CWAGraph: Retrieving What Was Never Explicitly Identified by Graph-based RAG

Zhitao Yin[1] **Ruiheng Zhu**[2] **Ganlin Xu**[1] **Hongru Hou**[1] **Jiaqing Liang**[1] **Deqing Yang**[1*]
[1]School of Data Science, Fudan University
[2]Department of Computer Science and Engineering, The Chinese University of Hong Kong
ztyin26@m.fudan.edu.cn, yangdeqing@fudan.edu.cn

Abstract

Unlike chunk-based Retrieval-Augmented Generation (RAG) methods, which retrieve information only from text passages, graph-based RAG methods improve their performance through entity-relation graphs. However, existing graph-based RAG methods typically represent graph facts under the Open World Assumption (OWA), where an absent edge (relation) means unknown rather than false. This principle is problematic for many closed corpora including technical manuals and regulations, where rules are generally stated through scope-bounded statements (such as "all X, except Y" or "only Y") named as *local completeness statements* (LCSs) in this paper. LCSs apply rules to every unmentioned in-scope entity which cannot be explicitly and completely represented by the OWA graph. To address this problem, we propose **CWAGraph**, which is a graph-based RAG framework augmented with a Closed World Assumption (CWA) layer representing local scope and boundaries for LCSs. The CWA layer identifies which entities are governed by each statement and makes the corresponding closed-world evidence retrievable from those entities, thus enhancing the performance of question answering (QA) involving LCS (called as LCS resolution in this paper). To better evaluate RAG systems' performance on LCS resolution, we further introduce a new QA benchmark **SetQA** consisting of 493 QA instances involving LCSs across two source domains and three difficulty levels. Our experiments on SetQA reveal that CWAGraph consistently outperforms chunk-based, long-context, and graph-based RAG baselines. Our code is available at https://github.com/YinZT1/CWAGraph

1 Introduction

Retrieval-Augmented Generation (RAG) [21, 12] has advanced knowledge-intensive tasks by coupling large language models (LLMs) with external knowledge retrieved at inference time, thereby improving factual grounding and generation quality. Standard RAG segments documents into fixed-length chunks retrieved via dense vector similarity [19], overlooking the relationships between entities. In contrast, graph-based RAG methods, e.g., GraphRAG [8], LightRAG [13], HippoRAG [14], and their successors, cast the corpus as a graph consisting of entities and relations, enabling cross-document reasoning and improving retrieval efficiency on the queries that require aggregating evidence across documents.

However, existing graph-based RAG approaches [8, 13] typically store structured knowledge in an entity–relation layer (graph) under the Open World Assumption (OWA), referred to as the **OWA layer** in this paper. In OWA layer, a stored edge (relation) asserts a fact and an absent edge means an unknown rather than false fact [28], which is insufficient for the retrieval over closed corpora. In a closed corpus such as exemption clauses, regulatory standards, or technical manuals [15, 3], many rules are stated through scope-bounded statements, e. g., "all X, except Y" or "only Y".

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

These statements define a local scope and a boundary within that scope, which are named as *local completeness statements* (LCSs). For example, an LCS "all noble gases are chemically inert under standard conditions, except xenon" implies that argon (Ar) is in fact inert because "the noble gases: He, Ne, Ar, Kr, Xe, Rn". As shown in Figure 1, this LCS is represented as ordinary triples `(noble gases, has_property, inert)` and `(xenon, is_exception_to, inert)`. These triples encode the boundary (exception) case of xenon, but not the closed-world semantics that xenon is the only boundary case. Therefore, even if the OWA layer contains the relevant rule, hierarchy, and queried entities, it cannot be concluded from the graph that argon is also inert, because the absence of an argon-exception edge remains unknown rather than false. This failure is critical in closed corpus retrieval, where unmentioned in-scope cases often determine whether a rule applies.

To address this limitation, we propose a novel graph-based RAG framework **CWAGraph**, which augments the OWA layer with a dedicated **Closed World Assumption (CWA) layer**. Unlike knowledge graph (KG) completion [2, 27], the designed CWA layer stores an LCS as a statement-level closed-world object and indexes it over its resolved scope, so that the original statement can be retrieved from any in-scope entity (including entities not named in the source), without asserting new edges in the OWA layer. The CWA layer contains three components (formalized in §3): *hierarchy* recovers the scoped universe, *peer* validates scoped entities attached at a compatible granularity, and *complement* keeps each statement self-contained by preserving its predicate, local boundary, and source evidence. Together, these structures recover the scope-and-boundary information that an OWA layer does not structurally preserve.

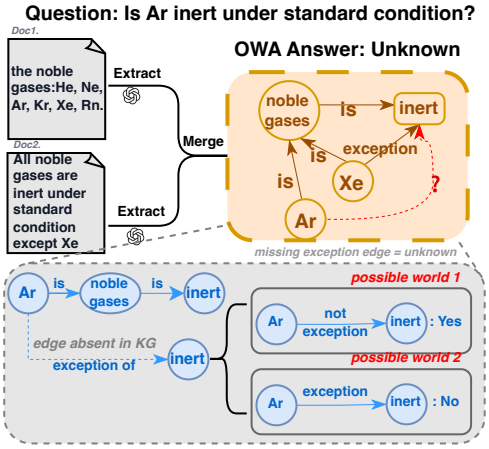

Figure 1: OWAGraph cannot resolve LCSs.

Furthermore, to better evaluate RAG systems' performance on question answering (QA) involving LCS that is called as LCS resolution in this paper, we construct **SetQA**, the first benchmark testing whether RAG systems can recover the closed-world structure declared by LCSs. Built from 1,373 documents across two source domains, SetQA contains 493 QA instances and spans three progressively harder task levels: *Easy-level* decides whether a queried entity falls within the induced boundary; *Medium-level* checks whether the resolved boundary can be unfolded into a set of covered entities; and *Hard-level* checks whether this unfolding is exhaustive. Our experiments upon SetQA reveal that CWAGraph consistently outperforms both state-of-the-art chunk-based and graph-based RAG baselines.

In summary, our contributions in this paper include:

1. We propose CWAGraph which augments the OWA layer with a dedicated CWA layer (hierarchy + peer + complement), making LCSs retrievable for in-scope entities not named in the source, without additional training.

2. We construct a new benchmark SetQA for evaluating RAG systems' capabilities of recovering the closed-world structure declared by LCSs, which spans two source domains and three task levels.

3. Our extensive experiments demonstrate that CWAGraph consistently outperforms chunk-based and graph-based RAG baselines on SetQA. Further analysis shows that CWAGraph's gains come from LCS resolution with addressable CWA records, rather than merely retrieving more text.

2 Related Work

RAG and Graph-based Retrieval Standard RAG systems retrieve text chunks via dense vector similarity [21, 12], with some extensions including agentic retrieval [38] and long-context LLMs as an alternative to fine-grained retrieval [36]. Graph-based RAG methods replace flat retrieval with entity–relation graphs to enable cross-document reasoning: GraphRAG [8] indexes the corpus as community-level summaries; LightRAG [13] combines graph indexing with dual-level retrieval; HippoRAG [14] uses personalized PageRank over an entity graph; HopRAG [23] augments retrieval

with logical hops over passage graphs. Other structured RAG variants, such as PathRAG, KAG, and PIKE-RAG, explore orthogonal retrieval and reasoning pipelines [4, 22, 32]. Related learning paradigms have also been explored in recommendation settings [17, 18], which are orthogonal to the closed-world retrieval problem studied here. These graph-based and structured RAG methods remain under OWA: an absent edge is unknown rather than false, so the local boundaries induced by LCSs are never first-class retrievable objects. Comparatively, our CWAGraph keeps the OWA layer for ordinary relational evidence and adds a dedicated CWA layer for scope-bounded boundaries.

Completeness Assumption in KGs Reasoning under closed-world or partially-closed assumptions has a long history in knowledge representation. Reiter [29] formalized the CWA for deductive databases, and Etzioni et al. [9] introduced local closed-world inference for planning. In the KG community, AMIE [10] mines logical rules under partial completeness, and Krompaß et al. [20] use a local CWA for KG embedding. More recent work specifies or predicts which parts of an existing KG can be treated as complete [7, 28]. These approaches annotate completeness over an already-constructed KG. We instead extract local completeness declarations directly from raw texts, resolve their scope and boundary, and materialize them as a retrievable CWA layer for RAG.

KG Structural Extensions A separate line of work enriches KG representation rather than the world assumption under which it is interpreted. Beyond RAG, prior work has long studied higher-order KG representation, including hypergraph learning, n-ary relation embeddings, hyper-relational KG embedding, and message passing [40, 33, 31, 11]. HyperGraphRAG [24] generalizes binary edges to n-ary hyperedges to capture relations among more than two entities. While this addresses representational arity, it remains open-world, i. e., a missing hyperedge is still unknown rather than false. Our CWAGraph addresses an orthogonal axis (closure rather than arity), and we therefore do not include HyperGraphRAG as a baseline.

3 From OWA Graphs to a CWA Layer

Definition 1: OWA and CWA Graph. A graph used in graph-based RAG is represented as

$$\mathcal{G} = (E, R),$$

where E is the entity set and R is the relation (edge) set. Under OWA, the stored edges are positive evidence and a missing edge means unknown [28]. Under CWA, a *scope* is declared complete: any in-scope entity absent from the stated side of a rule is inferred to belong to the *complement* [29, 9].

Definition 2: Local Completeness Statement (LCS). An LCS σ declares a scope $U_\sigma \subseteq E$, a predicate ϕ_σ, and a set of explicitly identified entities $B_\sigma \subseteq U_\sigma$. Under CWA, σ induces

$$U_\sigma = B_\sigma \cup R_\sigma, \qquad R_\sigma \cap B_\sigma = \emptyset,$$

where R_σ represents the set of entities without explicit declaration but in the scope U_σ. Recall the example in Figure 1, U_σ is the set of all noble gases, ϕ_σ is inertness, $B_\sigma = \{\mathrm{Xe}\}$, and $R_\sigma = \{\mathrm{He}, \mathrm{Ne}, \mathrm{Ar}, \mathrm{Kr}, \mathrm{Rn}\}$.

Definition 3: Graph Representation in CWAGraph. In our CWAGraph, there exists a comprehensive graph represented as

$$\mathcal{G}_{\mathrm{CWA}} = (E, R, H, P, C),$$

where set E, R constitute the graph's base OWA layer and set H, P, C constitute the graph's CWA layer. Here H stores hierarchy edges, P stores same-granularity peer cues, and C stores self-contained LCS constraints with their resolved scope and boundary side.

Proposition 1: Finite k-hop OWA retrieval cannot resolve LCSs. *Let $\mathcal{N}_k(e; \mathcal{G})$ be the k-hop neighborhood of an entity e in an OWA graph $\mathcal{G}$. For any finite k, σ cannot be resolved for R_σ from $\mathcal{N}_k(e; \mathcal{G})$ alone.* **Proof.** See Appendix B.1.

Proposition 2: Hierarchy and complement are jointly necessary for resolving LCSs. Proof. See Appendix B.2.

