Efficient KG-Augmented RAG with Reusable Graph Community Summaries

Maha Karkout, Maria Khodorchenko, Nikolay Butakov, Denis Nasonov
ITMO University, Saint Petersburg, Russia
mahaquarqout123@gmail.com, marivaxod@yandex.ru, alipoov.nb@gmail.com, denis.nasonov@gmail.com

Abstract

Retrieval-augmented generation (RAG) performs well for localized factual queries but struggles with complex questions requiring multi-section evidence integration. Graph-based approaches introduce relational structure, yet their practical integration into QA pipelines involves significant query-time overhead. We present a practical KG-augmented RAG (KG-RAG) design that builds a knowledge graph offline with an LLM, converts graph communities into reusable summaries, and retrieves these summaries jointly with textual evidence at query time. We compare dense RAG, pure GraphRAG, and the proposed hybrid on two benchmarks representing complementary retrieval paradigms: QASPER (intra-document reasoning over scientific papers) and ObliQA (cross-document reasoning over regulatory texts). Results show that pure GraphRAG does not consistently outperform dense retrieval, whereas the hybrid configuration systematically improves relevance, correctness, and completeness while maintaining substantially lower latency than full graph-based inference.

Keywords: GraphRAG; KG-RAG; Knowledge Graph Summarization; Multi-hop Question Answering; Large Language Models

1. Introduction

Large language models (LLMs) are central to contemporary question answering, especially when paired with retrieval-augmented generation (RAG) (Lewis et al., 2021). By grounding responses in retrieved passages, RAG improves factual consistency and reduces hallucination. However, dense retrieval remains limited for questions that require integrating dispersed evidence or reasoning over implicit relations across long documents (Parekh et al., 2026; Liu et al., 2023; Huang et al., 2025).

Knowledge graphs (KGs) offer an appealing complement because they expose entities, relations, and higher-level corpus structure. Yet graph-based QA introduces two practical trade-offs: graph abstraction can suppress fine-grained evidence needed for accurate answers, and graph-centric query-time reasoning can be substantially slower than standard retrieval (Edge et al., 2025). We therefore study a practical hybrid KG-augmented RAG (KG-RAG) design in which a KG is built offline, graph communities are summarized into reusable textual resources, and these summaries are retrieved jointly with raw evidence at inference time.

We evaluate on two benchmarks representing complementary retrieval regimes: QASPER (Dasigi et al., 2021), which requires document-bounded reasoning over scientific papers, and ObliQA (Gökhan et al., 2024), which requires cross-document reasoning over regulatory texts. These datasets are used not to claim broad domain coverage, but to test the hybrid setting under two structurally different retrieval conditions. We compare three configurations: (i) dense RAG, (ii) GraphRAG with global search, and (iii) a hybrid RAG augmented with reusable community summaries.

Our contributions are as follows:

- We introduce a hybrid KG-RAG framework that transforms communities in an LLM-constructed knowledge graph into reusable retrieval units, enabling relation-aware augmentation of dense retrieval without expensive query-time graph reasoning.
- We study how graph construction choices, notably ontology granularity and resulting community structure, affect QA behavior and when graph-derived abstraction is helpful.
- We provide empirical evidence for a quality-efficiency trade-off: retrieving graph-derived resources improves hard cases while avoiding heavy query-time global graph inference.

2. Related Work

Retrieval-Augmented Generation. RAG grounds LLM outputs in retrieved evidence and is the dominant paradigm for knowledge-intensive QA (Lewis et al., 2021; Gao et al., 2024). Its main weakness is dense retrieval over dispersed evidence, especially for document-level and structurally complex questions (Liu et al., 2023; Parekh et al., 2026; Huang et al., 2025).

Knowledge Graphs and LLMs. Recent work studies how KGs can ground, augment, and evaluate LLM outputs, while LLMs can in turn support KG construction and querying (Pan et al., 2024; Ma et al., 2025). Prompt-based entity and relation extraction now enables KG construction from unstructured text (Trajanoska et al., 2023; Wang and Tsung,

Proceedings of the Knowledge Graphs and Large Language Models Workshop 2026 (KG-LLM@LREC26), pages 110–119
May 16, 2026.

2025), but downstream utility remains highly sensitive to ontology quality (Paulheim, 2016).

