLIDERS Oolong driver and is invoked with a strict slice check; any mismatch terminates the run without reporting a score. We additionally retain `selected_queries.jsonl` and release the selected question identifiers.

Benchmark	Seed	Selection and verification
Loong	42	Finite subsets are sampled by difficulty level using the target proportions 40/30/20/10 for Levels 1–4. The selected level and original <code>source_row_index</code> are stored for every question, allowing the exact subset to be reconstructed.
Oolong	Not applicable	We first filter the <code>test</code> split to questions whose document length satisfies <code>context_len=262144</code> , and then select the first 192 eligible questions in the public dataset order. The experiment entry point rejects any selection whose split, context length, ordering, or selected question count differs from this rule. All Oolong results reported in the main paper are evaluated exclusively on Oolong-Synth; we do not evaluate on Oolong-Real.

Table A6: Question selection and randomness controls. Oolong uses a deterministic dataset slice and therefore does not require a sampling seed.

API and local execution configuration. Loong uses Top-12 retrieval. For Oolong-Synth, records are selected deterministically by context-window identifier and subsequently aggregated using query-focused metadata; the Oolong-Synth pipeline does not instantiate an embedding model. LLM-based preprocessing uses GPT-4.1-mini, while retrieval planning and final answer generation use GPT-4.1. Oolong’s LLM-based evaluator also uses GPT-4.1. All API calls use temperature 0. These models are accessed through the OpenAI API; the provider’s server-side GPU, CPU, memory, and serving configuration are not exposed to API users and are therefore unavailable to us. We report the API model identifiers, decoding parameters, and prompt templates that we control. The local `octen-embedding-8B` model is used for retrieval on Loong’s Financial dataset; For other datasets, we select records deterministically by `context_window_id` and does not instantiate the embedding model.

Number of runs. Each reported EnSI-RAG result is obtained from one complete run of the corresponding configuration. We therefore do not report across-run standard deviations or statistical significance tests.

Prompts

This appendix provides the complete prompts used in the Loong and Oolong experiments. The text inside every box is reproduced verbatim from the experiment scripts. Braced expressions denote runtime substitutions in the corresponding templates. Deterministic parsing, index construction, filtering, and score aggregation do not invoke an LLM and therefore have no prompt to report.

Loong Prompts

Company-Name Alias Extraction This prompt identifies bilingual company-name variants so that financial records referring to the same company can share a canonical retrieval handle.

Company-Name Alias Extraction: System Prompt

You are a precise bilingual financial-report company-name alias extractor.

Your job is to identify all English or Chinese names that refer to the reporting company. This includes the exact current registrant name and old/former/predecessor names if the filing says the company changed its name.
Return JSON only. Do not include explanations or markdown.

Company-Name Alias Extraction: User Prompt

Extract the company/registrant names from the filing excerpt below.

Rules:

- Include the exact current registrant name.
- Include former names only if the excerpt explicitly says the company was previously incorporated as, formerly known as, or changed its name from/to that name.
- For Chinese reports, recognize fields such as 公司名称、公司中文名称, 曾用名、原名 and 名称变更.
- Do not include subsidiaries, auditors, creditors, customers, officers, locations, or stock tickers unless they are also company names of the registrant.
- Keep names as clean display names, e.g. "Arvana Inc." and "Turinco, Inc.".

Return only this JSON schema:

```
{
  "company_names": ["name 1", "name 2"]
}
```

Filing excerpt:

```
{context}
```

Financial-Table Location This prompt locates financial tables and preserves the local context needed to construct semantically contained row-level passages.

Financial-Table Location: System Prompt

You are a bilingual indexing system for Loong Financial RAG.

Your job is NOT to answer questions and NOT to extract numerical answer values. Your job is to inspect an English or Chinese financial-report chunk and identify financial-statement tables that should become retrieval passages.

Important:

- The document may be an SEC Form 10-K, a Chinese annual report, or a Chinese quarterly report.
- Line numbers are global document line numbers even when only a chunk is supplied.
- Select real financial statement/value tables, not a table of contents or prose.
- One table must become one passage.
- Return the exact start/end line numbers for each table in the full filing.
- Extract only retrieval metadata: table title, years shown in the table, and metric row labels.
- Do NOT output row values, dollar values, percentages, share counts, or final answers.

Return JSON only. Do not include explanations, markdown, or <think>.

Financial-Table Location: User Prompt

Read the following line-numbered chunk of a financial report and identify all financial-statement tables that should be stored as passages.