Observation 1: Peer edges are necessary to detect and correct hierarchical errors. In practice, hierarchical errors may propagate, since an incorrect parent–child edge would misplace an entire subtree. Such errors can be detected and corrected by the independent signals from peer edges.

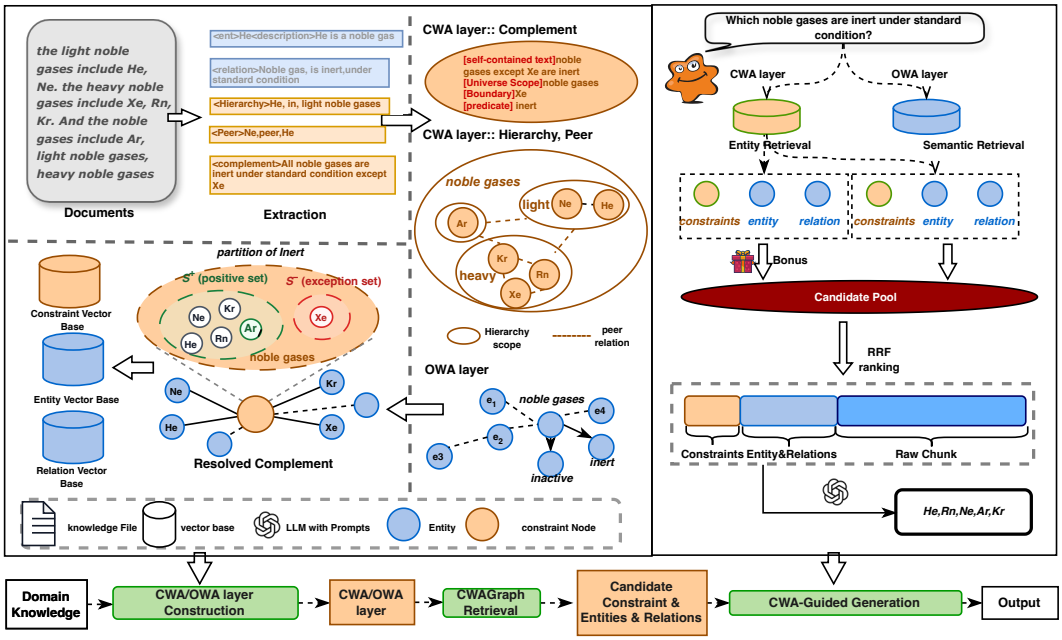

Figure 2: CWAGraph resolves LCSs into entity-addressable CWA layer structures for retrieval.

4 Methodology

As shown in Figure 2, our framework has two major stages. In *offline* stage, the CWA layer and OWA layer are first constructed from text and stores the resolved results for retrieval. In *online stage*, the framework retrieves results from both the CWA layer and OWA layer, then fuses the resulting evidence for generation.

4.1 Construction Strategy

To make each LCS retrievable from any in-scope entity, constructing the two layers of CWAGraph includes *CWA Constraint Joint Extraction*, *Local Completeness Statement Resolution*, and *CWAGraph Storage*.

CWA Constraint Joint Extraction. We jointly extract ordinary knowledge triples (entities and relations) based on OWA and typed CWA constraints, i.e., *Hierarchy*, *Peer*, and *Complement*, from each chunk d in the corpus $\mathcal{K}$ through a single LLM call. The call returns $(E_d, R_d, H_d, P_d, C_d^{\text{raw}})$.

a) Hierarchy & Peer. Given a document chunk d, CWAGraph extracts hierarchy constraints H_d and peer constraints P_d. H_d records membership from a specific entity to broader universal nodes. P_d records the relations of the same granularity (peer edges), indicating that peer entities should be placed at the same hierarchy level. Taking Figure 2 as an example, a hierarchy edge is stored as $(\text{Ar}, \texttt{in}, \text{noble gases})$ and a peer edge is stored as $(\text{Ar}, \texttt{peer}, \text{Ne})$.

b) Complement. CWAGraph extracts complement constraints c as self-contained structured LCS records. Raw LCS evidence is often context-dependent due to implicit subjects, table structure, and other document-level context. The complement makes the LCS self-contained by explicitly recording its universe scope u_c, boundary b_c, predicate ϕ_c, and source evidence s_c. In Figure 2, "all noble gases are inert under standard conditions except Xe" is converted into

`[self-contained text]`: all noble gases are inert under standard conditions except Xe
`[universe scope]`: noble gases `[boundary]`: Xe `[predicate]`: is inert

c) Entities & Relations. While the CWA layer is constructed, the entities and relations (E_d, R_d) of standard KG are also extracted from the same chunk. These knowledge triples are ordinary positive facts based on OWA and stored in the OWA layer. The extraction prompt template is provided in Appendix G.1.

LCS Resolution. This step turns the text references u_c and b_c into entity sets, so that the LCS applies to in-scope entities that the statement never names. It consists of three parts: conflict resolution, scope grounding, and boundary resolution.

Conflict Resolution. Hierarchy edges extracted from different chunks can disagree, and a single wrong edge can misplace a whole subtree. CWAGraph resolves such conflicts with two rules. (i) *Multiple parents*: an entity attached both to a category and to one of its sub-categories, as in $(A, \texttt{in}, B)$, $(A, \texttt{in}, C)$, and $(B, \texttt{in}, C)$, keeps only one parent. (ii) *Peer consistency*: a peer edge $(a, \texttt{peer}, b)$ implies that a and b share a parent, which provides a check independent of the hierarchy edges. For conflicts that these rules cannot settle, an LLM reads the involved hierarchy and peer edges together with their source descriptions and deletes the wrong edges. The prompt is in Appendix G.3. For example, suppose Xe is placed under heavy noble gases, while Kr is attached both to heavy noble gases and to noble gases. The peer edge $(\text{Kr}, \texttt{peer}, \text{Xe})$ indicates that Kr shares the parent of Xe, so the edge $(\text{Kr}, \texttt{in}, \text{noble gases})$ is removed.

Below, $\mathrm{Ground}(x)$ maps a text reference x to its node in H and returns this node together with all its descendants, and $\mathrm{Prune}(S)$ checks an entity set S against the peer edges and removes the entities that they contradict.

Scope Grounding. Scope grounding turns the universe reference into an entity set. CWAGraph first finds the entities related to u_c in H and P. It then grounds u_c through the hierarchy into the entity set $U_c = \mathrm{Ground}(u_c)$ and applies peer-guided pruning to remove misplaced entities:

$$\widehat{U}_c = \mathrm{Prune}(U_c).$$

Boundary Resolution. CWAGraph grounds the boundary reference in the same way and splits the universe scope into the boundary set $\widehat{B}_c$ and the remaining set R_c:

$$B_c = \mathrm{Prune}(\mathrm{Ground}(b_c)), \qquad R_c = \widehat{U}_c \setminus \widehat{B}_c, \qquad c^* = (c,\, \widehat{U}_c,\, \widehat{B}_c,\, R_c, s_c).$$

The resolved complement c^* thus instantiates the LCS σ of Definition 2. CWAGraph makes two further choices. It does not expand the LCS into implied OWA triples: $\widehat{B}_c$ and R_c stay as fields of the CWA record, and the OWA layer is unchanged. It also keeps ϕ_c and s_c as written, instead of mapping the predicate to a fixed mode such as allowed or prohibited, so the representation stays flexible.

CWAGraph Storage. To make each resolved LCS retrievable from its in-scope entities, CWAGraph stores the CWA layer separately from the OWA layer while sharing the same entity identifiers. The OWA layer stores ordinary positive entity-relation facts (E, R), while the CWA layer stores the constraints in two forms. First, hierarchy and peer constraints are stored as typed binary edges over entities, like $(a, \texttt{in}, b)$ and $(a_1, \texttt{peer}, a_2)$. Second, each resolved complement c^* is stored as a hyperedge-like constraint object, linking to all entities in the universe scope $\widehat{U}_c$ and preserving the boundary set $\widehat{B}_c$, the remaining set R_c and the predicate ϕ_c, the source evidence s_c.

$$\mathcal{S}_{\mathrm{CWA}} = (H, P, C^*, A_{C\leftrightarrow E}), \quad C^* = \{\, c_i^* \,\}_{i=1}^m, \quad c_i^* = (U_i, B_i, R_i, \phi_i, s_i),$$
$$A_{C\leftrightarrow E} = \{\, (c_i^*, e) \mid c_i^* = (U_i, B_i, R_i, \phi_i, s_c) \in C^*,\ e \in U_i \,\}.$$

Here, $A_{C\leftrightarrow E}$ is an entity-addressing index for retrieval, not an additional semantic constraint. This storage supports queries from any in-scope entity to all LCSs governing it, and from a resolved complement to the full scope it covers, thus making LCSs retrievable from any in-scope entity without rewriting any existing OWA triples.

Step-by-step pseudocode for the offline construction stage is provided in Appendix C.2. And detailed symbol and operator definitions are provided in Appendix C.1.

4.2 Retrieval Strategy

In this section, we introduce a dual retrieval strategy to pair the CWAGraph's structure, which can match user questions with relevant entities, relations, and closed-world constraints. Each retrieval module combines an entity-anchored path with a semantic path. The entity-anchored path starts from query entities, while the semantic path compensates for parsing or fuzzy-matching misses.

CWA Layer Retrieval. Given a query q, CWAGraph first extracts query entities and fuzzy-matches them to the stored entities, producing the matched entity set E_q. From E_q, the entity-anchored CWA path retrieves the resolved complements linked to these entities through the entity-addressing index $A_{C \leftrightarrow E}$, together with the local family context induced by hierarchy and peer constraints. The CWA semantic path retrieves resolved complements and CWA layer entities by cosine similarity. Each retrieved complement is represented by its self-contained text. The retrieved complements form the complement pool $\mathcal{Q}_{\mathrm{com}}(q)$, and the retrieved entities form the entity pool $\mathcal{Q}_{\mathrm{ent}}(q)$. For entity-anchored complements, we add a slight bonus during reranking, reflecting that these constraints are reached through the resolved LCS-to-entity links rather than by text similarity alone. The family context is not placed into either pool; it is carried forward to the generation stage.

OWA Layer Retrieval. In parallel, the OWA layer retrieves ordinary relational evidence from the storage. The anchored OWA path uses the same matched entities E_q to retrieve neighboring triples and entity descriptions, and the semantic OWA path retrieves additional entities and relations by similarity. The retrieved OWA entities are merged into $\mathcal{Q}_{\mathrm{ent}}(q)$, while the retrieved OWA relations form the relation pool $\mathcal{Q}_{\mathrm{rel}}(q)$.

Fusion / Rerank. Finally, each pool is ranked separately:

$$\mathcal{R}(q) = \big(\mathrm{RRF}^{+}(\mathcal{Q}_{\mathrm{com}}(q)),\ \mathrm{RRF}(\mathcal{Q}_{\mathrm{ent}}(q)),\ \mathrm{RRF}(\mathcal{Q}_{\mathrm{rel}}(q))\big),$$

where RRF ranks a pool by reciprocal-rank fusion [6] of its BM25 [30] and cosine-similarity rankings, and RRF^{+} additionally gives entity-anchored complements a small bonus. Step-by-step pseudocode for the online retrieval stage is provided in Appendix C.3.