Graph-Augmented Retrieval. GraphRAG combines LLM-based graph construction, community detection, and hierarchical summarization, but relies on multi-stage graph reasoning at query time (Edge et al., 2025). Related graph-augmented QA systems extend this direction to broader settings (Cao et al., 2025). RAPTOR provides multi-level textual abstraction without explicit graphs (Sarthi et al., 2024). Our work is positioned as a practical integration of these ideas: we reuse graph community summaries as retrieval units inside a standard dense pipeline, avoiding both heavy query-time graph reasoning and tree-construction assumptions.

3. Method

Our approach uses the Microsoft GraphRAG framework[1] to augment standard retrieval-augmented generation (RAG) with structured relational representations. The system has two stages: (1) **indexing**, which performs graph construction and community summarization, and (2) **inference**, which performs hybrid retrieval and answer generation.

A schematic overview is shown in Figure 1.

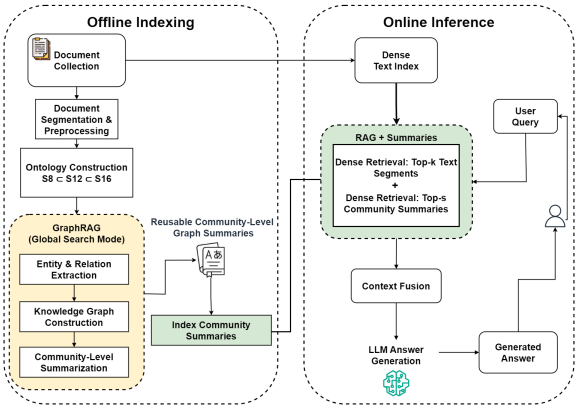

Figure 1: KG-RAG pipeline. Offline indexing builds a typed graph, detects communities, and indexes their summaries; online inference retrieves text segments and community summaries jointly for answer generation.

3.1. Indexing: Graph Construction and Summarization

During indexing, we construct a knowledge graph from the document collection $D = \{d_1, \ldots, d_n\}$. The GraphRAG workflow performs: (1) ontology-constrained entity extraction, (2) relation extraction within the same textual scope, (3) graph assembly, (4) community detection, and (5) community-level summarization.

Graph construction is guided by a typed ontology; details of ontology induction and hierarchical configurations are given in Section 3.1.1. Because graph construction, clustering, and summarization are all done offline, indexing yields entities, relations, communities, and community summaries. For downstream QA, the summaries are indexed as additional retrieval units alongside the original text segments.

3.1.1. Ontology Construction

Graph construction is ontology-guided: the ontology determines admissible entity categories, which in turn shape extracted entities and relations, community structure, and summary content (Paulheim, 2016). Because the summaries are retrieved at inference time, ontology granularity directly controls the level of semantic abstraction introduced into QA.

We therefore study three nested configurations, $S_8 \subset S_{12} \subset S_{16}$, to evaluate how semantic granularity affects graph structure and downstream QA.

Ontology construction follows two stages. First, we induce a candidate type inventory from cleaned, stratified segments using constrained LLM prompting with target cardinalities of 8, 12, and 16 types (e.g., *Model*, *Dataset*, *Regulation*, *Organization*); when this produces generic or inconsistent categories, we instead aggregate provisional extraction labels over the corpus. Second, we prune and canonicalize the inventory by filtering rare artifacts (threshold τ), merging redundant labels, and removing overly generic types. We then instantiate S_8, S_{12}, and S_{16} by selecting the most frequent stable types for S_8 and refining coarser categories for the larger variants while preserving nesting.

3.2. Inference: Hybrid Retrieval and Answer Generation

Given a question q, we retrieve top-k text segments $T_k(q)$ and top-s community summaries $G_s(q)$ using the same embedding model and similarity-search infrastructure. The final context is $C(q) = T_k(q) \cup G_s(q)$.

Operationally, the community summaries act as graph-derived textual resources retrieved alongside the original text. The resulting context combines direct evidence with relation-aware abstraction and is passed to the generation model via structured constrained prompting.

3.3. Benchmarks

QASPER is a document-bounded QA benchmark over about 888 NLP papers and about 5,000 ques-

[1] https://github.com/microsoft/graphrag

tions, often requiring multi-section intra-document reasoning (Dasigi et al., 2021). ObliQA is a regulatory QA benchmark over 40 policy documents and more than 1,000 questions, many of which require cross-document evidence integration (Gökhan et al., 2024). These complementary settings motivate per-paper graphs for QASPER and a single global graph for ObliQA.

4. Experimental Setup

4.1. Baseline: Text-Based Retrieval-Augmented Generation

We first establish a dense text-based Retrieval-Augmented Generation (RAG) baseline for both datasets. This system serves as the reference point against which graph-based configurations are evaluated.

