MixRAG: a Low-cost alternative to a Graph RAG for scientific papers

Luis Cesar de Azevedo[1], **Arthur P. Machado**[2] , **Ronaldo C. Prati**[1]

[1]Center of Mathematics, Computation and Cognition
Federal University of ABC (UFABC)
Av. dos Estados, 5001, Santo André, SP 09280-560, Brazil

[2]Universidade Estadual de Campinas (UNICAMP), Campinas, SP, Brazil ,

lcdeazevedo@gmail.com,ronaldo.prati@ufabc.edu.br

Abstract. *Retrieval-Augmented Generation (RAG) combines Large Language Models (LLMs) with retrieval systems to enhance the relevance and accuracy of AI-generated responses. For complex texts, such as academic papers, GraphRAG, a variant of RAG, further refines context quality by integrating Knowledge Graphs (KGs) into the retrieval process. Unlike traditional RAG, GraphRAG uses KGs to model entities, relationships, and hierarchies explicitly, capturing richer semantic connections (e.g., causal links in research findings or experimental dependencies). This structured, graph-based approach enables LLMs to reason over interconnected concepts, improving both the accuracy and coherence of outputs in technical or nuanced domains. However, the use of LLMs for constructing KGs becomes costly for large text sets, as the process requires repeated LLM processing of text to extract and structure entities and relationships. In this paper, we introduce MixRAG, a cost-effective RAG/GraphRAG hybrid approach. MixRAG optimizes resource usage by strategically generating a KG only from document abstracts, while storing the remainder as raw text in the vector database. This selective structuring balances the efficiency of a simple RAG with the informational depth of GraphRAG. Our experiments show that MixRAG reduces KG construction costs by more than 95%, while improving the response quality.*

1. Introduction

Large Language Models (LLMs) achieve state-of-the-art performance in natural language processing by encoding statistical patterns and linguistic structures into billions of parameters [Brown et al. 2020, Yin et al. 2023]. While they demonstrate strong reasoning and in-context learning capabilities [Achiam et al. 2023, Grattafiori et al. 2024], their internal knowledge remains static and lacks explicit symbolic grounding. Consequently, LLMs face challenges, notably hallucinations and fabricated citations, as they prioritize linguistic coherence over factual accuracy [Huang et al. 2023, Dahl et al. 2024]. This limits their reliability for domain-specific or temporally sensitive tasks.

To address these limitations, Retrieval-Augmented Generation (RAG) [Lewis et al. 2020] incorporates external knowledge by retrieving relevant context from a corpus via similarity search. However, traditional RAG often fails to retrieve context dispersed across multiple documents, leading to incomplete answers. To overcome

this, GraphRAG [Peng et al. 2024, Han et al. 2025] integrates Knowledge Graphs (KGs) to provide structured contextual information. By pre-extracting entities and relationships via LLM calls, GraphRAG explores semantic connections, significantly improving retrieval relevance. Numerous implementations have recently emerged, underscoring its growing adoption, such as DIGIMON, fast-graphrag, neo4j-graphrag-python, and Microsoft™ -graphrag.

Despite its benefits, constructing a KG over an entire corpus demands repeated LLM processing [Guo et al. 2024, Zhuang et al. 2025]. For documents like academic articles, extracting entities, relationships, and descriptions translates to high computational and operational costs.

To minimize these costs while maintaining contextual quality, we propose MixRAG, a hybrid method optimized for academic articles. MixRAG constructs a KG exclusively from article abstracts, leveraging their brevity and dense core contributions while storing full-text content in a standard vector database. This strategy generates a context from the abstract-based KG, complemented by full-text similarity search, ensuring high retrieval efficiency and accuracy at a fraction of the cost.

Our primary contributions are as follows:

- MixRAG Framework: A cost-efficient hybrid approach that constructs KGs only from abstracts while storing full-text chunks in a vector database, balancing the depth of GraphRAG with the efficiency of Naive RAG.
- Significant Cost Reduction: Demonstrates a 95% reduction in KG construction costs compared to full-text GraphRAG, validated through real-world experiments on a scientific corpus.
- Enhanced Response Quality: Empirical results from domain specialists show that MixRAG outperforms GraphRAG by 22% in accuracy, relevance, and technical detail, proving the viability of selective graph structuring.