4.3 Generation Strategy

In this section, we introduce a CWA-guided generation strategy, which organizes evidence into typed context blocks rather than a flat retrieved list. This generation strategy includes *CWA Layer Knowledge Fusion* and *Final Generation Context*.

CWA Layer Knowledge Fusion. The CWA layer contributes two typed blocks: a complement block and a family block. The complement block contains the reranked LCS records from Section 4.2. The family block contains the hierarchy and peer context associated with the retrieved complements. We select family entries by intersecting the family context with the universe scopes $\widehat{U}_c$ of the retrieved complements, so that the generator sees the local comparison structure without receiving the full hierarchy.

Final Generation Context. The final generation input concatenates four typed blocks:

$$\mathrm{Context}(q) = \textsc{Family} \oplus \textsc{Complement} \oplus \mathrm{OWA} \oplus \textsc{Chunk},$$

where FAMILY and COMPLEMENT are the two CWA blocks above, OWA is the formatted OWA entity-relation evidence, CHUNK is the remaining retrieved text chunks under the context budget, and $\oplus$ is ordered concatenation. The COMPLEMENT, OWA, and CHUNK blocks are filled from $\mathcal{R}(q)$ in rank order. Thus, the generator receives ordinary entity-relation evidence, source-level complement descriptions, and a family context selected by the resolved CWA layer, rather than a full materialization of all implied facts.

5 Experiments

We conduct experiments to answer the following four research questions. **RQ1**: Whether CWAGraph improves LCS retrieval across source domains? **RQ2**: Can CWAGraph retrieve relevant LCSs and leverage them for implicit in-scope entities? **RQ3**: Which design choices of CWA layer drive performance gain more? **RQ4**: How much is the remaining headroom from replacing extracted H, P and C with oracle components?

5.1 Experimental Setup

SetQA Construction Given that few QA samples involving LCSs exist in current RAG benchmarks, we specially construct **SetQA** that is the first benchmark dedicated to evaluate RAG systems' capabilities on the QA involving LCSs.

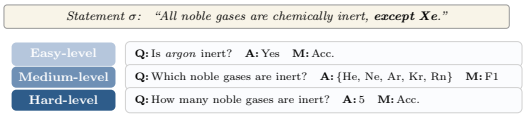

Figure 3: QA samples of SetQA derived from one LCS.

Table 1: SetQA's statistics.

Source Domain	#Docs	#Qs	#Easy	#Medium	#Hard
Technical Manuals	1,000	262	108	89	65
Food Safety Regulations	373	231	81	80	70

We collect QA data of SetQA from two source domains: **Technical Manuals**, consisting of 1,000 technical documents, and **Food Safety Regulations**, consisting of 373 regulatory articles. We first use an LLM to extract candidate LCSs from documents, followed by manual verification. Following previous benchmarks stratifying questions by difficulty [5], we build three progressively harder levels of questions around verified LCSs in these sources. As illustrated in Figure 3, **Easy-level** questions test whether a resolved LCS can be used for an implicit in-scope entity. **Medium-level** questions test whether the system can recover the in-scope entities covered by the resolved boundary. **Hard-level** questions test whether this recovery is complete by asking for the exact size of the in-scope set. Table 1 summarizes the statistics of SetQA, and its construction details are provided in Appendix A.

Baselines We compare CWAGraph against seven publicly available baselines. **Naive RAG** [21, 12] is the classic chunk-based RAG method; **Agentic RAG** [38] also adopts chunk-based index with LLM-based query decomposition and sufficiency checking; **Long-Context LLM** is a BM25-ranked [30] whole-document retrieval baseline with a larger context budget; **LightRAG** [13], **GraphRAG** [8], **HopRAG** [23], and **HippoRAG** [14] are the graph-based RAG baselines based on OWA graphs. To ensure fair comparisons, all methods use the same source documents, context budget, answer-generation prompt, and evaluation prompt. We use `text-embedding-3-large` as the embedding model and **GPT-5.2** as the backbone LLM for graph construction, answer generation and evaluation. All parameters are reported in Appendix D.

Evaluation Metrics Each SetQA level uses a metric matched to its answer form. **Easy-level Accuracy** is binary yes/no accuracy. **Medium-level F1** is set-level F1 against the gold applicable set, following set-level evaluation conventions for multi-answer QA [1], with set alignment performed by an LLM judge [39]. **Hard-level Accuracy** is exact-match accuracy on the integer count of the applicable set. The three levels progressively test stronger forms of LCS application: Easy tests whether one queried entity can be placed under the LCS, Medium tests whether the applicable side can be partially recovered, and Hard tests whether it has been fully recovered. **Task-Macro Score** is the unweighted average of these three level scores.

For RQ2, **Hit@K** means whether the top-K retrieved evidence contains the gold evidence. **Easy-level Accuracy** | **Hit@5** measures answer accuracy on questions where **Hit@5** is true. Baseline Hit@K diagnostics at Medium and Hard levels are reported in Appendix E.4.

Additional baseline settings, prompt templates, and LLM/token usage details are provided in Appendix D, Appendix G, and Appendix H, respectively.

5.2 Main Results (RQ1)

Table 2 reports all compared methods' performance on the two domains of SetQA, from which we have the following observations and analysis.

Overall comparison across methods. CWAGraph obtains the highest Task-Macro Score in both source domains, improving over the strongest baseline from 37.5 to 79.6 on Technical Manuals and from 32.3 to 64.3 on Food Safety Regulations. Interestingly, existing graph-based RAG baselines often underperform chunk-based RAG methods, as their reliance on binary relational graphs based on OWA causes knowledge loss during extraction, lack of boundary semantics, and granularity fragmentation.

Comparison across question levels. CWAGraph improves not only Easy-level decisions, but also Medium-level set recovery and Hard-level complete coverage. Its gain is largest on Medium-level F1, where the system must recover the in-scope entities covered by the LCS's resolved boundary.

Comparison across source domains. CWAGraph's advantage holds in the two domains, despite their different writing conventions and scope organization. It suggests that CWAGraph is not tied to one corpus format.Beyond these results, a more detailed case study of CWAGraph's remaining errors is provided in Appendix F.

Table 2: All compared methods' performance on SetQA.

Source Domain	Method	Task-Macro Score	Easy Acc.	Medium F1	Hard Acc.
Technical Manuals	Naive RAG	33.9	52.8	13.5	35.4
	Agentic RAG	37.0	59.3	14.7	36.9
	Long-Context	37.4	67.6	19.9	24.6
	LightRAG	37.5	54.6	19.5	38.5
	GraphRAG	22.6	25.0	16.7	26.2
	HopRAG	26.7	49.1	14.2	16.9
	HippoRAG	26.4	40.7	15.4	23.1
	CWAGraph	**79.6**	**92.6**	**69.4**	**76.9**
Food Safety Regulations	Naive RAG	19.9	33.3	14.9	11.4
	Agentic RAG	16.2	35.8	8.6	4.3
	Long-Context	32.3	59.3	17.7	20.0
	LightRAG	23.2	39.5	20.1	10.0
	GraphRAG	22.7	55.6	8.8	10.0
	HopRAG	14.0	25.9	7.4	8.6
	HippoRAG	16.4	29.6	9.7	10.0
	CWAGraph	**64.3**	**81.5**	**65.7**	**45.7**

5.3 Effectiveness of Evidence Retrieval (RQ2)

To answer RQ2, we compare all methods' performance on evidence (statement) retrieval and the answer accuracy based on retrieved evidence in Table 3.

Table 3: Easy-level retrieval and conditional accuracy.

Source Domain	Method	Hit@1	Hit@5	Easy Acc.	Easy Acc. | Hit@5
Technical Manuals	Naive RAG	65.7	87.0	52.8	57.4
	Agentic RAG	52.8	78.7	59.3	64.7
	Long-Context	29.5	73.9	67.6	76.9
	LightRAG	63.0	81.5	54.6	58.0
	GraphRAG	33.3	45.4	25.0	46.9
	HopRAG	35.5	66.8	49.1	47.2
	HippoRAG	66.7	82.4	40.7	43.8
	CWAGraph	**74.1**	**98.1**	**92.6**	**92.5**
Food Safety Regulations	Naive RAG	8.9	27.8	33.3	40.9
	Agentic RAG	6.2	21.0	35.8	41.2
	Long-Context	20.3	70.9	59.3	64.3
	LightRAG	8.6	38.3	39.5	71.0
	GraphRAG	34.6	45.7	55.6	**86.5**
	HopRAG	33.8	58.0	25.9	45.5
	HippoRAG	23.5	40.7	29.6	42.4
	CWAGraph	**77.8**	**96.3**	**81.5**	83.3

The results reveal that CWAGraph performs the best on evidence retrieval as it achieves the highest Hit@5 in both source domains, while several baselines retrieve the gold LCSs at moderate Hit@5 rates, especially Naive RAG, Long-Context, HopRAG, and HippoRAG on Technical Manuals. However, retrieving the raw statements is not sufficient, the methods must also resolve which entities the statements govern.

The results also show that evidence retrieval is not enough for satisfactory QA. On Technical Manuals, Naive RAG reaches 87.0 Hit@5, but only achieves 57.4 Easy-level Accuracy; HippoRAG reaches 82.4 Hit@5 but only 43.8 conditional accuracy. CWAGraph keeps both numbers high, indicating that the retrieved LCS has been resolved into entity-addressable complement and family evidence rather than merely placed in context. The same performance gap remains across domains: Long-Context achieves 70.9 Hit@5 yet only 64.3 conditional accuracy; HopRAG achieves 58.0 Hit@5 but 45.5 conditional accuracy. CWAGraph achieves 96.3 Hit@5 and 83.3 conditional accuracy. It implies the same conclusion across source domains that the QA failures on SetQA are not only attributed to evidence retrieval failures, but also to the failures of applying the retrieved LCSs to implicit in-scope entities.

5.4 Ablation Study (RQ3)

We focus on Technical Manuals to investigate the impacts of CWAGraph's key modules, where retrieval labels are richest and the structure can be diagnosed most clearly. Figure 4 shows the ablation study results of Technical Manuals. We also report the ablation study results of Food Safety Regulations in Appendix E.2.

Impact of peer. The results show that removing peer relations in the CWA layer lowers CWAGraph's Task-Macro Score from 79.6 to 73.0, showing that peer constraints provide an independent same-level comparison signal for validating hierarchy attachments during scope resolution.

Impact of LCS resolution. Removing LCS resolution causes a large Medium-level and Hard-level performance drop of CWAGraph, while Easy-level accuracy remains comparatively acceptable. It confirms that converting raw LCS records into entity-linked resolved complement objects is essential, beyond merely retrieving a relevant statement. We also test an LCS-only baseline in Appendix E, confirming that the gains do not come from simply packaging LCS for retrieval.

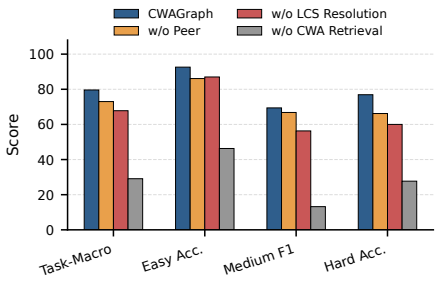

Figure 4: Ablation study results on Technical Manuals.