Retrieval. All documents are indexed in Elasticsearch using `e5-mistral-7b-instruct` embeddings and cosine similarity. For QASPER, text is segmented into paragraph-level units (about 48,000 indexed segments), and retrieval is restricted to the paper associated with each question to prevent cross-document leakage. For ObliQA, all 40 regulatory documents are indexed without document-level restriction. Retrieval depth was selected empirically over $k \in \{5, 10, 20, 25\}$; based on recall and context-length trade-offs, we use Top-$k = 20$ throughout.

Generation. Answers are generated using Qwen3-32B (Qwen Team, 2025)[2] with temperature $= 0.6$, Top-$P = 0.95$ and a maximum of 400 tokens. The same model is used for all LLM-dependent stages: entity and relation extraction, community summarization, answer generation, and evaluation. Outputs are produced in structured JSON format, with retry-based parsing to enforce schema validity.

The prompt instructs the model to: (i) use only the retrieved context, (ii) avoid external knowledge, and (iii) return *"Insufficient information"* if the context does not support an answer.

This configuration is shared across all systems to ensure controlled comparison.

4.2. Gold Answers and Evaluation Protocol

QASPER. We evaluate generated answers against the provided gold annotations and evidence spans.

ObliQA. Because ObliQA does not provide canonical abstractive gold answers, we build reference answers using Qwen3 conditioned on the provided passages and retain only answers that pass a confidence-and-coverage filtering step. This yields 1,250 QA pairs. Because the same model family is also involved in reference construction, these scores should be interpreted primarily as controlled comparisons across systems.

4.3. Failure-Focused Evaluation

To analyze cases where dense retrieval is insufficient, we adopt a strict evaluation rule.

Each generated answer is evaluated using an LLM-as-a-Judge protocol (Zheng et al., 2023) (Qwen3) on three criteria: relevance, correctness, and completeness, each scored from 1 (very poor) to 5 (fully correct and comprehensive), with 3 indicating partial correctness. We treat these scores primarily as a controlled comparative signal across systems rather than an absolute substitute for human evaluation, because judge-based assessment may introduce model-dependent bias. This risk is partially mitigated by using the same judge model, prompt format, and scoring rubric for all evaluated configurations.

An answer is considered successful if: Relevance ≥ 4 and Correctness ≥ 4 and Completeness ≥ 3. Otherwise, it is classified as failed. This strict criterion ensures that only strongly correct and sufficiently complete answers are considered recovered. The failure subset includes cases where: the model returned "Insufficient information", the retrieved context did not contain relevant evidence, or the answer failed to meet the correctness and completeness thresholds.

Applying this rule to the RAG baseline yields failure-focused subsets of questions on which all subsequent configurations are evaluated: ObliQA: 188 questions; QASPER: 186 questions.

For QASPER, these failed questions correspond to 99 distinct papers. Only these papers are subsequently indexed using GraphRAG, ensuring that graph construction focuses on documents where baseline retrieval proved insufficient.

These failure subsets form the benchmark for evaluating graph-based and hybrid configurations.

4.4. GraphRAG Indexing and Configuration

We apply the Microsoft GraphRAG workflow described in Section 3 to construct structured relational representations for the failure-focused subsets. The same LLM endpoint (`Qwen3`) and embedding model (`e5-mistral-7b-instruct`) are used throughout extraction, summarization, and retrieval to maintain consistency across systems.

While the core workflow remains unchanged, the graph construction strategy differs between the two datasets, reflecting their distinct retrieval paradigms (Section 3).

[2] https://huggingface.co/Qwen/Qwen3-32B

4.4.1. QASPER: Per-Paper Graph Construction (Chat-with-the-Document)

In the chat-with-the-document setting, each question targets a specific paper and reasoning is bounded by that document. To preserve document boundaries and prevent cross-document leakage, we construct independent GraphRAG graphs per paper. Each graph captures the intra-document relational structure - how research entities, methods, datasets, and results connect within a single text.

Prior to graph construction, QASPER papers undergo additional cleaning to reduce extraction noise, including removal of LaTeX artifacts and equation fragments, filtering of table-like segments, and structural normalization.

Graph construction is performed at the section level rather than paragraph level to preserve discourse coherence during entity and relation extraction.

For each of the 99 selected papers: (1) A separate GraphRAG project is initialized. (2) Sections serve as input units. (3) Entity and relation extraction are performed using the corpus-induced ontology. (4) Community detection is applied with a maximum cluster size of 10. (5) Community summaries are generated and embedded.

