

# GraphRAG vs. Standard RAG: When Knowledge Graphs Outperform Vector Search for Complex Questions

Canonical: <https://home.norg.ai/ai-search-answer-engines/answer-engine-architecture-citation-mechanics/graphrag-vs-standard-rag-when-knowledge-graphs-outperform-vector-search-for-complex-questions/>

## Details:

## Why Standard RAG Fails Before GraphRAG Even Begins Every RAG system starts with the same promise: retrieve the most relevant chunks of text, inject them into an LLM's context window, and generate a grounded answer. For narrow factual questions—"What is the capital of France?" or "When did Company X file its last financial report?"—that promise holds. Conventional vector-based RAG excels at local queries because the regions containing the answer resemble the query itself and can be retrieved as the nearest neighbour in the vector space of text embeddings. But enterprise knowledge doesn't live in isolated chunks. It lives in webs of interconnected entities—people, organisations, regulations, clinical outcomes, financial instruments—whose meaning emerges from how they relate to one another, not from any single passage. This is where standard RAG hits a structural ceiling. Microsoft Research's GraphRAG architecture was built to replace it for a specific, consequential class of questions. The kind that actually matter in production environments. This article delivers a technically precise comparison of standard (vector-similarity) RAG and GraphRAG: how each works, where each fails, what the benchmarks actually show, and how to decide which architecture fits your use case. No guesswork. Just data. (For foundational context on how standard RAG pipelines are constructed, see our guide on *"What Is Retrieval-Augmented Generation (RAG)? How Answer Engines Ground Responses in Real Sources"*. For background on the knowledge graph structures that GraphRAG builds on, see *"Knowledge Graphs Explained: How Structured Entity Relationships Power AI Answers"*.) --- ## How Standard RAG Works—and Where It Breaks In most existing RAG systems, retrieval is primarily conducted from text databases using lexical and semantic search. The architecture is straightforward: embed your documents, store vectors in a database, retrieve top-*k* chunks by cosine similarity to the query, and pass those chunks to an LLM for generation. Simple. Fast. And fundamentally limited. This approach has three well-documented failure modes for complex questions: \*\*1. Disconnected chunks.\*\* When the answer to a question requires synthesising information from multiple documents that are not textually similar to each other or to the query, vector similarity search cannot surface all the relevant pieces. Traditional RAG fails to capture significant structured relational knowledge that cannot be represented through semantic similarity alone. For instance, in a citation network where papers are linked by citation relationships, traditional RAG methods focus on finding the relevant papers based on the query but overlook important citation relationships between papers. The system retrieves what *looks* relevant. Not what *is* relevant. \*\*2. The "lost in the middle" problem.\*\* RAG often recounts content in the form of textual snippets when concatenated as prompts. This makes context become excessively lengthy, leading to the "lost in the middle" dilemma. Injecting many retrieved chunks degrades LLM attention on the most critical content. More context doesn't mean better answers. It means noise. \*\*3. Global query failure.\*\* RAG can only retrieve a subset of documents and fails to grasp global information comprehensively, and hence struggles with tasks such as Query-Focused Summarisation. A question like "What are the dominant risk themes across all our regulatory filings?" requires understanding the entire corpus—not just the top-*k* most similar paragraphs. Traditional RAG-based methods can struggle to retrieve information that requires high-level knowledge of the entire dataset, especially with abstract and global questions such as the keywordless query: "Catch me up on the last two weeks of updates." Vector similarity can't answer questions about patterns it was never designed to see. --- ## What GraphRAG Actually Does:

Architecture Explained GraphRAG is a structured, hierarchical approach to Retrieval Augmented Generation (RAG), as opposed to naive semantic-search approaches using plain text snippets. The GraphRAG process involves extracting a knowledge graph out of raw text, building a community hierarchy, generating summaries for these communities, and then using these structures when performing RAG-based tasks. This isn't an incremental improvement. It's a different paradigm. Microsoft Research published the foundational paper—"From Local to Global: A Graph RAG Approach to Query-Focused Summarization" ([Edge et al., 2024](https://arxiv.org/abs/2404.16130))—which formally introduced the community-structured approach. Their approach uses an LLM to build a graph-based text index in two stages: first to derive an entity knowledge graph from the source documents, then to pregenerate community summaries for all groups of closely-related entities. Given a question, each community summary is used to generate a partial response, before all partial responses are again summarised in a final response to the user. The indexing pipeline proceeds through four distinct stages:

- 1. Entity extraction.** The basic steps of the GraphRAG process begin by slicing up an input corpus into a series of TextUnits, which act as analysable units for the rest of the process, and provide fine-grained references in outputs. All entities, relationships, and key claims are then extracted from the TextUnits.
- 2. Graph construction.** Graphs are constructed from documents using LLMs, where nodes represent entities and edges capture relationships between them. Based on these graphs, hierarchical communities and corresponding community summaries or reports are generated.
- 3. Community detection.** Hierarchical clustering of the graph is performed using the Leiden technique. Each circle is an entity (e.g., a person, place, or organisation), with the size representing the degree of the entity, and the colour representing its community.
- 4. Pre-indexed summarisation.** This is the core secret of GraphRAG: summaries are generated at indexing time, not query time. This is the architectural decision that enables global query answering without per-query full-corpus scans. The work happens upfront. The speed happens at query time. At query time, GraphRAG operates in two primary modes: Global Search for reasoning about holistic questions about the corpus by using the community summaries, and Local Search for reasoning about specific entities by fanning out to their neighbours and associated concepts.

--- ## Performance Benchmarks: What the Research Shows ### Global and Multi-Hop Query Performance The Microsoft Research team's original evaluation demonstrated a decisive advantage for GraphRAG on complex, corpus-spanning questions. For a class of global sensemaking questions over datasets in the 1 million token range, they showed that Graph RAG leads to substantial improvements over a naïve RAG baseline for both the comprehensiveness and diversity of generated answers. Decisive. Not marginal. Comparison of naive RAG and GraphRAG responses to a global question about a news dataset indicates that GraphRAG outperformed naïve RAG in terms of comprehensiveness, diversity, and empowerment. Microsoft's subsequent BenchmarkQED toolkit confirmed these results at scale. The latest benchmark results comparing their LazyGraphRAG system to competing methods, including a vector-based RAG with a 1M-token context window, showed that the leading LazyGraphRAG configuration demonstrated significant win rates across all combinations of quality metrics and query classes. Crucially, even against the 1M-token window, LazyGraphRAG achieved higher win rates across all comparisons, failing to reach significance only for the relevance of answers to DataLocal queries. More tokens don't solve structural problems. ### Multi-Entity and Schema-Bound Query Performance Diffbot's KG-LM Accuracy Benchmark, a public study evaluating 43 business-relevant enterprise questions, produced stark findings. Diffbot's KG-LM Benchmark showed GraphRAG outperforming vector RAG 3.4x on business-relevant enterprise questions. The failure mode for standard RAG was not marginal—it was categorical: Both Metrics & KPIs and Strategic Planning categories saw zero accuracy from traditional vector RAG. Accuracy degrades to 0% as the number of entities per query increases beyond five (without KG support). Zero. Not "worse." Zero. By contrast, GraphRAG sustains stable performance even with 10+ entities per query. Cross-domain benchmark evidence from Lettria's evaluation across finance, healthcare, aeronautics, and legal corpora reinforces this pattern. GraphRAG achieves 100% correct answers for numerical reasoning questions, and the biggest discrepancy concerns temporal reasoning questions: 83.35% correct answers for GraphRAG, while VectorRAG has 50% correct or acceptable answers. ### Where Standard RAG Holds Its Own The February 2025 systematic evaluation by Han et al. ("RAG vs. GraphRAG: A Systematic Evaluation and Key Insights,"