Impact of CWA layer retrieval. Removing CWA layer retrieval causes the largest performance drop of CWAGraph, from 79.6 to 29.1 Task-Macro Score. It shows that the CWA layer is the main resolution path of CWAGraph for SetQA task rather than auxiliary metadata. We also test CWAGraph's performance on general QA in Appendix E.5. Results indicate that CWA layer does not degrade the performance on general QA, compared to Naive RAG and LightRAG.

5.5 Oracle Analysis (RQ4)

To estimate the remaining headroom in automatic CWA layer construction, we conduct an oracle analysis on Technical Manuals. Specifically, we replace the extracted hierarchy-peer structure, the extracted complement records, or both of them with human verified data (denoted as Oracle H, P, Oracle C, and Oracle Full CWA, respectively), while keeping the same retrieval and answer pipeline in CWAGraph. Figure 5 displays the performance of these oracles, along with CWAGraph's performance (from Table 2). Concrete oracle scores, including those on Food Safety Regulations, are reported in Appendix E.1 and Appendix E.3.

Hierarchy-peer construction errors. Oracle H, P improves CWAGraph's Medium-level F1 from 69.4 to 87.3, showing that scope recovery remains a major source of error.

Complement construction errors. Oracle C provides a smaller gain, moving Task-Macro Score from 79.6 to 80.3, and Medium-level F1 from 69.4 to 71.5. It suggests that complement extraction and boundary normalization are not the only remaining bottleneck on Technical Manuals.

Full oracle CWA layer. Oracle Full CWA raises Task-Macro Score from 79.6 to 89.3. Since the answer pipeline and backbone LLM are unchanged in this analysis, the remaining errors involve retrieval and answer generation rather than CWA layer construction.

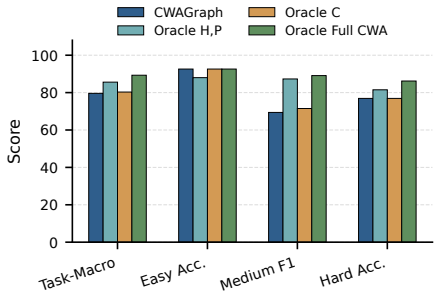

Figure 5: Oracle analysis on Technical Manuals.

6 Limitations and Future Work

Scope of completeness reasoning. The LCS frame in this paper covers single-statement local completeness. We do not handle distributed completeness, where closure is jointly determined by multiple statements, nor corpus-level closure, where the absence of any mention across the entire corpus is treated as evidence against a fact. Extending the resolution algorithm to handle either case is left to future work.

Domain coverage. We evaluate on two source domains, Technical Manuals and Food Safety Regulations. The prevalence study in Appendix A.4 also shows that LCS-style wording appears in an external tariff corpus. All benchmark documents are in English.

Public benchmark scale. LCS-rich corpora at scale are often proprietary, paywalled, or require domain-expert annotation, such as legal contracts, medical guidelines, and internal compliance standards. This constrains publicly releasable QA construction at this stage. SetQA is a feasible open evaluation under these constraints; scaling it further will require institutional access or expert annotators beyond this work's scope.

Offline cost. CWAGraph offline construction is more expensive than pure chunk-based RAG and some graph-based RAG, mainly because it adds complement parsing and hierarchy-peer scope resolution on top of standard graph extraction. Reducing this overhead, for example through batched resolution or a smaller specialized extractor, is left to future work.

7 Conclusion

In this paper, we introduced CWAGraph, a novel graph-based RAG framework for LCSs that standard OWA graphs cannot resolve. By introducing Hierarchy, Peer and Complement, our CWAGraph resolves each LCS's operative scope, making it addressable from in-scope entities, and then retrieving it through an entity-anchored CWA path. Our experiments on specially constructed SetQA demonstrate CWAGraph's consistent performance gains over chunk-based, long-context, and graph-based RAG baselines.

Acknowledgments and Disclosure of Funding

This work received no third-party funding. The authors declare no competing interests.

References

[1] Samuel Joseph Amouyal, Tomer Wolfson, Ohad Rubin, Ori Yoran, Jonathan Herzig, and Jonathan Berant. Qampari: An open-domain question answering benchmark for questions with many answers from multiple paragraphs. *arXiv preprint arXiv:2205.12665*, 2022.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[3] Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, J Scott McCarley, Michael McCawley, et al. The techqa dataset. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 1269–1278, 2020.

[4] Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. Pathrag: Pruning graph-based retrieval augmented generation with relational paths. In *Proceedings of the AAAI conference on artificial intelligence*, volume 40, pages 30183–30191, 2026.

[5] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL `https://arxiv.org/abs/1803.05457`.

[6] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 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, 2009.

[7] Fariz Darari, Werner Nutt, Giuseppe Pirro, and Simon Razniewski. Completeness statements about rdf data sources and their use for query answering. In *International Semantic Web Conference*, pages 66–83. Springer, 2013.

[8] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

[9] Oren Etzioni, Keith Golden, and Daniel S Weld. Sound and efficient closed-world reasoning for planning. *Artificial Intelligence*, 89(1-2):113–148, 1997.

[10] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*, pages 413–422, 2013.

[11] Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. Message passing for hyper-relational knowledge graphs. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 7346–7359, 2020.

[12] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, Haofen Wang, et al. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1):32, 2023.

[13] Zirui Guo, Lianghao Xia, Yanhua Yu, Tian Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2(3), 2024.

[14] Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024.

[15] Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. Cuad: An expert-annotated nlp dataset for legal contract review. *arXiv preprint arXiv:2103.06268*, 2021.

[16] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, 2020.

[17] Hongru Hou, Jiachen Sun, Wenqing Lin, Wendong Bi, Xiangrong Wang, and Deqing Yang. Heterogeneous influence maximization in user recommendation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 5747–5754, 2025.

[18] Hongru Hou, Tiehua Mei, Denghui Geng, Jinhui Huang, Ao Xu, Hengrui Chen, Jiaqing Liang, and Deqing Yang. Prorl: Effective reinforcement learning for proactive recommendation via rectified policy gradient estimation. *arXiv preprint arXiv:2605.28293*, 2026.

[19] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 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, 2020.

[20] Denis Krompaß, Stephan Baier, and Volker Tresp. Type-constrained representation learning in knowledge graphs. In *International semantic web conference*, pages 640–655. Springer, 2015.

[21] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[22] Lei Liang, Zhongpu Bo, Zhengke Gui, Zhongshu Zhu, Ling Zhong, Peilong Zhao, Mengshu Sun, Zhiqiang Zhang, Jun Zhou, Wenguang Chen, et al. Kag: Boosting llms in professional domains via knowledge augmented generation. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 334–343, 2025.

[23] Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. Hoprag: Multi-hop reasoning for logic-aware retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1897–1913, 2025.

[24] Haoran Luo, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Yifan Zhu, et al. Hypergraphrag: Retrieval-augmented generation via hypergraph-structured knowledge representation. *arXiv preprint arXiv:2503.21322*, 2025.

[25] Tiehua Mei, Minxuan Lv, Leiyu Pan, Zhenpeng Su, Hongru Hou, Hengrui Chen, Ao Xu, and Deqing Yang. Good reasoning makes good demonstrations: Implicit reasoning quality supervision via in-context reinforcement learning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 30452–30469, 2026.

[26] Wangyi Mei, Zhouhong Gu, Zhenhan Bai, Yin Cai, Lefan Zhang, Zhenxin Ding, Bo Chen, Yan Gao, Yi Wu, Yao Hu, Jiaqing Liang, and Deqing Yang. Deep research as rubric for reinforcement learning. *arXiv preprint arXiv:2606.01091*, 2026. doi: 10.48550/arXiv.2606.01091. URL https://arxiv.org/abs/2606.01091.

[27] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015.

[28] Simon Razniewski, Hiba Arnaout, Shrestha Ghosh, and Fabian Suchanek. Completeness, recall, and negation in open-world knowledge bases: A survey. *ACM Computing Surveys*, 56(6):1–42, 2024.

[29] Raymond Reiter. On closed world data bases. In *Readings in artificial intelligence*, pages 119–140. Elsevier, 1981.

[30] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009. ISSN 1554-0669. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

[31] Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. Beyond triplets: hyper-relational knowledge graph embedding for link prediction. In *Proceedings of the web conference 2020*, pages 1885–1896, 2020.

[32] Jinyu Wang, Jingjing Fu, Rui Wang, Lei Song, and Jiang Bian. Pike-rag: specialized knowledge and rationale augmented generation. *arXiv preprint arXiv:2501.11551*, 2025.

[33] Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. On the representation and embedding of knowledge bases beyond binary relations. *arXiv preprint arXiv:1604.08642*, 2016.

[34] Ganlin Xu, Zhoujia Zhang, Wangyi Mei, Jiaqing Liang, Weijia Lu, Xiaodong Zhang, Zhifei Yang, Xiaofeng Ma, Yanghua Xiao, and Deqing Yang. Logical consistency is vital: Neural-symbolic information retrieval for negative-constraint queries. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1828–1847, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.92. URL https://aclanthology.org/2025.findings-acl.92/.

[35] Ganlin Xu, Linghao Zhang, Zhitao Yin, Hongda Xi, Chen Yang, Jiaqing Liang, Weijia Lu, Sihang Jiang, Yanghua Xiao, and Deqing Yang. When rag meets query planning: Logical query trees for exploratory reasoning problems. *Proc. ACM Manag. Data*, 4(4), September 2026. doi: 10.1145/3837131. URL https://doi.org/10.1145/3837131.

[36] Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. Retrieval meets long context large language models, 2024. URL https://arxiv.org/abs/2310.03025.

[37] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380, 2018.

[38] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[39] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

[40] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems*, 19, 2006.

A SetQA Construction

SetQA is built from two source corpora: Huawei S-series network switch manuals (*Technical Manuals*) and GB 2760, China's national food additive standard (*Food Safety Regulations*). The LLM is used only to collect candidate LCS segments. Multiple expert annotators determine the final statement, universe scope, predicate, boundary, and QA labels. During construction, annotators do not see any CWAGraph or baseline output. All gold answers are fixed before any method is evaluated.

A.1 Pipeline

Statement extraction. An LLM reads each chunk and returns candidate text segments that may express an LCS. We use a high-recall prompt: any segment with closed-scope wording, such as "all", "only", "the following", "except", or "otherwise", is kept in its original form without rewriting or labeling.

Scope reconciliation. For each retained statement, multiple expert annotators independently build the universe scope from the corpus, then reconcile their drafts in two rounds before the scope is fixed.

Question instantiation. From each statement and its reconciled scope, we make three question levels. **Easy-level** questions pick an in-scope entity that is not explicitly mentioned in the source statement and ask whether the predicate applies to it. **Medium-level** questions ask which in-scope entities the predicate holds for. **Hard-level** questions ask how many in-scope entities the predicate holds for [34].

Human verification. Multiple annotators check each step independently. We keep consensus items and remove cases that remain ambiguous after review. For each question, both annotators verify that the wording matches the scope, that the gold answer follows from the statement and the reconciled scope, and that the question cannot be solved by simple word overlap. Annotators never see any CWAGraph or baseline output during construction.