Extraction prompts and community report templates are adapted to reflect scientific research structure and include dataset-specific examples, ensuring alignment between entity typing and downstream QA objectives.

At inference time, only the graph corresponding to the question's associated paper is queried. This design enforces strict document scoping and controlled evaluation aligned with QASPER's formulation.

4.4.2. ObliQA: Global Regulatory Graph

In contrast, ObliQA contains regulatory questions that may require cross-document reasoning and institutional linkage. We therefore construct a single global graph over the entire corpus of 40 regulatory documents. Each document is provided as a structured JSON file containing ordered passages.

All documents are included in a unified GraphRAG project. Input segments correspond to the ordered passage units provided in the document files. Entity and relation extraction are guided by the corpus-derived regulatory ontology. Community detection is performed globally across the corpus, and community summaries are generated and embedded for retrieval.

Extraction prompts and community summary templates are adapted to reflect regulatory terminology, institutional actors, and procedural relationships present in ObliQA.

4.4.3. Query Mode and Retrieval

For the GraphRAG configuration, question answering is performed using the framework's global search mode, which applies multi-stage LLM reasoning over retrieved community reports and graph artifacts.

For the Hybrid (RAG + Summaries) configuration, dense similarity search retrieves Top-$k = 20$ textual segments and Top-$s = 5$ community summaries. These are concatenated into a unified context and provided to the generation model in a single inference pass.

Conceptually, the two configurations differ in how community information is incorporated. In global search mode, community reports actively participate in hierarchical query-time reasoning, where intermediate outputs are iteratively combined into a final answer. In contrast, in the summary-based configuration, community summaries function as additional retrieval units within a standard retrieval-and-generation pipeline. The graph therefore shapes how information is organized and exposed to the model, rather than introducing additional reasoning stages during inference.

4.5. Ontology Induction (Dataset-Specific Instantiation)

We instantiate the ontology construction procedure described in Section 3.1.1 separately for each dataset. Below we report the dataset-specific induction process and the resulting nested configurations used for GraphRAG node typing.

4.5.1. QASPER

Motivation. QASPER questions primarily concern research artifacts such as models, datasets, evaluation metrics, experimental setups, and reported results. The ontology must therefore capture the structural organization of scientific papers.

Induction Procedure. Ontology induction is performed on 300 cleaned paragraph-level segments sampled from the QASPER corpus (approximately 47,000 paragraphs), stratified across papers and section types to ensure structural diversity.

An LLM (Qwen3) is prompted to generate entity type inventories under controlled constraints, targeting 8, 12, and 16 distinct entity types. This procedure yields three ontology variants corresponding to increasing semantic granularity.

Resulting Ontology Structure. The smallest configuration S_8 captures core research structure and includes entity types such as Model, Dataset, Method, Result, EvaluationMetric, Task, Baseline, and Section.

The S_{12} and S_{16} configurations extend this base with progressively finer experimental distinctions,

including categories such as Experiment, Hyperparameter, TrainingStrategy, Architecture, and EvaluationSetting.

4.5.2. ObliQA

Motivation. ObliQA consists of regulatory and compliance documents characterized by institutional actors, legal rules, financial instruments, and procedural requirements. Its semantic inventory is domain-specific and reflects regulatory structures.

Due to this institutional specificity, applying the generative schema induction strategy used for QASPER produced unstable and overly generic categories that failed to capture regulatory distinctions reliably. Instead, ontology induction for ObliQA is derived empirically from structured extraction outputs.

Extraction and Raw Type Inventory. Triplet extraction and provisional entity typing are performed using the Wikontic pipeline.[3] This system segments documents into chunks, extracts $(subject, relation, object)$ triplets, and assigns provisional entity type labels during alignment (Chepurova et al., 2026).

This process yields a corpus-level inventory of entity types. High-frequency raw types include regulation (875), organization (609), document (588), legal concept (575), financial product (509), rule (444), approval process (304), legal person (292), and legal instrument (282).

The distribution exhibits a long-tail pattern, with many low-frequency and overly specific categories.

Pruning and Canonicalization. To construct a stable ontology for graph building, we apply frequency-based filtering followed by quality-guided consolidation.

Entity types are ranked by corpus frequency, and those occurring fewer than 70 times are excluded. Such types typically correspond to extraction artifacts, rare procedural subtypes, or overly specific domain variants. This threshold preserves semantically stable categories while reducing noise from the long-tail distribution.