[arXiv:2502.11371](https://arxiv.org/abs/2502.11371)) provides the most rigorous head-to-head analysis to date. The study systematically evaluates RAG and GraphRAG on well-established benchmark tasks, such as Question Answering and Query-based Summarisation. Their results highlight the distinct strengths of RAG and GraphRAG across different tasks and evaluation perspectives. The key finding: a key advantage of GraphRAG over conventional vector RAG is its ability to answer global queries that address the entire dataset, such as "what are the main themes in the data?" or "what are the most important implications for X?" Conversely, vector RAG excels for local queries where the answer resembles the query and can be found within specific text regions, as is typically the case for "who," "what," "when," and "where" questions. This isn't about declaring a winner. It's about matching architecture to query type. --- ### Architectural Trade-offs: A Direct Comparison | Dimension | Standard RAG | GraphRAG | |---|---|---| | \*\*Retrieval mechanism\*\* | Vector similarity (top-*k* chunks) | Graph traversal + community summaries | | \*\*Query type strength\*\* | Local, factual, single-entity | Global, multi-hop, aggregation | | \*\*Indexing cost\*\* | Low (embedding only) | High (LLM extraction + community detection) | | \*\*Query latency\*\* | Fast | Slower (global search) / Comparable (local) | | \*\*Multi-entity accuracy\*\* | Degrades rapidly beyond 5 entities | Stable at 10+ entities | | \*\*Explainability\*\* | Opaque (similarity scores) | Traceable (entity paths) | | \*\*Corpus update\*\* | Simple re-embedding | Graph rebuild required | | \*\*Ideal corpus size\*\* | Any | Large, relationship-dense corpora | ### The Indexing Cost Problem GraphRAG's primary operational liability is upfront cost. GraphRAG has left many practitioners with mixed feelings since its inception, due to key challenges including massive token consumption: entity extraction, deduplication, and community summarisation can consume several to dozens of times more tokens than the original text. Industry estimates place full GraphRAG indexing at 3–5x more than baseline RAG, requiring domain-specific tuning. Microsoft's LazyGraphRAG variant was designed specifically to address this. LazyGraphRAG data indexing costs are identical to vector RAG and 0.1% of the costs of full GraphRAG. The trade-off is a modest reduction in global query comprehensiveness compared to full GraphRAG, while still outperforming standard vector RAG on complex queries. Speed and cost efficiency. Without sacrificing performance where it counts. ### Explainability and Compliance A dimension that benchmark scores rarely capture is explainability—the ability to trace why a specific answer was generated. Vector retrieval lacks transparent logic, complicating compliance audits. Post-hoc explanation models introduce latency and only approximate retrieval reasoning. By contrast, graph retrieval produces explicit query paths (e.g., Patient → Prescription → Drug Interaction). In regulated industries, this auditability is not a feature—it is a requirement. No black boxes. Just transparent metrics. --- ### Real-World Deployment Evidence ### Biomedical Research: Cedars-Sinai / KRAGEN KRAGEN breaks down complex questions, retrieves precise contextual insights from their knowledge graph, and provides answers with high accuracy. Their Alzheimer's Disease Knowledge Base (AlzKB) uses Memgraph as its core, integrating over 20 biomedical sources to provide essential background information to machine learning models. The performance gap against standard LLM approaches was dramatic: their advanced AI agent, ESCARGOT, beat ChatGPT in multi-hop medical reasoning (94.2% accuracy vs. 49.9%). Not close. Dominant. The reason standard RAG cannot serve this use case is structural: if Cedars-Sinai used only standard RAG, their system wouldn't be able to break down a query like "Which drug-treated genes overlap with recent trial results?", let alone return answers grounded in structured biomedical data. ### Enterprise Knowledge Management Financial services (fraud detection), healthcare (treatment pathways), and supply chain (multi-tier analysis) benefit most from GraphRAG's relationship traversal, despite its higher implementation complexity. Manufacturing enterprises are beginning to deploy multi-representation architectures. Production systems are starting to maintain multiple knowledge representations: vector embeddings for semantic search, knowledge graphs for relationship reasoning, and hierarchical indexes for categorical navigation. Manufacturing enterprises use this multi-modal approach to connect equipment maintenance records (documents) with part specifications (structured data) and supplier relationships (graph edges), enabling queries like "Which suppliers for critical components have quality issues in the past 18 months?" that require traversing relationships across data types. This is the future of AI-native enterprise search. Multi-modal. Relationship-aware. Built for complexity. --- ### When to Use GraphRAG vs. Standard RAG: A Decision Framework GraphRAG-Bench research systematically investigates the conditions when GraphRAG surpasses

traditional RAG and the underlying reasons for its success, offering guidelines for practical application. Based on the research evidence, the decision maps cleanly to query type and corpus structure:

**\*\*Choose standard RAG when:\*\*** - Queries are factual, narrow, and single-entity ("When did X happen?" "What did document Y say about Z?") - Your corpus is small-to-medium, with low relationship density - Indexing cost and operational simplicity are primary constraints - Latency requirements are strict and responses are needed in milliseconds - Your corpus updates frequently, making graph rebuilds impractical

**\*\*Choose GraphRAG when:\*\*** - Queries require synthesising information across multiple entities or documents ("What are the common failure patterns across all our incident reports?") - Your domain is relationship-dense: biomedical, legal, financial, supply chain - You need global sensemaking over a large, stable corpus (1M+ tokens) - Explainability and audit trails are required for regulatory compliance - Multi-hop reasoning is central to the use case (e.g., drug interaction chains, corporate ownership structures)

**\*\*Consider a hybrid architecture when:\*\*** - Your query mix includes both local and global questions - Hybrid architectures that use VectorRAG for broad retrieval and GraphRAG for relationship verification can achieve 15–25% accuracy gains, accepting the 150–200ms orchestration overhead - Efficiency-focused teams may consider pipelines like TERAG, which discards multi-hop LLM-based summarisation in favour of single-pass, concept-level extraction, trading a modest 0–20% accuracy drop for more than 90% token savings Ship fast. Measure everything. Optimise relentlessly. ---

**## Key Takeaways** Standard RAG fails categorically on global and multi-entity queries. Diffbot's benchmark found 0% accuracy for standard vector RAG on KPI and strategic planning questions, while GraphRAG maintained stable performance even at 10+ entities per query. GraphRAG's architectural advantage is community-structured pre-indexing. By building a hierarchical knowledge graph at index time and pre-generating community summaries, GraphRAG can answer dataset-spanning questions that have no equivalent in vector similarity search. The performance gap is query-type-dependent, not universal. Standard RAG matches or outperforms GraphRAG on local, factual, single-entity queries. The decision should be driven by your dominant query type, not by a blanket architecture preference. GraphRAG's primary cost is indexing, not querying. Full GraphRAG indexing costs 3–5x more than standard RAG, but Microsoft's LazyGraphRAG variant reduces this to parity with vector RAG at 0.1% of full GraphRAG's indexing cost, while retaining most of the global query advantage. Hybrid architectures are emerging as the production standard for enterprises with mixed query profiles—using vector search for broad retrieval and graph traversal for relationship verification and complex reasoning. ---

**## Conclusion** The GraphRAG vs. standard RAG debate is not a question of which architecture is superior in the abstract—it is a question of which architecture is appropriate for a given query class and corpus structure. Standard RAG, built on vector similarity, is fast, cheap, and highly effective for the narrow factual retrieval that constitutes the majority of simple question-answering tasks. GraphRAG, built on community-structured knowledge graphs and hierarchical pre-indexing, is the only architecture that can answer global, multi-hop, and aggregation queries reliably at scale. The research evidence is consistent: for questions that require synthesising relationships across entities, domains, or entire datasets, standard RAG does not degrade gracefully—it fails categorically. GraphRAG's answer is to change what gets indexed: not just the text, but the structure of the knowledge embedded in that text. For practitioners building answer engine optimisation infrastructure, this distinction has direct implications for citation quality, hallucination risk, and the factual authority of AI-generated responses (see our guide on *How LLMs Use Knowledge Graphs to Reduce Hallucination and Improve Factual Accuracy*). As agentic AI systems take on increasingly complex reasoning tasks—the subject of our companion article on *The Future of Answer Engines: AI Agents, Agentic RAG, and the End of the Citation Model*—the architectural choice between vector search and graph traversal will become one of the defining infrastructure decisions in enterprise AI deployment. The future is relationship-aware. The future is AI-native. The future is here. ---