2. Background

2.1. Naive RAG and Knowledge Graphs

Retrieval-Augmented Generation (RAG) [Lewis et al. 2020] enhances Large Language Model (LLM) outputs by grounding them in external corpora. It relies on a retriever to fetch the top-k relevant document chunks from a vector database using similarity search (e.g., via high-dimensional embeddings), and a generator that synthesizes the final answer based on this context. While effective for localized queries, standard RAG struggles to connect cross-domain knowledge scattered across multiple documents.

Knowledge Graphs (KGs) address this limitation by structuring data into entities (nodes) and relationships (edges), capturing complex dependencies [Hogan et al. 2021]. By explicitly modeling concepts and their interactions (e.g., hierarchical or causal links), KGs provide a structured representation that facilitates multistep reasoning and mitigates the information fragmentation inherent in purely vector-based retrieval.

2.2. GraphRAG and Extensions

GraphRAG merges the structured reasoning of KGs with the generative power of LLMs [Han et al. 2024, Zhang et al. 2025]. In this architecture, text chunks are augmen-

ted with KG entities and relationships extracted via LLM prompts. These graph structures, along with community-level summaries, are embedded into a vector space to enable retrieval. GraphRAG employs global search to synthesize overarching corpus trends by aggregating community insights, and local search to traverse semantic links for granular queries. This dual approach improves contextual coherence and entity resolution.

However, constructing KGs for large corpora introduces computational and practical challenges. Extracting entities, labeling relationships, and resolving nodes require repeated LLM invocations, imposing monetary and latency overheads [Min et al. 2025, Peng et al. 2024]. Furthermore, processing full-length documents often introduces noisy or redundant graph structures, which can degrade generation quality [Zhang and Soh 2024].

To ensure a comprehensive and unbiased overview of the state-of-the-art and avoid overlooking hybrid implementations, we systematically screened recent literature (2024–2026) across open science repositories and academic metadata databases, including OpenAlex and Google Scholar. Our search strategy utilized combinations of keywords such as "GraphRAG", "Knowledge Graph cost", "hybrid retrieval-augmented generation", and "LLM resource optimization". Titles and abstracts were reviewed to map out existing paradigms.

Recent extensions to GraphRAG explore hybrid approaches to optimize these systems. For instance, [Yu and McQuade 2025] proposed continuous KG updates via autonomous agents to reduce hallucinations, while [Matsumoto et al. 2024] adapted GraphRAG prompting for complex biomedical queries. Other efforts focus on leveraging document structure during retrieval [Munikoti et al. 2023] or integrating external, pre-existing KGs alongside unstructured data [Lee et al. 2025].

Unlike these methodologies, which still rely on processing full corpora for graph construction, our proposed MixRAG specifically targets operational cost optimization. By intelligently extracting the KG exclusively from document abstracts while retaining full texts in vector storage, MixRAG is, to the best of our knowledge, the first hybrid GraphRAG design that restricts KG construction to abstracts in scientific corpora while retaining full texts exclusively in vector storage, explicitly targeting extraction cost reduction rather than retrieval quality alone.

3. MixRAG

Applying GraphRAG to full academic articles is computationally prohibitive due to their length and structural complexity. To reduce costs while maintaining semantic fidelity, we propose MixRAG, a hybrid architecture that leverages both GraphRAG and standard RAG. MixRAG restricts the computationally expensive Knowledge Graph (KG) construction exclusively to article abstracts, which reliably encapsulate core contributions and key entities with minimal redundancy. Concurrently, the full-length documents are indexed using standard RAG.

Unlike standard GraphRAG, which builds a single comprehensive index from the entire corpus, MixRAG (Algorithm 1) builds a dual index. It generates a graph index, I, encompassing entities, relationships, descriptions, and communities extracted solely from abstracts. Simultaneously, a standard vector index, T, is built from the full-text chunks. Both indices explicitly map their elements to the original document identifiers.