The remaining types are inspected for redundancy and semantic overlap. For example, legal person and juridical person are unified under the canonical category LegalEntity. Closely related institutional variants are consolidated under Organization, while overly generic labels such as entity, text, or information are removed.

This process yields a compact and reproducible type inventory suitable for GraphRAG node typing.

Resulting Ontology Structure. The smallest configuration S_8 captures core regulatory structure and includes entity types such as Regulation, Organization, LegalEntity, Rule, FinancialInstrument, ApprovalProcess, LegalInstrument, and Policy.

The S_{12} and S_{16} configurations extend this base with progressively finer regulatory distinctions. S_{12} introduces structured obligation categories such as ComplianceRequirement, RiskManagementRequirement, CapitalRequirement, and ReportingRequirement. S_{16} further refines the representation by incorporating additional domain-specific concepts including Exposure, Collateral, LegalDefinition, and ClientCategory.

[3] https://github.com/screemix/Wikontic

Ontology	Entities	Relations	Communities
S_8	171	164	34
S_{12}	197	199	38
S_{16}	197	189	37

Table 1: Structural statistics for QASPER under different ontology variants (average per paper).

5. Experiments and Results

5.1. Ontology Variant Analysis (Graph-Level Study)

Before running QA inference, we analyze how ontology granularity affects graph structure. For each dataset, graphs are constructed under three ontology configurations and their structural properties are examined to select a configuration for downstream evaluation.

5.1.1. QASPER

Graph statistics averaged per paper are reported in Table 1.

S_8 produces compact graphs centered on high-level research concepts. However, experimental configuration details frequently collapse into generic nodes, limiting representation of methodological structure.

S_{12} introduces explicit representation of experimental components and hyperparameters, resulting in richer relational structure and more specialized communities.

S_{16} increases contextual coverage but also expands extraction toward peripheral content (e.g., acknowledgements and funding references), leading to community fragmentation without improving structural density.

Based on structural coherence and interpretability, S_{12} was selected for graph construction in QASPER.

5.1.2. ObliQA

Aggregate graph statistics for the global regulatory graph are reported in Table 2.

Ontology	Entities	Relations	Communities
S_8	∼6k	∼11k	∼800
S_{12}	∼22k	∼19k	∼1400
S_{16}	∼38k	∼30k	∼2000

Table 2: Structural statistics for ObliQA under different ontology variants (global graph).

Model	Relevance	Correctness	Completeness	Overall
Baseline RAG	3.12	2.29	1.79	2.40
GraphRAG	3.12	2.19	1.93	2.41
RAG + Summaries	3.34	2.29	2.14	2.59

Table 3: Mean evaluation scores on ObliQA (188 questions). Overall is the mean of the three criteria.

Under S_8, distinct regulatory obligations are frequently merged into broad *Rule* or *Regulation* nodes, limiting fine-grained obligation tracing.

S_{12} improves representation of compliance processes and supervisory structure, yielding more differentiated communities.

S_{16} captures finer-grained distinctions central to regulatory reasoning, including capital requirements, exposure limits, and legal definitions. The resulting communities align closely with regulatory workflows and financial supervision mechanisms.

Based on structural coherence and domain coverage, S_{16} was selected for graph construction in ObliQA.

5.2. Question Answering Performance

All systems are evaluated using a 1-5 scoring framework across three dimensions: *Relevance*, *Correctness*, and *Completeness*. We report macro-averaged mean scores and complement them with score distributions and runtime statistics. We compare three configurations: (i) Baseline RAG, (ii) GraphRAG, and (iii) RAG + Summaries (Hybrid).

5.2.1. QA Performance on ObliQA

All configurations are evaluated on the same set of 188-questions failure-focused subset of the baseline RAG system.

Mean scores. Table 3 reports mean scores across the three criteria. GraphRAG yields only marginal improvement over the baseline in completeness (+0.14) while slightly reducing correctness. In contrast, the Hybrid configuration improves all dimensions, with the largest gain in completeness (+0.35 over baseline, +0.21 over GraphRAG).

Score distributions. Mean values can hide important behavior differences. Figure 2 shows score histograms for relevance, correctness, and completeness. The Hybrid configuration produces the strongest right-shift, particularly for relevance and completeness, indicating improved alignment with question intent and more developed answers.

Model	Mean (s)	Median (s)	p95 (s)	Max (s)
GraphRAG	311.29	256.77	604.31	1230.09
RAG + Summaries	11.94	11.64	18.85	26.25

Table 4: Runtime per question on ObliQA.