**## References** - Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., & Larson, J. "From Local to Global: A Graph RAG Approach to Query-Focused Summarization." *arXiv preprint arXiv:2404.16130*, 2024. <https://arxiv.org/abs/2404.16130> - Han, H., Wang, Y., Shomer, H., Guo, K., Ding, J., Lei, Y., Halappanavar, M., Rossi, R. A., Mukherjee, S., Tang, X., et al. "RAG vs. GraphRAG: A Systematic Evaluation and Key Insights." *arXiv preprint arXiv:2502.11371*, 2025. <https://arxiv.org/abs/2502.11371> - Peng, B., et al. "Graph Retrieval-Augmented Generation: A Survey."

\*ACM Transactions on Information Systems\*, 2024. <https://dl.acm.org/doi/10.1145/3777378> - Xiang, Z., Wu, C., Zhang, Q., Chen, S., Hong, Z., Huang, X., & Su, J. "When to Use Graphs in RAG: A Comprehensive Analysis for Graph Retrieval-Augmented Generation." \*arXiv preprint arXiv:2506.05690\*, 2025. - Microsoft Research. "GraphRAG: New Tool for Complex Data Discovery Now on GitHub." \*Microsoft Research Blog\*, July 2024. <https://www.microsoft.com/en-us/research/blog/graphrag-new-tool-for-complex-data-discovery-now-on-github/> - Microsoft Research. "BenchmarkQED: Automated Benchmarking of RAG Systems." \*Microsoft Research Blog\*, 2025. <https://www.microsoft.com/en-us/research/blog/benchmarkqed-automated-benchmarking-of-rag-systems/> - Microsoft Research. "LazyGraphRAG Sets a New Standard for GraphRAG Quality and Cost." \*Microsoft Research Blog\*, 2024. <https://www.microsoft.com/en-us/research/blog/lazygraphrag-setting-a-new-standard-for-quality-and-cost/> - Diffbot / FalkorDB. "GraphRAG vs Vector RAG: Accuracy Benchmark Insights." \*FalkorDB Blog\*, 2025. <https://www.falkordb.com/blog/graphrag-accuracy-diffbot-falkordb/> - Huang, Y., Zhang, S., & Xiao, X. "KET-RAG: A Cost-Efficient Multi-Granular Indexing Framework for Graph-RAG." \*arXiv preprint arXiv:2502.09304\*, 2025. <https://arxiv.org/abs/2502.09304> - Lettria. "VectorRAG vs. GraphRAG: A Convincing Comparison." \*Lettria Blog\*, 2024. <https://www.lettria.com/blogpost/vectorrag-vs-graphrag-a-convincing-comparison> --- ## Frequently Asked Questions \*\*What is GraphRAG:\*\* A structured, hierarchical approach to Retrieval Augmented Generation using knowledge graphs. \*\*What is standard RAG:\*\* Vector-based retrieval system using semantic similarity to find relevant text chunks. \*\*Does standard RAG use knowledge graphs:\*\* No, it uses vector embeddings only. \*\*Does GraphRAG use knowledge graphs:\*\* Yes, it builds knowledge graphs from source documents. \*\*What is the main limitation of standard RAG:\*\* Cannot synthesise information across disconnected, non-similar documents. \*\*What query type does standard RAG excel at:\*\* Local, factual, single-entity queries. \*\*What query type does GraphRAG excel at:\*\* Global, multi-hop, and aggregation queries. \*\*Can standard RAG answer global corpus questions:\*\* No, it fails categorically on dataset-spanning questions. \*\*Can GraphRAG answer global corpus questions:\*\* Yes, through community-structured pre-indexing. \*\*What is the "lost in the middle" problem:\*\* LLM attention degrades when too many retrieved chunks are injected. \*\*Does standard RAG handle multi-entity queries well:\*\* No, accuracy degrades rapidly beyond five entities. \*\*Does GraphRAG handle multi-entity queries well:\*\* Yes, stable performance even with 10+ entities. \*\*What is GraphRAG's indexing