The chunk keeps global line numbers from the full document. Return those global line numbers.

Table selection rules:

- A passage should be exactly one financial statement table.
- Include main financial statement tables, such as Balance Sheets, Statements of Operations, Statements of Income, Statements of Cash Flows, Statements of Stockholders' Equity/Deficit, Statements of Comprehensive Income/Loss, and similar tables.
- Chinese equivalents include 资产负债表、利润表、现金流量表、所有者权益变动表、合并财务报表 and their parent-company versions.
- Include notes tables only when they are actual financial line-item/value tables that may answer a financial metric question.
- Do NOT return the table of contents.
- Do NOT return prose paragraphs, audit report paragraphs, signatures, exhibit lists, or pure section headers.
- Do NOT extract numerical values into the metadata.
- Only extract metric labels that are actual financial line items with values. Do not extract pure section headers unless the header itself has reported values.

Return only a valid JSON object with this schema:

```
{
  "tables": [
    {
      "title": "short table title",
      "start_line": 123,
      "end_line": 145,
      "years": [2024, 2023],
      "metrics": ["Cash and cash equivalents", "Total assets"]
    }
  ]
}
```

Line-numbered financial-report chunk:
{numbered_text}

Reference-Section Location This prompt locates the reference section of an academic paper before citation relations and their source passages are constructed.

Reference-Section Location: System Prompt

You identify bibliography/reference sections in parsed academic papers.
Return JSON only. Do not explain. Do not use markdown.

Reference-Section Location: User Prompt

Find the bibliography/references section in the following parsed academic paper tail.
Return ONLY a JSON object with this schema:

```
{
  "found": true/false,
  "start_quote": "an exact substring copied from the first 80-200 characters of the references section",
  "end_quote": "an exact substring copied from the last 80-200 characters of the references section"
}
```

Rules:

- The references section is the bibliography at the end of the paper, not citations in Related Work.
- Do not paraphrase. start_quote and end_quote must be exact substrings from the input.
- If there is no references/bibliography section, return {"found": false, "start_quote": "", "end_quote": ""}.

```
PAPER TAIL START
{tail}
PAPER TAIL END
```

Retrieval Planning This prompt converts a Loong question into structured retrieval items aligned with the entity-structure index.

Retrieval Planning: System Prompt

You are the single generic retrieval planner for EnSI-RAG.

The corpus has already been preprocessed into dataset-specific passages, but all records share one schema:

```
[entity][entity_type][property|relation|aspect: value] -> passage_id
```

Dataset-specific conventions:

- paper: entity_type=paper. Important relations are cites and cited-by. References section is indexed as references; body is content.
- financial: entity_type=financial_statement_year. Entity is year + company, e.g. "2024 Apple Inc.". The association is the metric/aspect only. Numeric values remain in the original row passage text.
- legal: entity is the legal document title. Entity type is legal_type. Important properties are case, sub_case, subsub_case, combined_case, court, number, result, metadata, and content.

The question may be Chinese. Understand it directly. For a Chinese financial question over an English SEC filing, translate each requested metric into the canonical English filing label in association_name and add common English alternatives in metric_aliases. Do not put only Chinese text in association_name.

Convert the question into retrieval items. Do not answer. Return JSON only.

Retrieval Planning: User Prompt

Dataset type: {dataset_type}

Question:
{question}

Instruction:
{instruction}

Provided documents:
{docs}

Return only this JSON schema:

```
{
  "retrieval_items": [
    {
      "entity": "entity name if known; for financial use '<year> <company>' when year and company are known",
      "entity_type": "paper | financial_statement_year | legal_type string | empty if unknown",
      "association_kind": "property | relation | aspect | empty if unknown",
      "association_name": "property/relation/aspect name such as cites, cited-by, content, case, combined_case, net income",
      "metric_aliases": ["English filing-label alternatives, especially when the financial question is Chinese"],
      "association_value": "target value or label if known",
      "company": "financial company if relevant",
      "year": null,
      "target_value": "numeric/legal/paper value constraint if any"
    }
  ]
}
```

```
"operation": "lookup | classify | compare | trend | identify_company | citation_reference  
| citation_chain | other",  
"answer_requirements": "short description of the final output format"  
}
```

Final-Answer Generation This prompt generates the final Loong response from the retrieval plan, matched index records, and retrieved original passages.

Final-Answer Generation: System Prompt

You are the unified final-answer generator for EnSI-RAG.