After construction, the human-verified LCS pool for Technical Manuals contains 146 statements. Before question instantiation, we filter this pool by ground-truth availability (the complete answer set must be recoverable from the corpus), semantic deduplication, and entity diversity. In total, SetQA contains 493 QA pairs that reference 223 distinct verified LCSs.

A.2 Candidate Statement Extraction Prompt

This prompt is used only to collect candidate LCS segments. It does not determine the final scope, predicate, boundary, or answer labels. These are fixed during human verification.

Candidate Statement Extraction Prompt

Role. You are an extraction assistant for closed-corpus QA benchmark construction.

Task. Extract candidate local completeness statements from the document chunk.

Definition. A local completeness statement states a local scope and a boundary for whether a rule, property, permission, support relation, or prohibition applies. The statement may be one sentence, one bullet, or a short contiguous passage. If several adjacent sentences jointly define the same boundary, return the whole passage as one `raw_text` segment.

Keep segments with:

- closed-scope wording such as "only", "all", "every", "the following", or a complete list;
- boundary wording such as "other", "remaining", "otherwise", "all other", "except", or "except for";
- condition-complete applicability, such as "only when ...", "if and only if ...", or "no ... unless ...";
- nearby positive and negative cases when they jointly define one local boundary.

Skip:

- command definitions or feature definitions;
- default states;
- headings, parameter lists, or view/syntax lists;

- lifecycle, failure, or operational side-effect lists unless they define whether the queried rule or property itself applies;
- general requirements that do not close a local scope.

Instructions.

- Extract only original contiguous text segments from the chunk.
- Do not rewrite, normalize, infer, or explain.
- Prefer high recall: borderline candidates are acceptable because humans will verify them later.
- Output JSON only.

Input.

```
<document_chunk>
{chunk}
</document_chunk>
```

Output format.

```
[
  {"raw_text":  "..."}
]
```

A.3 Example

A schematic example is shown below.

- **Statement:** *"All noble gases are chemically inert under standard conditions, except xenon."*
- **Reconciled scope:** {helium, neon, argon, krypton, xenon, radon}
- **Easy question:** *Is argon chemically inert under standard conditions?* (gold: yes)
- **Medium question:** *Which noble gases are chemically inert under standard conditions?* (gold: {helium, neon, argon, krypton, radon})
- **Hard question:** *How many noble gases are chemically inert under standard conditions?* (gold: 5)

Actual SetQA items follow the same procedure on the two source corpora.

A.4 LCS Prevalence in Source Corpora

To characterize how often LCS-style statements appear in source material, we count local contexts that contain at least one candidate boundary or closed-scope expression. The evaluated corpora use the same benchmark source chunks as SetQA. We also include China tariff rules as a supplementary external corpus, segmented into text windows, to check whether similar patterns appear outside the two SetQA domains.

Table 4: LCS candidate prevalence in the evaluated source corpora and external reference domains.

Domain	Source	Chunk unit	Chunks	Cand.	%
Technical manuals	Huawei S-series manuals	Benchmark chunk	1003	621	61.91
Food regulations	GB 2760 standard	Benchmark chunk	372	146	39.25
Tariff rules	China tariff rules	3000-char fixed	563	330	58.61

LCS candidates appear in a substantial fraction of both evaluated corpora and in the external tariff reference domain. This supports treating LCS resolution as a recurring corpus phenomenon rather than a pattern specific to the two SetQA domains.

B Proofs

B.1 Proof of Proposition 1

We restate Proposition 1 for convenience.

Proposition 1. *Let $\mathcal{N}_k(e; \mathcal{G})$ be the k-hop neighborhood of an entity e in an OWA graph $\mathcal{G}$. For any finite k, σ cannot be resolved for R_σ from $\mathcal{N}_k(e; \mathcal{G})$ alone.*

Proof. This proof concerns the representation limit of an OWA graph, not the quality of the extractor. We allow the graph to contain all positive source-supported facts that help the query entity, including universe membership and all explicitly named boundary members. The only information not available in an OWA graph is a closure certificate stating that the named boundary side is complete.

Consider an in-scope entity $e \in U_\sigma$ whose membership in R_σ is not explicitly stored in $\mathcal{G}$. Even with all positive source-supported facts available, the extracted graph has no closure certificate that fixes whether e belongs to B_σ or to R_σ.

Under OWA, a missing edge is unknown rather than false. Therefore, the same $\mathcal{N}_k(e; \mathcal{G})$ is consistent with two completions: one where $e \in B_\sigma$, and one where $e \in R_\sigma$. Since both completions agree with all positive edges stored in $\mathcal{G}$, no procedure operating only on $\mathcal{N}_k(e; \mathcal{G})$ can distinguish them. Hence σ cannot be resolved for such entities from finite k-hop OWA retrieval alone. □

Remark. Standard KG extraction outputs only facts stated explicitly in the source. Every output triple must trace back to a specific text segment, so anything requiring rule-based implication is left to downstream reasoning rather than handled at extraction time. This holds for both LLM-based and rule-based pipelines - any implied triple, including those an LCS would induce, is not produced during extraction.

B.2 Proof of Proposition 2

We restate Proposition 2 for convenience.

Proposition 2. *Under the representation setting in Definition 3, hierarchy and complement are each necessary for resolving local completeness statements.*

Proof. We show necessity by removing each component and constructing, in each case, two worlds that are consistent with the same represented structure yet yield different LCS resolutions.

Without **Hierarchy**, complement records may state a predicate-bearing boundary, but the representation has no vertical membership relation that determines whether a concrete entity belongs to the universe scope. Consider an entity e and a scope class U_σ. In one world, e is a member of U_σ and belongs to R_σ; in another world, e is outside U_σ entirely. Without hierarchy, no vertical membership edge connects e to U_σ, so both worlds produce the same represented structure. Since the induced split differs, the LCS cannot be resolved.

Without **Complement**, the represented structure contains only entity organization. Hierarchy can state vertical membership, but it records neither the predicate nor the local completeness declaration for this statement. For any unnamed entity e in the ground-truth $U_\sigma \setminus B_\sigma$, neither positive nor negative predicate status is stored in the representation. By the same indistinguishability argument as Proposition 1, the LCS cannot be resolved. □

C Algorithm

This appendix provides the step-by-step pseudocode referenced in Section 4. We separate offline construction from online retrieval.

C.1 Symbol and Operator Definitions

The symbols and auxiliary operators used in Algorithms 1 and 2 are defined below. They correspond to steps that the main text names at a high level but does not expand line by line.

Grounding. For a textual reference x in a complement record, $\textsc{Ground}_H(x)$ returns the hierarchy nodes that x refers to:

$$\textsc{Ground}_H(x) = \mathrm{Resolve}(\mathrm{Cand}_H(x), x),$$

where $\mathrm{Cand}_H(x)$ collects hierarchy nodes whose name or aliases match the normalized x. A singleton candidate is returned directly; otherwise the grounding-disambiguation prompt in Appendix G.2

selects those that are aliases or co-references of x. If no candidate matches, the complement record is dropped.

Peer-guided pruning. $\Gamma_P(\widehat{U}; H)$ returns a local hierarchy H' used to resolve a single LCS:

$$H' = H \setminus \Delta_P(\widehat{U}; H),$$

where $\Delta_P(\widehat{U}; H)$ collects hierarchy edges inside $\mathrm{Desc}^*_H(\widehat{U})$ under which two peer-linked entities attach to the same universe at incompatible depths.

Fuzzy entity matching. At retrieval time, $\textsc{FuzzyMatch}(V_q, E)$ maps query strings V_q to entity identifiers in E using the normalization and alias rules from $\textsc{Ground}$, with a fallback to lexical and embedding similarity over entity names and descriptions. Strings with no match are dropped from the entity-anchored path but can still match evidence through the semantic path.

RRF with entity-anchor bonus. Standard reciprocal-rank fusion is

$$\mathrm{RRF}(x) = \sum_{s \in S(x)} \frac{1}{k_0 + \mathrm{rank}_s(x)},$$

with $S(x)$ the retrieval signals returning x and k_0 the smoothing constant. For complement records we add an entity-anchor indicator:

$$\mathrm{RRF}^+(x) = \mathrm{RRF}(x) + \lambda \cdot \mathbf{1}[x \in C^*_{\mathrm{ent}}(q)],$$

where $C^*_{\mathrm{ent}}(q)$ are complements reached through the entity-addressing index $A_{C \leftrightarrow E}$. Entities and OWA triples are scored by plain RRF.

C.2 CWAGraph Construction

Algorithm 1 CWAGraph Construction

Require: Corpus $\mathcal{K}$, extraction model $\mathcal{M}_{\text{ext}}$
Ensure: CWAGraph $\mathcal{G}_{\text{CWA}} = (E, R, H, P, C^*)$, entity-addressing index $A_{C \leftrightarrow E}$

$E, R, H_{\text{raw}}, P_{\text{raw}}, C_{\text{raw}}, C^*, A_{C \leftrightarrow E} \leftarrow \emptyset$
for each chunk $d \in \mathcal{K}$ **do**
  $(E_d, R_d, H_d, P_d, C_d^{\text{raw}}) \leftarrow \mathcal{M}_{\text{ext}}(d)$
  $E \cup= E_d;\ R \cup= R_d;\ H_{\text{raw}} \cup= H_d;\ P_{\text{raw}} \cup= P_d;\ C_{\text{raw}} \cup= C_d^{\text{raw}}$
end for
$H \leftarrow H_{\text{raw}};\ P \leftarrow P_{\text{raw}}$
for each retained $c = (u_{\text{all}}, b^{\text{ref}}, \phi_c, s_c) \in C_{\text{raw}}$ **do**
  $\widehat{U}_{\text{all}} \leftarrow \textsc{Ground}_H(u_{\text{all}});\ H' \leftarrow \Gamma_P(\widehat{U}_{\text{all}}; H)$
  $U_{\text{all}} \leftarrow \textsc{Desc}^*_{H'}(\widehat{U}_{\text{all}})$
  $\widehat{B}_c \leftarrow \textsc{Ground}_{H'}(b^{\text{ref}})$
  $B_c \leftarrow \textsc{Desc}^*_{H'}(\widehat{B}_c) \cap U_{\text{all}};\ R_c \leftarrow U_{\text{all}} \setminus B_c$
  $c^* \leftarrow (U_{\text{all}}, B_c, R_c, \phi_c, s_c)$
  $C^* \cup= \{c^*\}$
  for each $e \in U_{\text{all}}$ **do**
    $A_{C \leftrightarrow E} \cup= \{(c^*, e)\}$
  end for
end for

C.3 CWAGraph Retrieval Strategy

Algorithm 2 CWAGraph Retrieval Strategy

Require: Query q, CWAGraph $\mathcal{G}_{\text{CWA}}$, entity-addressing index $A_{C \leftrightarrow E}$
Ensure: Ranked evidence $\mathcal{R}(q)$, family context $\text{Fam}_{H,P}(q)$