Algorithm 1: MixRAG indexing process

Input: Corpus $D = \{d_1, d_2, \ldots, d_n\}$, Abstracts $A = \{a_1, a_2, \ldots, a_n\}$, Knowledge Base as KG
Output: Hybrid Index I, Text Index T
 # Step 1: Building the KG from abstracts;
I = Apply the GraphRAG indexing process to A;
Associate the document identifier to elements in I;
 # Step 2: Building the chunk index from full length articles;
T = Apply the standard RAG indexing to D;
Associate the document identifier to chunks in T;
return I, T

During retrieval (Algorithm 2), MixRAG preserves GraphRAG's native query types via parameter t (`global` or `local`), using the graph index I to identify relevant entities and their associated document identifiers. These identifiers dynamically filter the full-text index T, restricting the vector similarity search to only the most pertinent documents. To build the final context, we combine $k\%$ of the top-ranked elements from I and $(100 - k)\%$ from T.

Algorithm 2: MixRAG retrieval and generation process

Input: Prompt p, Hybrid index I, Text index T, percentage k of mixing from I and T, type $t \in \{$`'global'`, `'local'`$\}$
Output: Answer a

 # Step 1: Retrieval from I
Generate the embedding v_p from the prompt p ;
Retrieve relevant elements (entities, relationships, communities and descriptions) according to the type of query t;
Retrieve associated document identifiers from these elements;
 # Step 2: Retrieval from Text Index T
Filter relevant documents $R \leftarrow$ documents identifications from I ;
Relevant Chunks $K \leftarrow$ search by similarity between prompt p and chunks $c \in T$ where doc$(c) \in R$;

 # Step 3: Build final context F
$C_I \leftarrow k\%$ of top ranked elements from I;
$C_T \leftarrow (100 - k)\%$ of top ranked elements from T;
$C \leftarrow C_I \cup C_T$

 # Step 4: Generation
Send (p, C) as input to an LLM ;
Receive the answer a ;
return a

By dynamically balancing structured graph insights with context-rich text fragments via the tunable parameter k, MixRAG reduces the financial and computational overhead of full-text KG construction. This dual-index design ensures responses remain

accurate, enabling literature analysis at a fraction of the cost.

4. Evaluation

To validate MixRAG, we selected a domain-specific corpus focused on optimizing perovskite solar cells using surfactants, a critical research area in renewable energy where the literature synthesis is complex. The corpus was constructed by querying the Web of Science for "perovskite* AND surfactant* [AND solar cell*]", clustering the results via BERTopic, and filtering for the relevant "Perovskite Solar Cell" topic. After deduplication, we downloaded 95 articles and their supporting information, yielding a final corpus of 198 documents. A domain expert formulated eight questions (Table 1) exploring morphology, stability, and chemical interactions to rigorously assess response quality.

Tabela 1. Questions for evaluating the Systems

1) How do surfactants influence the morphology of the active layer in perovskite solar cells?	5) How can surfactants influence the nucleation and growth processes of perovskite crystals during active layer deposition?
2) Which chemical groups appear most frequently in the molecular structure of surfactants applied to perovskite solutions and which reflect good efficiency and stability results in solar cells?	6) What are the challenges and limitations of using surfactants in perovskite solar cells?
3) How do surfactants affect the stability of perovskite solar cells under different environmental conditions (humidity, light, temperature)?	7) What influence does the size of hydrophobic chains in the structure of surfactants have on the structural quality of perovskite films?
4) What analytical techniques are used to evaluate the impact of surfactants on the morphological and electrochemical properties of perovskite films?	8) What are the ideal surfactant concentration ranges for application in perovskite solutions? How does the quantity of these surfactants affect the quality of films?

We implemented MixRAG as an extension of Microsoft™ GraphRAG (v0.5.0). To balance structured and unstructured knowledge, the mixing parameter was empirically set to $k = 40$ (40% elements from the abstract-based KG and 60% full-text chunks). This baseline was selected to prevent the dense, high-level graph insights from overshadowing the granular experimental details found in the broader text. While an exhaustive ablation study to assess the impact of varying k was beyond the scope of this initial validation, its dynamic optimization is slated for future work. Both systems utilized *gpt-4o-mini* for generation and *text-embedding-3-small* for embeddings, employing cosine similarity for vector retrieval.