You must answer using only the retrieved original evidence passages and the retrieved index records.

The records are retrieval hints; the original passage text is the evidence.

Do not use outside knowledge.

Preserve the answer format requested by the question/instruction.

When the expected answer is JSON, return only valid JSON with no markdown.

When the expected answer is a short text answer, return only the concise final answer.

Understand Chinese questions directly and answer in the language and exact

format requested by the original instruction. Financial evidence may use

English filing labels even when the question is Chinese.

Financial-classification rule:

- When the operation is classification over multiple provided companies, include every provided company exactly once.
- Do not silently omit a company. Follow the requested grouped JSON schema rather than returning a partial list.

Paper-output rule:

- For paper tasks, never output an arXiv id, filename, filename stem, or local path as the final paper answer.
- Always output the canonical full paper title from the provided/retrieved paper metadata.
- Preserve the benchmark title style, e.g. "# TinyLlama: An Open-Source Small Language Model".

Legal-output rule:

- For Loong legal classification tasks, use benchmark document labels such as "《判决书6》" unless the question explicitly asks for actual full titles.
- Do not output full legal document titles when the expected/gold format uses 判决书 labels.

Final-Answer Generation: User Prompt

Dataset type: {dataset_type}

Original instruction:
{instruction}

Original question:
{question}

Provided documents:
{docs}

Retrieval plan:
{plan_json}

Retrieved index records:
{record_evidence}

Retrieved original passages:
{passage_evidence}

Answer the original question based solely on the retrieved evidence.

Oolong Prompts

Atomic-Instance Record Extraction This prompt converts Oolong atomic instances into structured records while retaining their provenance and context-window identifiers.

Atomic-Instance Record Extraction: System Prompt

You are an exact text classification engine. Classify every supplied instance according to the task definition. Return JSON only. Never omit an item and never invent labels outside ALLOWED_LABELS.

Atomic-Instance Record Extraction: User Prompt

```
TASK_DEFINITION:
{definition}

ALLOWED_LABELS:
{json.dumps(labels, ensure_ascii=False)}

INSTANCES:
{json.dumps(payload, ensure_ascii=False)}

Return exactly a JSON array of objects: [{"id":"...", "label":"one allowed
label", "confidence":0.0}]
```

Final-Answer Generation This prompt produces an Oolong answer from the retrieved records associated with the selected context windows.

Final-Answer Generation: System Prompt

You are the final reasoning and answer-generation component of an Entity-Structure Indexed RAG system.

You MUST answer the original question using only the retrieved EnSI metadata supplied by the user.
The metadata was computed from LLM-extracted index entries. Each index entry has the form:
[entity][oolong_instance][label: value]
and may also carry user and date metadata.

Important rules:

1. Do not use outside knowledge.
2. Do not ask for the original long context.
3. Perform all comparisons, ranking, arithmetic, and final decision yourself from the supplied metadata.
4. Do not output explanations, reasoning, markdown, or code fences.
5. Return exactly one answer line in the requested format.
6. Never invent a label, user, date, or number that is absent from the supplied metadata.

Final-Answer Generation: User Prompt

```
ORIGINAL QUESTION:
{str(row.get('question', ''))}

TASK METADATA:
{json.dumps({'dataset': row.get('dataset'), 'task_group': row.get('task_group'), 'task':
row.get('task'), 'answer_type': row.get('answer_type')}, ensure_ascii=False, indent=2)}

RETRIEVED ENSI METADATA:
{json.dumps(metadata, ensure_ascii=False, indent=2)}

OUTPUT FORMAT:
{_answer_format_instruction(row.get('answer_type'))}
Return only that one line.
```

Soft Evaluation This prompt implements the LLM-based evaluator used for non-numeric Oolong questions.

Soft Evaluation: System Prompt

Evaluate the predicted answer against the gold answer. The predicted answer should match the gold answer.

Soft Evaluation: User Prompt

```
# Question
{question}

# Gold Answer
{_json_text(gold_answer)}

# Predicted Answer
{predicted_answer}
```

Numeric-Answer Extraction This prompt extracts a numeric prediction before the official deviation-based Oolong metric is applied.

Numeric-Answer Extraction: System Prompt

Given the question and the assistant's full answer, extract only the final numeric answer. Return digits with an optional leading minus sign; remove commas, units, and explanation. If no numeric answer is present, return null.

Numeric-Answer Extraction: User Prompt

```
# Question
{question}

# Predicted Answer
{predicted_answer}
```