$V_q \leftarrow \textsc{ParseEntities}(q)$; $\widetilde{E}_q \leftarrow \textsc{FuzzyMatch}(V_q, E)$
$C^*_{\text{ent}}(q) \leftarrow \{\, c \in C^* \mid \exists e \in \widetilde{E}_q : (c, e) \in A_{C \leftrightarrow E} \,\}$
$\text{Fam}_{H,P}(q) \leftarrow \textsc{FamilyContext}(\widetilde{E}_q, C^*_{\text{ent}}(q), H, P)$
$C^*_{\text{sem}}(q) \leftarrow \textsc{TopKComplement}(q, C^*)$; $E_{\text{sem}}(q) \leftarrow \textsc{TopKEntity}_{\text{CWA}}(q)$
$P_{\text{com}}(q) \leftarrow \{\, \tau(c) \mid c \in C^*_{\text{ent}}(q) \cup C^*_{\text{sem}}(q) \,\}$
$P_{\text{ent}}(q) \leftarrow \widetilde{E}_q \cup E_{\text{sem}}(q)$
$E_{\text{OWA}}(q), R_{\text{OWA}}(q) \leftarrow \textsc{RetrieveOWA}(q, \widetilde{E}_q, E, R)$
$P_{\text{ent}}(q) \leftarrow P_{\text{ent}}(q) \cup E_{\text{OWA}}(q)$; $P_{\text{rel}}(q) \leftarrow R_{\text{OWA}}(q)$
$\mathcal{R}(q) \leftarrow \textsc{Merge}\{\text{RRF}^+(P_{\text{com}}(q)), \text{RRF}(P_{\text{ent}}(q)), \text{RRF}(P_{\text{rel}}(q))\}$

D Experimental Details

D.1 Baselines

The baseline configurations are summarized below. Unless stated otherwise, we keep each baseline's implementation defaults and only set the parameters needed to align each baseline to the same source documents, embedding model, LLM backbone, answer protocol, and context budget [35].

Table 5: Construction settings and retrieval units.

Method	Retrieval unit	Construction setting	Budget
Naive RAG	Dense text chunks	512 / 64 on Technical Manuals; Food Safety Regulations uses pre-chunked JSONL	8192
Agentic RAG	Dense chunks + query rewriting	Reuses the Naive RAG index	8192
Long-Context	Whole documents	No retrieval-time chunking	32768
LightRAG	KG + text chunks	1200 / 100; entity gleaning = 1	8192
GraphRAG	Local graph search	1200 / 100; community max cluster size 10; max gleanings 0	8192
HopRAG	Graph traversal entries	Implementation defaults unless specified	8192
HippoRAG	Passage graph retrieval	1200 / 100 on Technical Manuals; Food uses pre-chunked JSONL; graph type `facts_and_sim_passage_node_unidirectional`	8192
CWAGraph (ours)	Entity + relation + CWA constraints	1200 / 100; joint OWA/CWA extraction; hierarchy-peer-complement resolution	8192

Table 6: Retrieval settings.

Method	Retrieval setting
Naive RAG	Cosine top-200 over `text-embedding-3-large`; greedy budget packing
Agentic RAG	Up to 3 turns; LLM decomposes into 1–3 subqueries and checks sufficiency; retrieve top-200 per subquery, deduplicate, greedy pack
Long-Context	BM25 over whole documents with `jieba`; greedy pack by BM25 rank
LightRAG	`mix` mode; `top_k=60`; `chunk_top_k=20`; no external reranker; `only_need_context=True`
GraphRAG	Local search; community level 2; context budget 8192
HopRAG	`max_hop=3`; `topk=7`; `initial_topk=5`
HippoRAG	Retrieve top-200; linking top-5; max QA steps 3; no DSPy reranker; greedy pack ranked passages
CWAGraph (ours)	Dual-path CWA + OWA retrieval; BM25/cosine/RRF fusion; context budget 8192

All methods use GPT-5.2 for graph construction, traversal, or query processing when the method calls an LLM, and all embedding-based methods use `text-embedding-3-large`. Final answers are generated by the shared answer-generation script under the same prompt, context budget, sampling temperature 0, and 2048 output-token cap.

For LightRAG and GraphRAG, the Technical Manuals entity types are {`model`, `feature`, `version`, `protocol`, `concept`, `other`} and the Food Safety Regulations entity types are {`food_category`, `additive`, `concept`, `other`}. These type vocabularies adapt the released pipelines to the two domains but do not encode benchmark answers.

For HopRAG, `initial_topk` is set to 5 and `topk` to 7 to keep the traversal pool comparable to the other graph baselines under the same 8192-token context budget. The released/default-style setting

uses a much wider entry pool (e.g., top-20); under multi-hop traversal, this expands the candidate nodes before final packing and can expose the reader to far more chunks than the other baselines. We therefore use the recorded init5/top7 setting while keeping the remaining HopRAG logic unchanged.

D.2 Evaluation Settings

All methods use the answer-generation prompt and the set-matching prompt shown later in this appendix. Sampling temperature is 0 and the output cap is 2048 tokens. Easy-level Accuracy parses the model's yes/no decision; Hard-level Accuracy is exact integer match; Medium-level F1 is computed by the set-matching prompt against the gold set.

Hit@K uses a single LLM-judge prompt for all methods, because GraphRAG, LightRAG, and HippoRAG reorganize source content into summaries that direct string match cannot reliably detect. The judge sees the gold statement and a top-K retrieved unit and decides whether the unit semantically supports the gold statement.

Table 7: Hit@K matching rules for baseline retrieval analysis.

Method	Hit@K rule
Naive RAG	LLM-supported semantic match between the gold statement and top-K retrieved chunks.
Agentic RAG	LLM-supported semantic match between the gold statement and top-K retrieved chunks after query expansion.
Long-Context	Match the gold source chunk to its parent document and check whether a supporting parent document appears in the top-K BM25-ranked documents.
LightRAG	LLM-supported semantic match between the gold statement and the retrieved LightRAG context units.
GraphRAG	LLM-supported semantic match between the gold statement and ranked local-search source rows, with majority voting on unstable judgments.
HopRAG	Use the ranked retrieval output under `max_hop=3`, `topk=7`, and `initial_topk=5`.
HippoRAG	LLM-supported semantic match between the gold statement and the top-K ranked passages returned by HippoRAG.

D.3 Human Validation of LLM Judge

To verify the reliability of the LLM-based set-matching judge used for Medium-level F1, we randomly sample 200 item-level matching decisions from Medium-level evaluations, 100 from each source domain. Multiple human annotators independently inspect the TP/FP/FN matching decisions and identify LLM-judge errors [26].

The manual review is run in two rounds. Round 1 covers all 200 item-level decisions and identifies 9 LLM-judge errors (4.50%). Round 2 also covers all 200 decisions and identifies 11 errors (5.50%).

The two rounds are highly consistent on the binary error labels: they agree on 198 of 200 decisions (99.00%), with Cohen's κ of 0.895. The observed error rate is about 5%, supporting the reliability of the Medium-level F1 judge used in the main results.

E Additional Results

E.1 Exact Analysis Tables on Technical Manuals

The following tables provide the exact values visualized in the ablation and oracle analysis figures. To isolate whether CWAGraph's gains come from the CWA graph structure or from simply preserving LCS text in a retrievable form, we construct a text-only LCS baseline called **LCS Only**. This variant stores the same self-contained LCS records extracted during CWA constraint joint extraction and retrieves them via cosine similarity, but does not perform scope resolution, boundary resolution, or entity-addressing indexing. No hierarchy, peer, or resolved complement structure is used.

Table 8: Exact ablation results on Technical Manuals.

Variant	Task-Macro Score	Easy Acc.	Medium F1	Hard Acc.
CWAGraph	79.6	92.6	69.4	76.9
w/o Peer	73.0	86.1	66.8	66.2
w/o LCS Resolution	67.8	87.0	56.3	60.0
LCS Only	34.2	72.2	18.1	12.3
w/o CWA Retrieval	29.1	46.3	13.2	27.7

Table 9: Exact oracle results on Technical Manuals.

Variant	Task-Macro Score	Easy Acc.	Medium F1	Hard Acc.
CWAGraph	79.6	92.6	69.4	76.9
Oracle H, P	85.6	88.0	87.3	81.5
Oracle C	80.3	92.6	71.5	76.9
Oracle Full CWA	**89.3**	**92.6**	**89.1**	**86.2**

E.2 Additional Component Analysis on Food Safety Regulations

We additionally report Food Safety Regulations component ablations under the final pipeline. All ablation variants score below the full CWAGraph result. The **LCS Only** variant follows the same text-only setup described above.

Table 10: Additional Food Safety Regulations component ablations under the final pipeline.

Variant	Task-Macro Score	Easy Acc.	Medium F1	Hard Acc.
CWAGraph	64.3	81.5	65.7	45.7
w/o Peer	60.3	80.2	62.2	38.6
w/o LCS Resolution	41.6	59.3	44.0	21.4
LCS Only	22.7	44.4	10.9	12.9
w/o CWA Retrieval	16.5	27.2	9.5	12.9

E.3 Food Safety Regulations Oracle CWA

As a supplementary diagnostic upper bound, we also evaluate Food Safety Regulations oracle settings that replace the extracted H, P layer, the extracted C records, or both.

Table 11: Supplementary oracle CWA results on Food Safety Regulations. Oracle H, P replaces the extracted hierarchy-peer layer, Oracle C replaces the extracted complement records, and Oracle Full CWA replaces both, while keeping the rest of the pipeline unchanged.

Variant	Task-Macro Score	Easy Acc.	Medium F1	Hard Acc.
CWAGraph	64.3	81.5	65.7	45.7
Oracle H, P	73.8	85.2	86.1	50.0
Oracle C	67.7	81.5	70.1	51.4
Oracle Full CWA	**75.6**	**85.2**	**87.3**	**54.3**

E.4 Retrieval Diagnostics at Medium and Hard Levels

The main text reports Easy-level Hit@K. The table below complements that analysis with retrieval hit rates for Medium and Hard questions, using the same semantic Hit@K judge and the same retrieved evidence units as the main retrieval diagnostic.

Table 12: Retrieval diagnostics at Medium and Hard levels.

Source Domain	Method	Med. Hit@1	Med. Hit@5	Hard Hit@1	Hard Hit@5
Technical Manuals	Naive RAG	30.3	42.7	15.4	35.4
	Agentic RAG	30.3	42.7	15.4	35.4
	Long-Context	69.7	91.0	58.2	90.9
	LightRAG	15.7	49.4	10.8	44.6
	GraphRAG	12.4	33.7	12.3	26.2
	HopRAG	13.5	50.6	15.4	35.4
	HippoRAG	39.3	60.7	30.8	55.4
Food Safety Regulations	Naive RAG	15.2	57.0	42.9	72.9
	Agentic RAG	13.9	57.0	42.9	72.9
	Long-Context	30.4	82.3	37.1	82.9
	LightRAG	16.5	65.8	37.1	80.0
	GraphRAG	28.7	43.8	21.4	42.9
	HopRAG	30.0	58.8	57.1	78.6
	HippoRAG	35.0	67.5	52.9	81.4

E.5 Shared-Corpus Comparison on Multi-hop QA

To test whether CWAGraph remains competitive outside LCS-style questions, we compare it with chunk-based, long-context, and graph-based baselines on two multi-hop QA benchmarks under a shared-corpus protocol. For HotpotQA [37], we uniformly sample 200 questions from the distractor dev set and build one corpus from the deduplicated union of their context passages (1,995 passages). For 2WikiMultiHopQA [16], we sample 200 questions in the same way (1,478 passages). We use `deepseek-v4-flash` as backbone; embeddings use `text-embedding-3-large`. We report LLM-judge accuracy together with the standard EM and F1.