4.1. Cost Evaluation

Table 2 compares the construction costs of processing the entire corpus with GraphRAG versus processing only the 95 abstracts with MixRAG. MixRAG achieved a 98% real cost reduction, aligning closely with the 97% estimate. The variance between estimated and real costs (40% for GraphRAG, 21% for MixRAG) reflects the exponential increase in LLM API calls required to resolve entities in the full-text corpus.

4.2. Performance Evaluation

Initial self-judgment evaluation [Gu et al. 2024] using *gpt-4o-mini* proved inconclusive, as the model assigned maximum scores to all responses. Consequently, we conducted a

Tabela 2. Cost comparison between MixRAG and GraphRAG

			Cost (US$)		
	Documents	Words	Estimated[a]	Real	Variation (%)
GraphRAG	168[b]	640,590	7.24	11.99	40%
MixRAG	95[c]	17,170	0.19	0.24	21%
Overall Reduction (%)			**97%**	**98%**	

[a]Estimated at $0.0000113/word (based on Microsoft guidelines).

[b]95 Papers plus Supporting Information.

[c]Used only the 95 papers, as supporting information lacks abstracts.

blind quantitative and qualitative assessment by a domain expert. Responses were scored from 1 to 5 across four criteria, defined in Table 3.

Tabela 3. Evaluation Criteria and Scoring Scale

Criterion	Definition
Accuracy	Is the answer based on the content provided?
Relevance	From a technical point of view, is the information provided relevant?
Correction	Is the answer free of conceptual errors and incompatible terms?
Details	How explanatory is the answer on the technical aspects?
Scores: 1-2 (Weakly met); 3 (Moderately met); 4-5 (Almost or completely met)	

Table 4 details the expert's scoring. MixRAG outperformed GraphRAG across all criteria, achieving a 22% higher total score (153 vs. 125). We attribute this to MixRAG's hybrid design, which reduces noise by isolating entities in the abstracts while dynamically filtering the vector database for precision. Validating this hypothesis, the expert's blind qualitative assessment concluded that MixRAG *"delivered the most coherence and depth on the topic."*

Tabela 4. Specialist Judgment comparison between MixRAG and GraphRAG

Answer	Accuracy		Relevance		Correction		Details		Total	
	Graph	Mix	Graph	Mix	Graph	Mix	Graph	Mix	Graph	Mix
1	4	**5**	3	**5**	**5**	**5**	3	**5**	15	**20**
2	**4**	3	**3**	**3**	**5**	**5**	2	**4**	14	**15**
3	4	**5**	3	**5**	**5**	**5**	2	**5**	14	**20**
4	5	**5**	3	**5**	**5**	**5**	3	**5**	16	**20**
5	4	**5**	3	**4**	**5**	**5**	3	**4**	15	**18**
6	5	**5**	4	**5**	**5**	**5**	4	**5**	18	**20**
7	5	**5**	4	**5**	**5**	**5**	3	**5**	17	**20**
8	4	**5**	4	**5**	**5**	**5**	3	**5**	16	**20**
Total	35	**38**	27	**37**	**40**	**40**	23	**38**	125	**153**

5. Conclusion and Future Work

This paper proposes MixRAG, a hybrid retrieval-augmented generation framework designed to address the high computational costs of applying GraphRAG to scientific do-

cuments. Constructing KGs from full-text articles incurs significant resource overhead, particularly when dealing with lengthy and semantically dense academic literature. MixRAG mitigates this by strategically balancing structured graph data (derived exclusively from abstracts) with unstructured text fragments (from full articles). This approach reduces KG construction costs by over 95% while maintaining and even improving response quality.

Our evaluation demonstrates that, within our materials science case study, MixRAG outperformed GraphRAG in answer quality, as assessed by a domain expert, at a fraction of the cost. This improvement stems from its dual-index design: the abstract-derived KG captures high-level relationships and core findings, while the full-text vector index retains granular experimental details often omitted in summaries. These results highlight MixRAG's potential as a cost-effective, high-precision tool for comprehensive scientific literature analysis.