Model	Relevance	Correctness	Completeness	Overall
Baseline RAG	3.50	1.92	1.80	2.41
GraphRAG	2.79	1.96	1.63	2.13
RAG + Summaries	4.13	2.78	2.51	3.14

Table 5: Mean evaluation scores on QASPER (186 questions). Overall is the mean of the three criteria.

Runtime. Table 4 reports runtime per question. GraphRAG is substantially slower due to multi-step graph operations, while the Hybrid approach maintains near-RAG latency.

Per-question delta analysis. To complement aggregate metrics, we analyze per-question differences between the Hybrid and baseline RAG configurations. For each question, we compute the overall-score delta

$$\Delta = \text{Hybrid} - \text{RAG},$$

where the overall score is the mean of relevance, correctness, and completeness.

Figure 3 shows the distribution of per-question deltas on ObliQA. The distribution is centered around zero, with a balanced spread across positive and negative values. This reflects a more heterogeneous effect of the Hybrid configuration, where improvements are concentrated on a subset of questions rather than uniformly distributed.

At the same time, the positive tail extends further than the negative side, indicating that when improvements occur, they tend to be larger in magnitude.

5.2.2. QA Performance on QASPER

Evaluation is conducted on the 186-question failure-focused subset of the baseline RAG system. GraphRAG operates over paper-scoped graphs: each question queries only its corresponding paper graph.

Mean scores. Table 5 shows mean scores. GraphRAG degrades overall performance relative to the baseline, reducing both relevance and completeness. The Hybrid configuration substantially improves all metrics, with the strongest gain in relevance, indicating better contextual grounding and alignment.

Score distributions. Figure 4 shows score histograms. The Hybrid configuration yields a pronounced right-shift across all dimensions, indicating more relevant, correct, and complete answers.

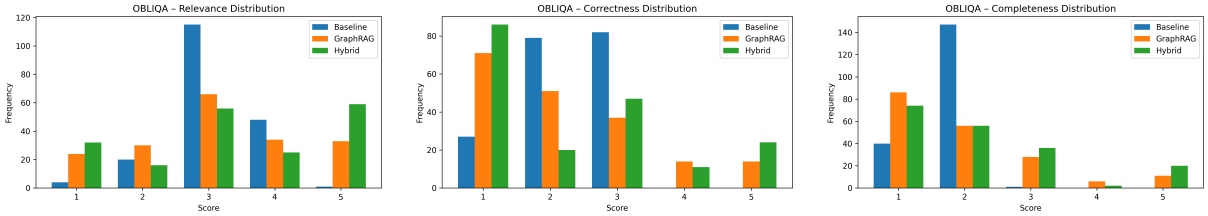

Figure 2: Score distributions on ObliQA (failure subset).

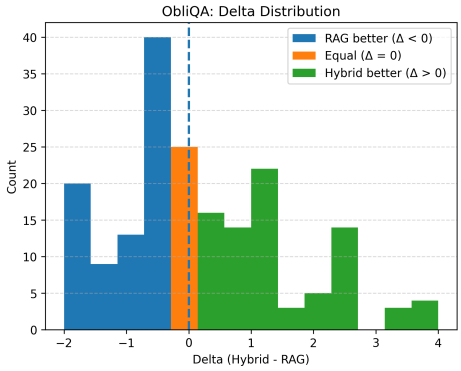

Figure 3: Distribution of per-question overall-score deltas on ObliQA (Δ = Hybrid − RAG).

Model	Mean (s)	Median (s)	p95 (s)	Max (s)
GraphRAG	70.25	63.99	153.15	165.70
RAG + Summaries	5.04	4.35	8.85	16.04

Table 6: Runtime per question on QASPER.

Runtime. Table 6 reports runtime per question. GraphRAG runs over paper-scoped graphs rather than a global corpus graph. The Hybrid approach is faster on average and notably faster in median latency, while still achieving higher quality.

Per-question delta analysis. Figure 5 shows the corresponding delta distribution on QASPER. The distribution is clearly shifted toward positive values, with most questions exhibiting positive deltas. This indicates that the Hybrid configuration improves performance across a large portion of the dataset.

Furthermore, the presence of larger positive deltas shows that these improvements are not only frequent but also substantial.

An illustrative example of a positive Δ case is shown in Figure 6. It highlights how the Hybrid configuration recovers both answer structure and numeric evidence that are missing in the baseline.