The context budget is 4k tokens for all methods except Long-Context, which uses 16k tokens, keeping the 4:1 ratio of the main experiments. Open-domain passages rarely state LCSs: on the HotpotQA corpus, CWAGraph extracts 121 complement candidates, of which 86 pass verification. CWAGraph therefore gives the CWA block a less token budget at most 400 tokens.

Table 13: Shared-corpus comparison on 200 HotpotQA and 200 2WikiMultiHopQA questions. Acc. is measured by an LLM judge; EM and F1 follow the standard definitions. Long-Context uses a 16k-token budget; all other methods use 4k tokens.

	HotpotQA			2WikiMultiHopQA		
Method	Acc.	EM	F1	Acc.	EM	F1
Naive RAG	84.5	62.5	77.5	52.5	43.5	49.6
Long-Context	80.0	59.0	72.1	45.0	38.0	41.8
LightRAG	79.5	60.0	73.8	47.5	39.0	43.9
HippoRAG	79.0	59.5	72.9	62.0	53.0	57.8
CWAGraph	**85.5**	**65.0**	**78.6**	**64.5**	**54.5**	**59.6**

On HotpotQA and 2WikiMultiHopQA, CWAGraph obtains the highest accuracy, EM, and F1. CWAGraph thus remains competitive with the strongest baselines on standard multi-hop QA.

E.6 Backbone and Judge Robustness

All main experiments use GPT-5.2 for construction, generation, and judging. To check that the gains do not depend on this model, we rerun CWAGraph on SetQA Technical Manuals with two models from other families: Deepseek-v4-flash and Claude Sonnet 4.6, keeping prompts and pipeline code unchanged. Easy and Hard answers are scored by exact matching; Medium Set-F1 is scored by LLM.

Replacing GPT-5.2 with a different-family model does not lower the Task-Macro Score (84.5 vs. 77.8). The weaker `deepseek-v4-flash` loses 5.4 points (72.4 vs. 77.8) and stays 34.9 points above LightRAG, the strongest baseline on Technical Manuals.

Table 14: CWAGraph with different backbones

Backbone	Task-Macro Score	Easy Acc.	Medium F1	Hard Acc.
CWAGraph (GPT-5.2)	77.8	92.6	64.0	76.9
CWAGraph (Claude Sonnet 4.6)	84.5	99.1	85.2	69.2
CWAGraph (`deepseek-v4-flash`)	72.4	97.2	61.5	58.5
LightRAG (GPT-5.2)	37.5	54.6	19.5	38.5

F Case Study

We inspect representative CWAGraph errors to identify which step failed. The answer context is exactly what the generator receives. The graph or scope box is used only for analysis and is not included in the generation prompt.

Across both source domains, CWAGraph produces roughly 170 predictions with non-perfect scores. Through manual inspection of 100 sampled errors from this pool, we find that most samples fall into four recurring error types: missing hierarchy children, missing entity-to-LCS links, near-miss retrieval, and answer composition errors when the context already contains the needed evidence.

The four cases below give one representative example for each type.

F.1 Case 1: Incomplete Hierarchy for Scope Closure

Recurring trigger. This error appears when the complement is retrieved correctly, but descendant closure is computed over an incomplete hierarchy. In technical manuals, hierarchy is often carried by section headings, table continuation, or cross references. When the parent context lies outside the chunk or is only implicit in the document, chunk-local hierarchy-peer extraction can miss leaf entities even when the LCS is correctly extracted.

Question. In the `S5732-H` series, which product models support both VXLAN and wireless IPv4 multicast access?

Gold. {`S5732-H24S6Q`, `S5732-H24UM2C`, `S5732-H24UM2CC`, `S5732-H48S6Q`, `S5732-H48UM2C`, `S5732-H48UM2CC`, `S5732-H48XUM2CC`}

Prediction. {`S5732-H24S6Q`, `S5732-H24UM2CC`, `S5732-H48S6Q`, `S5732-H48UM2CC`, `S5732-H48XUM2CC`}

Answer context

```
<Constraint>
  <Hierarchy+Peer>
  ...
  <Complement>
  Only the following models support wireless IPv4 multicast access:
  ..., S5732-H, S5732-H-K, ...
  Only the following models support VXLAN:
  ..., S5732-H, S5732-H-K, ...
</Constraint>
<OWA>
  ...
</OWA>
<Chunk>
  ...
</Chunk>
```

Graph / scope explanation

```
S5732-H
  |- S5732-H24S6Q
  |- S5732-H24UM2C [missing]
  |- S5732-H24UM2CC
```

```
  |- S5732-H48S6Q
  |- S5732-H48UM2C [missing]
  |- S5732-H48UM2CC
  |- S5732-H48XUM2CC
```

The retrieved complement records are sufficient: both relevant LCS records name `S5732-H` as an in-scope family. The error occurs when this family is unfolded through the extracted hierarchy. Because two descendants are missing from the closure, the resolved scope misses two models before generation starts. The case illustrates a construction-side limitation rather than a retrieval failure.

F.2 Case 2: Missing Entity Addressing for a Resolved LCS

Recurring trigger. This error appears when a resolved LCS is stored under one entity name, but the question is anchored at a more specific entity under a nearby category. Here, the LCS is linked to `cheese` under `dairy and dairy products`, while the query entity `whey-protein cheese` is under `cheese / processed-cheese products and analogues`. The two branches are semantically close, but they are not connected in $A_{C \leftrightarrow E}$. As a result, entity-anchored retrieval cannot reach the LCS from the query entity.

Question. Is whey-protein cheese allowed to use propylene glycol alginate?

Gold. Yes.

Prediction. Unknown.

Answer context

<Constraint>
semantic hit: fat-emulsion products may use propylene glycol alginate.
Resolved complement: dairy and dairy products may use propylene glycol alginate, except pasteurized milk, sterilized milk, fermented milk, milk powder / cream powder, and cream.
</Constraint>

<OWA>
...
</OWA>

<Chunk>
...
</Chunk>

Complement state

```
resolved complement c*:
  scope = dairy and dairy products
  predicate = propylene glycol alginate is allowed
  exceptions = milk / fermented milk / milk powder / cream
scope tree used by c*:
  dairy and dairy products
  |- cheese
  |- milk
  |- fermented milk
  |- milk powder / cream powder
  |- cream
stored entity key for covered cheese products = cheese
```

Query-anchor state

```
query anchor e = whey-protein cheese
query-side category path:
  cheese / processed-cheese products and analogues
  |- whey-protein cheese
```

```
alias comparison:
  stored key from c*:  cheese
  query key:  whey-protein cheese
  semantic relation:  query key is a finer cheese item
observed index:  cheese -> c*, but whey-protein cheese -> no c* [missing]
missing A[C,E]: (c*, whey-protein cheese) [missing]
```

The answer should be yes: whey-protein cheese is a cheese item within the dairy scope and is not one of the excluded categories. The failure is in $A_{C\leftrightarrow E}$. The CWA record is attached to `cheese`, but the query starts from `whey-protein cheese`, whose parent is a similar cheese category rather than the same `cheese` node. Because this granularity mismatch is not bridged, $A_{C\leftrightarrow E}$ contains `cheese -> c*` but not `whey-protein cheese -> c*`.

F.3 Case 3: Scope–Predicate Retrieval Miss

Recurring trigger. This error appears when retrieval returns evidence that matches only part of the question. In dense food-additive tables, many rows share the same additive, and many rows share the same food category. The retrieved records may contain phosphate rules for other foods and cheese-analogue rules for other additives, while missing the record that combines cheese analogue with phosphates.

Question. Is cheese analogue allowed to use phosphates?

Gold. Yes.

Prediction. Unknown.

Answer context

<Constraint>
Retrieved top evidence:
CWA#1 wheat-flour phosphate rule
CWA#2 miscellaneous-grain phosphate rule
CWA#3 oil-product phosphate rule
CWA#4 cheese-analogue rule for **KCl**, not phosphates
CWA#5 frozen-drink phosphate rule
Missing from top evidence: dairy-scope phosphate rule for cheese analogue.
</Constraint>

<OWA>
...
</OWA>

<Chunk>
...
</Chunk>

Retrieved CWA state

```
cheese analogue
  |- retrieved:  KCl rule
  |- missing:  phosphate rule
```

The retrieved evidence is close but not enough. Some records are about phosphates, and one is about cheese analogue, but no retrieved record says that phosphates are allowed for cheese analogue. The missing row is the only one needed to answer yes, so the generator returns unknown.

F.4 Case 4: Answer Composition Failure

Recurring trigger. This error appears even when the context contains all needed evidence. The model must combine two retrieved facts: the device supports DHCP Snooping, and it also supports NAC. Since the question asks whether the device supports DHCP Snooping and does not support NAC, the correct answer is no.

Question. Does `S5720-28X-PWR-LI-ACF` satisfy: supports DHCP Snooping and does not support NAC?

Gold. No.

Prediction. Unknown.

Answer context

<Constraint>
CWA#1: all relevant `S300/S500/S2700/S5700/S6700` **models support DHCP Snooping, except** `S5751-L, S5731-L, S5731S-L.`
CWA#4: all relevant `S300/S500/S2700/S5700/S6700` **models support NAC traditional mode, except** `S5751-L, S5731-L, S5731S-L.`
</Constraint>
<OWA>
...
</OWA>
<Chunk>
...
</Chunk>

CWA layer state

```
S5720-28X-PWR-LI-ACF
  |- supports:  DHCP Snooping
  |- supports:  NAC traditional mode
```

Here the CWA evidence is complete. The queried model is covered by the DHCP Snooping record and also by the NAC record. Since the question asks whether the model supports DHCP Snooping and does not support NAC, the correct answer is false. The remaining failure is therefore in answer composition, not in CWA layer construction or retrieval.

F.5 Summary

These examples show four main failure types: missing children in hierarchy closure, missing entity-to-LCS links, near-miss retrieval, and answer composition errors after correct retrieval. In many cases, the LCS text is already present. The harder parts are linking it to the right scope, making it reachable from covered entities, retrieving the exact record, and using the retrieved facts correctly.

G Prompts

This section reports the prompt templates that correspond to the method and evaluation protocol described in the paper. Recent work has studied the role of reasoning quality in in-context demonstrations [25]. The few-shot examples used at extraction time are unrelated to either source corpus, therefore not leaking benchmark content.

G.1 Graph Extraction Prompt

Graph Extraction Prompt

Role. You are a knowledge graph extractor.

Task. Extract entities, relations, and constraints from the input text.

Definitions.

- **Entity**: an exact name from the original text, with a short description and an entity type from the domain vocabulary.
- **Relation**: a direct and explicit positive binary fact between two entities.
- **Hierarchy constraint**: a subset relation written as `A,in,B`, where `A` is the child and `B` is the parent.