Although this preliminary work is constrained by a single-domain pilot evaluation, it establishes a robust baseline for cost-efficient hybrid retrieval. Future efforts will focus on expanding this framework into cross-domain benchmarks with a larger expert panel for broader statistical validation, optimizing the parameter k, and exploring the impact of different LLMs and embedding models on the overall efficiency of the hybrid retrieval strategy.

Referências

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *NIPS'2020*, 33:1877–1901.

Dahl, M., Magesh, V., Suzgun, M., and Ho, D. E. (2024). Large legal fictions: Profiling legal hallucinations in large language models. *J. Leg. Anal,*, 16(1):64–93.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., et al. (2024). A survey on llm-as-a-judge. *The Innovation*.

Guo, Z., Xia, L., Yu, Y., Ao, T., and Huang, C. (2024). Lightrag: Simple and fast retrieval-augmented generation. *ArXiv*, abs/2410.05779.

Han, H., Shomer, H., Wang, Y., Lei, Y., Guo, K., Hua, Z., Long, B., Liu, H., and Tang, J. (2025). Rag vs. graphrag: A systematic evaluation and key insights. *ArXiv*, abs/2502.11371.

Han, H., Wang, Y., Shomer, H., Guo, K., Ding, J., Lei, Y., Halappanavar, M., Rossi, R. A., Mukherjee, S., Tang, X., He, Q., Hua, Z., Long, B., Zhao, T., Shah, N., Javari, A., Xia, Y., and Tang, J. (2024). Retrieval-augmented generation with graphs (graphrag). *ArXiv*, abs/2501.00309.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. D., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., et al. (2021). Knowledge graphs. *ACM Comput Surv.*, 54(4):1–37.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., and Liu, T. (2023). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43:1 – 55.

Lee, M.-C., Zhu, Q., Mavromatis, C., Han, Z., Adeshina, S., Ioannidis, V. N., Rangwala, H., and Faloutsos, C. (2025). Hybgrag: Hybrid retrieval-augmented generation on textual and relational knowledge bases. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 879–893. Association for Computational Linguistics.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NIPS'2020*, volume 33.

Matsumoto, N., Moran, J., Choi, H., Hernandez, M. E., Venkatesan, M., Wang, P., and Moore, J. H. (2024). Kragen: a knowledge graph-enhanced rag framework for biomedical problem solving using large language models. *Bioinformatics*, 40(6):btae353.

Min, C., Bansal, S., Pan, J., Keshavarzi, A., Mathew, R., and Kannan, A. V. (2025). Towards practical graphrag: Efficient knowledge graph construction and hybrid retrieval at scale. *arXiv preprint arXiv:2507.03226*.

Munikoti, S., Acharya, A., Wagle, S., and Horawalavithana, S. (2023). Atlantic: Structure-aware retrieval-augmented language model for interdisciplinary science. *arXiv preprint arXiv:2311.12289*.

Peng, B., Zhu, Y., Liu, Y., Bo, X., Shi, H., Hong, C., Zhang, Y., and Tang, S. (2024). Graph retrieval-augmented generation: A survey. *ACM Trans. Inf. Syst.*

Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., and Chen, E. (2023). A survey on multimodal large language models. *Natl. Sci. Rev.*, 11.

Yu, H. Q. and McQuade, F. (2025). RAG-KG-IL: A multi-agent hybrid framework for reducing hallucinations and enhancing llm reasoning through rag and incremental knowledge graph learning integration. *arXiv preprint arXiv:2503.13514*.

Zhang, B. and Soh, H. (2024). Extract, define, canonicalize: An llm-based framework for knowledge graph construction. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 9820–9836.

Zhang, Q., Chen, S., Bei, Y.-Q., Yuan, Z., Zhou, H., Hong, Z., Dong, J., Chen, H., Chang, Y., and Huang, X. (2025). A survey of graph retrieval-augmented generation for customized large language models. *ArXiv*, abs/2501.13958.

Zhuang, L., Chen, S., Xiao, Y., Zhou, H., Zhang, Y., Chen, H., Zhang, Q., and Huang, X. (2025). Linearrag: Linear graph retrieval augmented generation on large-scale corpora. *ArXiv*, abs/2510.10114.