6. Discussion

The results across both datasets reveal a consistent pattern: the hybrid configuration delivers the strongest improvements, while pure GraphRAG provides limited gains over dense retrieval alone. Taken together, these results position the contribution of this work primarily at the level of practical system design and empirical analysis. We highlight four observations.

1. Graph structure alone does not guarantee better answers. GraphRAG improves information organization through entity extraction, relation modeling, and community summarization. However, QA requires precise textual grounding, and graph-level abstractions compress fine-grained details. This leads to persistent mid-range scores (2-3) and limited gains in high-confidence correctness.

2. Hybrid integration preserves precision while adding structure. By combining dense retrieval with graph summaries, the hybrid configuration consistently shifts score distributions toward higher values. Improvements appear not only in means but also in distributional behavior: more answers reach the 4-5 range, particularly in relevance and completeness. This pattern also appears at the per-question level, where positive changes are observed across many individual cases while performance remains comparable elsewhere. This confirms that graph summaries are most effective when used to enrich rather than replace retrieval.

Summary quality likely governs how much the hybrid design can help. Community summaries are beneficial when they preserve question-relevant relations while remaining specific enough to support downstream answer generation. If summarization is too coarse, graph-level abstraction may highlight the right topical region but still lose details required for correctness and completeness. This interpretation is consistent with our results: pure GraphRAG shows limited benefit, while the hybrid setting performs better because summaries add relational context and retrieved text preserves the precise evidence needed for grounding.

3. Efficiency considerations. Across both ObliQA and QASPER, pure GraphRAG introduces substantial inference overhead, while the hybrid approach maintains near-RAG latency. This consistent quality–efficiency balance favors hybrid augmentation.

4. Graph construction reflects the retrieval paradigm. In QASPER's chat-with-the-document setting, per-paper graphs capture intra-document structure; the large relevance gain (+0.63) suggests that localized summaries help align answers with document organization. In ObliQA's collection-

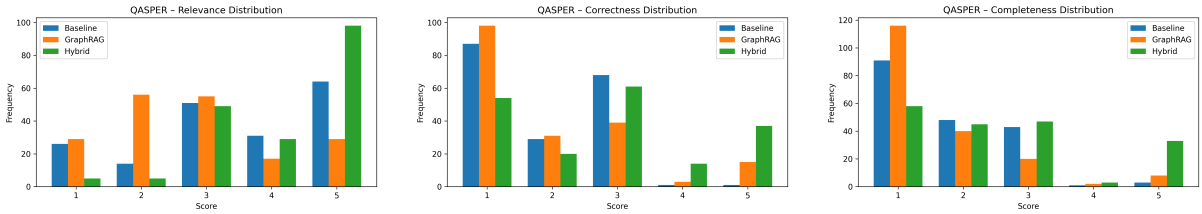

Figure 4: Score distributions on QASPER (failure subset).

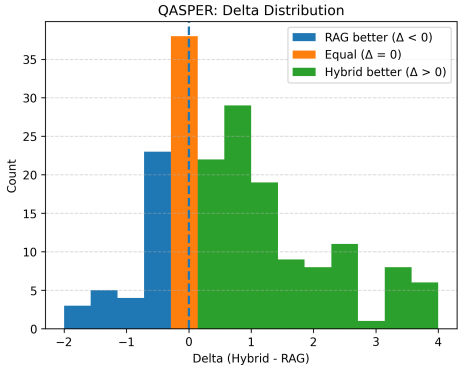

Figure 5: Distribution of per-question overall-score deltas on QASPER (Δ = Hybrid − RAG).

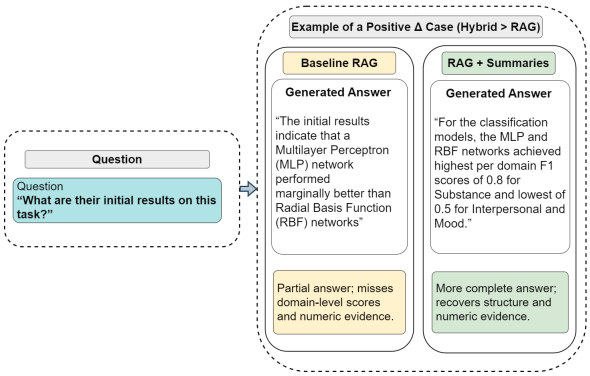

Figure 6: Example of a positive Δ case. Baseline RAG captures only a coarse comparison and omits domain-level scores and numeric evidence, while the Hybrid configuration recovers both answer structure and quantitative details.

based setting, the global graph exposes cross-document regulatory linkages; completeness gains (+0.35) reflect the ability of community summaries to surface dispersed provisions. The choice between per-document and global construction thus follows directly from the retrieval paradigm.