- **Peer constraint**: a same-granularity relation written as `A,peer,B`, where `A` and `B` are sibling-level entities under a shared parent or compared against each other.
- **Complement constraint**: a predicate-bearing local partition with fields `predicate`, `universe_scope`, `boundary_scope`, and `self_contained_LCS`.

Entity rules.

- Preserve original entity names; do not use pronouns or referring expressions as entity names.
- Prefer definitions from the original text for `entity_description`.
- If the text gives no definition, use only a minimal contextual description or type label.
- Do not introduce external knowledge or restate hierarchy relations in entity descriptions.

Relation rules.

- Extract only direct, explicit binary facts.
- Decompose n-ary statements into binary relations.
- Do not use relation records to express membership; use hierarchy instead.

Constraint rules.

- Write hierarchy edges in child-to-parent direction, even when the source text is parent-to-child.
- Extract peer from coordination, comparison, sibling-list, or same-category wording that indicates same granularity; do not extract peer from mere co-occurrence.
- Extract complement only when the text defines whether a predicate applies within a local scope; identify the predicate, full universe scope, and explicitly named boundary scope.
- Positive inclusion lists are hierarchy or relation evidence, not complement evidence, unless they explicitly close an applicability boundary.
- Make the `self_contained_LCS` standalone. Replace pronouns and local references with concrete nouns from the chunk context; if this is impossible, do not extract the constraint.

Output schema. Fields in each record are separated by `<|#|>`.

- `entity`: `entity_name`; `entity_description`; `entity_type`.
- `relation`: `source_entity`; `target_entity`; `relationship_keywords`; `relationship_description`.
- `constraint / hierarchy`: `A,in,B`; `description`; `constraint_confidence`; `type_A`; `type_B`; `type_confidence`.
- `constraint / peer`: `A,peer,B`; `description`; `constraint_confidence`; `type_A`; `type_B`; `type_confidence`.
- `constraint / complement`: `predicate`; `universe_scope`; `boundary_scope`; `self_contained_LCS`; `constraint_confidence`.

Runtime fields. Domain configuration: {domain_config}; examples: {few_shot}; input segment: {segment_text}. After finishing, output `<|COMPLETE|>`.

G.2 Grounding Disambiguation Prompt

Grounding Disambiguation Prompt

Role. You resolve ambiguous entity names against the current hierarchy.

Task. For each ambiguous name mentioned by a constraint, select all hierarchy candidates that are aliases or co-referential expressions of that name.

Rules.

- Zero or more candidates may be selected.
- Do not select a candidate merely because it is related, nearby, a parent, or a child.
- If no candidate is an alias or co-reference, return `null` for that name.

Input.

```
Constraint: {desc}
Already resolved entities: {unambiguous}
Ambiguous names and candidates: {ambiguous}
```

Output. Return JSON only: `{"name1": ["cand_a", "cand_b"], "name2": null}`.

G.3 Peer-Guided Pruning Prompt

Peer-Guided Pruning Prompt

Role. You perform peer-guided pruning for local hierarchy resolution.

Task. We extracted two kinds of relations from the documents: membership `A,in,B`, where `A` is a sub-member or subclass of `B`, and same-level peer `A,peer,B`, where `A` and `B` are peers under the same parent. When a peer relation contradicts the hierarchy, decide which involved relations should be deleted so that the local hierarchy remains consistent.

Conflict input.

```
A: {a}
hierarchy path: {a_path}
direct parent: {a_parent}
B: {b}
hierarchy path: {b_path}
direct parent: {b_parent}
Triggered peer relation:
{a},peer,{b} (occurrence={occ}) description: {peer_desc}
All membership relations involving A or B, numbered H1..Hn:
{related_h}
All peer relations involving A or B, numbered P1..Pn:
{related_p}
```

Output. Return strict JSON only:

```
{
"delete_hierarchy": ["<membership triple verbatim>", ...],
"delete_peer": ["<peer triple verbatim>", ...],
"reason": "<one sentence>"
}
```

Triples are written as `a,in,b` or `a,peer,b`, without occurrence count or description. Return two empty arrays if nothing should be deleted.

G.4 Query Entity Parsing Prompt

Query Entity Parsing Prompt

Role. You identify concrete entity names used to anchor retrieval.

Task. Extract the concrete entity names involved in the question.

Rules.

- Return only entity names that literally appear in the question.
- Do not return question words or modifier phrases.
- Usually do not return generic words; however, keep a category name, series name, or hierarchy phrase if it explicitly limits the answer scope.
- If there is no entity, return `[]`.

Input. Question: {question}

Output. Return a strict JSON array, with no other text: `["entity_1", "entity_2", ...]`.

G.5 Answer Generation Prompt

Answer Generation Prompt

Role. You answer a closed-corpus QA item using only the retrieved information.

Task. Answer the question based on the provided context.

Input.

```
## Retrieved Information
{context}

## Question
{question}

{answer_format}
```

Answer formats.

- Easy-level: answer only `yes`, `no`, or `unknown`.
- Medium-level: list all items satisfying the condition as a JSON array.
- Hard-level: return only one integer.

G.6 Hit@K Retrieval Judge Prompt

Hit@K Retrieval Judge Prompt

Role. You are judging retrieval recall.

Task. Find the earliest retrieved text that has the same meaning as any gold evidence description. Do not judge whether it answers the question. Do not require exact wording. Keyword overlap alone is not enough.

Input.

```
Question:
{question}

Gold evidence descriptions:
{golden_desc}

Ranked retrieved texts:
{ranked_texts}
```

Output. Return only JSON: `{"hit_rank": 1|2|3|4|5|null, "reason": "short reason"}`.

G.7 Set-Matching Evaluation Prompt

Set-Matching Evaluation Prompt

Role. You are a set-matching evaluator.

Task. Given a gold answer set and a predicted set, determine whether each predicted item refers to the same thing as an item in the gold set.

Input.

```
## Reference Context
{context}

## Items to Evaluate
Gold answer: {gt_list}
Prediction: {pred_list}
```

Matching rules.

- Aliases, abbreviations, and additional modifiers are treated as matches; different categories are not matches.
- Each gold item can match at most one predicted item; matching is one-to-one.
- Prefer the closest literal match when multiple matches are possible.

- Use the context to judge ambiguous cases.

Output. Return JSON only: `{"tp": [[gt_item, pred_item], ...], "fp": [pred_item, ...], "fn": [gt_item, ...]}`.

H LLM Usage

We use LLMs for CWAGraph construction, baseline graph construction, query-time parsing, answer generation, Medium-level set matching, and Hit@K retrieval judging. This section reports the LLM token usage logged by each method's own build and retrieval pipelines, separated into **offline** construction and **online** retrieval/query processing. Notation: M = 10^6 tokens, K = 10^3 tokens.

Table 15: Offline construction LLM usage.

Method	Source Domain	LLM Calls	LLM Input	LLM Output	Total LLM Tokens
CWAGraph	Technical Manuals	3,503	6.64M	3.43M	10.07M
CWAGraph	Food Safety Regulations	3,876	7.42M	5.28M	12.70M
LightRAG	Technical Manuals	1,384	4.73M	3.00M	7.73M
LightRAG	Food Safety Regulations	2,427	7.86M	3.86M	11.71M
GraphRAG	Technical Manuals	11,016	13.71M	8.81M	22.52M
GraphRAG	Food Safety Regulations	6,883	9.65M	6.82M	16.47M
HopRAG	Technical Manuals	5,156	6.33M	0.94M	7.28M
HopRAG	Food Safety Regulations	5,341	7.41M	1.81M	9.22M
HippoRAG	Technical Manuals	2,984	4.30M	0.89M	5.19M
HippoRAG	Food Safety Regulations	4,917	6.60M	2.18M	8.78M

CWAGraph offline LLM usage covers chunk-level joint extraction, complement parsing, and hierarchy-peer scope resolution. Baseline rows report the build-time token logs of each baseline's released implementation.

Table 16: Online retrieval/query LLM usage.

Method	Source Domain	LLM Calls	LLM Input	LLM Output	Total LLM Tokens
CWAGraph	Technical Manuals	439	102.1K	6.8K	109.0K
CWAGraph	Food Safety Regulations	382	116.9K	12.7K	129.5K
LightRAG	Technical Manuals	263	162.6K	33.5K	196.1K
LightRAG	Food Safety Regulations	231	146.7K	43.2K	189.9K
GraphRAG	Technical Manuals	0	0	0	0
GraphRAG	Food Safety Regulations	0	0	0	0
HopRAG	Technical Manuals	7,261	3.79M	0.07M	3.86M
HopRAG	Food Safety Regulations	8,269	4.16M	0.08M	4.24M
HippoRAG	Technical Manuals	310	901.8K	13.5K	915.3K
HippoRAG	Food Safety Regulations	253	805.3K	14.1K	819.5K

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions claimed in the abstract and Section 1 (CWA layer for retrievable LCSs, the SetQA benchmark, consistent gains over baselines) match the experimental results in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Discussed in Appendix 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Propositions 1 and 2 are stated in Section 3 with assumptions explicit; full proofs are in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Method details are in Section 4 and Appendix C; baseline configurations, evaluation protocol, prompts, and LLM usage are in Appendix D, Appendix G, and Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: An anonymized supplementary package provides the SetQA data, CWAGraph implementation, baseline/evaluation scripts, and reproduction instructions. The package will be made public after acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Hyperparameters, retrieval settings, and the evaluation protocol are reported in Section 5.1 and Appendix D.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The main, ablation, and oracle tables report results from the saved final evaluation artifacts. We do not include explicit error bars or statistical significance tests; all per-question outputs and retrieval artifacts are retained for auditability.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The experiments use commodity CPU workers for orchestration and all LLM/embedding calls are made through the OpenAI API; no local GPU is required. Appendix H reports the dominant compute cost as per-method LLM token usage.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics; our research conforms in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: The work supports closed-corpus question answering for documents such as regulations and technical manuals, which can aid compliance review and operator-facing decision support. A potential negative impact is over-reliance on retrieved answers in high-stakes settings; outputs should remain source-grounded decision support rather than final authority.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: SetQA is constructed from publicly accessible technical documentation (Huawei S-series manuals) and a national standard (GB 2760). The benchmark contains no personal data, sensitive content, or material requiring controlled release.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: The paper cites the baseline methods and identifies the public source corpora. The supplementary package contains our own implementation and evaluation scripts, and does not redistribute third-party baseline implementations. Source-corpus usage and OpenAI API usage are documented in the paper and appendix.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: SetQA and the CWAGraph implementation are new assets. They are documented in the appendix and included in the anonymized supplementary package, with public release planned after acceptance.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or human-subject studies. Annotation was performed on publicly available documents to construct QA pairs, and no participant data was collected.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Annotation involved multiple expert annotators reviewing publicly available technical and regulatory documents to construct QA pairs. No personal data or sensitive material was involved, and IRB approval was not required for this kind of low-risk text annotation under our institution's policies.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: GPT-5.2 is used for SetQA candidate extraction, CWAGraph construction, baseline graph/query processing where applicable, answer generation, and judging; described in Section 4 and Appendix H.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.