Overall. Dense retrieval remains essential for evidence grounding, but graph-derived summaries provide complementary contextual organization. The hybrid design consistently improves answer quality without incurring the latency of full graph-based reasoning, validating the proposed approach across both retrieval paradigms.

7. Conclusion

This work investigated whether graph-based semantic structure improves question answering beyond dense retrieval. Through experiments on scientific (QASPER) and regulatory (ObliQA) datasets, we compared pure GraphRAG, dense RAG, and a hybrid configuration integrating reusable community-level summaries.

The results show that graph structure alone does not consistently translate into higher factual accuracy or completeness. In contrast, combining dense retrieval with graph-derived summaries yields systematic improvements across relevance, correctness, and completeness, while maintaining favorable efficiency characteristics.

These findings suggest that structured semantic abstraction is most effective when used to enhance retrieval rather than replace it. By demonstrating that reusable community summaries can improve retrieval quality without the overhead of full graph-based global search inference, this work offers an empirical characterization of when graph-derived summaries help within a dense retrieval setting.

8. Ethics Statement and Limitations

Ethical Considerations. This work evaluates graph-based summarization and retrieval augmentation for complex QA using publicly available datasets (QASPER and ObliQA), no personal or sensitive data are involved. The framework relies on large language models for extraction, graph construction, summarization, and answer generation. As with all LLM-based systems, outputs may contain inaccuracies or inherited biases. Although retrieval grounding and structured summaries mitigate unsupported generation, they do not eliminate risk; deployment in high-stakes settings therefore requires appropriate human oversight and verification.

Limitations. Several limitations should be noted.

First, answer quality is evaluated using an LLM-as-a-judge protocol, despite structured criteria and stability checks, automated evaluation may intro-

duce model-dependent bias, and human assessment would provide stronger validation.

Second, graph construction depends on ontology design and extraction prompts. Although we analyze alternative configurations and select stable variants, different schemas may lead to varying structural properties and downstream behavior.

Finally, experiments are restricted to scientific and regulatory documents; broader domain validation is needed to assess generalizability.

9. Acknowledgements

This work was supported by the Russian Science Foundation, agreement no. 24-71-00115, https://rscf.ru/en/project/24-71-00115/.

10. Bibliographical References

Yukun Cao, Zengyi Gao, Zhiyang Li, Xike Xie, S. Kevin Zhou, and Jianliang Xu. 2025. Lego-graphrag: Modularizing graph-based retrieval-augmented generation for design space exploration. *Proceedings of the VLDB Endowment*, 18(10):3269–3283.

Alla Chepurova, Aydar Bulatov, Mikhail Burtsev, and Yuri Kuratov. 2026. Wikontic: Constructing wikidata-aligned, ontology-aware knowledge graphs with large language models.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. From local to global: A graph rag approach to query-focused summarization.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Xinyue Huang, Ziqi Lin, Fang Sun, Wenchao Zhang, Kejian Tong, and Yunbo Liu. 2025. Enhancing document-level question answeringvia multi-hop retrieval-augmented generationwith llama 3. *Preprints*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts.

Chuangtao Ma, Yongrui Chen, Tianxing Wu, Arijit Khan, and Haofen Wang. 2025. Large language models meet knowledge graphs for question answering: Synthesis and opportunities.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

Jash Rajesh Parekh, Pengcheng Jiang, and Jiawei Han. 2026. Structure-augmented reasoning generation.

Heiko Paulheim. 2016. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8:489–508.

Qwen Team. 2025. Qwen3 technical report.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

Milena Trajanoska, Riste Stojanov, and Dimitar Trajanov. 2023. Enhancing knowledge graph construction using large language models.

Luxuan Wang and Fugee Tsung. 2025. Automated knowledge graph construction for supply chain datasets assisted by llms. In *2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)*, pages 2738–2743.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

11. Language Resource References

Dasigi, Pradeep and Lo, Kyle and Beltagy, Iz and Cohan, Arman and Smith, Noah A. 2021. *QASPER: A Dataset for Question Answering over Scientific Papers*. PID https://allenai.org/data/qasper.

Gökhan, Tuba and Wang, Kexin and Gurevych, Iryna and Briscoe, Ted. 2024. *ObliQA: Obligation-Based Question Answering Dataset for Regulatory Compliance*. PID https://github.com/RegNLP/ObliQADataset.