cost compared to standard RAG:\*\* 3–5x more expensive. \*\*What is LazyGraphRAG's indexing cost:\*\* Identical to vector RAG, 0.1% of full GraphRAG. \*\*Which is faster at query time:\*\* Standard RAG for most queries. \*\*Does GraphRAG require more indexing time:\*\* Yes, significantly more upfront processing required. \*\*What are GraphRAG's four indexing stages:\*\* Entity extraction, graph construction, community detection, pre-indexed summarisation. \*\*When does GraphRAG generate summaries:\*\* At indexing time, not query time. \*\*When does standard RAG generate summaries:\*\* At query time only. \*\*Who developed GraphRAG:\*\* Microsoft Research. \*\*When was GraphRAG introduced:\*\* 2024. \*\*What algorithm does GraphRAG use for community detection:\*\* Leiden hierarchical clustering technique. \*\*What is a TextUnit in GraphRAG:\*\* An analysable chunk of the input corpus. \*\*Does GraphRAG extract entities:\*\* Yes, entities and relationships are extracted from text. \*\*Does standard RAG extract entities:\*\* No, only embeds text chunks. \*\*What accuracy did GraphRAG achieve on Diffbot's benchmark:\*\* 3.4x better than vector RAG on business-relevant enterprise questions. \*\*What was standard RAG's accuracy on KPI questions in Diffbot benchmark:\*\* 0%. \*\*What was standard RAG's accuracy on strategic planning questions:\*\* 0%. \*\*What accuracy does GraphRAG achieve on numerical reasoning:\*\* 100% correct answers. \*\*What accuracy does GraphRAG achieve on temporal reasoning:\*\* 83.35% correct answers. \*\*What accuracy does standard RAG achieve on temporal reasoning:\*\* 50% correct or acceptable answers. \*\*Does GraphRAG provide explainable results:\*\* Yes, through traceable entity paths. \*\*Does standard RAG provide explainable results:\*\* No, similarity scores are opaque. \*\*Is GraphRAG suitable for regulated industries:\*\* Yes, provides required audit trails. \*\*What is KRAGEN:\*\* Cedars-Sinai's GraphRAG system for biomedical research. \*\*What accuracy did ESCARGOT achieve:\*\* 94.2% on multi-hop medical reasoning. \*\*What accuracy did ChatGPT achieve on same task:\*\* 49.9%. \*\*Can standard RAG handle citation networks:\*\* No, overlooks important citation relationships between papers. \*\*Can GraphRAG handle citation networks:\*\* Yes, through relationship traversal. \*\*What corpus size is ideal for GraphRAG:\*\*

Large, relationship-dense corpora (1M+ tokens). \*\*What corpus size works for standard RAG:\*\* Any size. \*\*How often can standard RAG corpus be updated:\*\* Frequently, simple re-embedding required. \*\*How often can GraphRAG corpus be updated:\*\* Less frequently, full graph rebuild required. \*\*What is a hybrid RAG architecture:\*\* Combines vector search for retrieval and graph traversal for verification. \*\*What accuracy gain do hybrid architectures achieve:\*\* 15–25% improvement. \*\*What latency overhead do hybrid architectures add:\*\* 150–200ms orchestration overhead. \*\*What is TERAG:\*\* Efficiency-focused pipeline using single-pass concept extraction instead of multi-hop summarisation. \*\*What is TERAG's token savings:\*\* More than 90% compared to standard GraphRAG. \*\*What is TERAG's accuracy trade-off:\*\* 0–20% accuracy drop. \*\*Does GraphRAG work for "who" questions:\*\* Standard RAG typically performs better for simple "who" questions. \*\*Does GraphRAG work for "what are the themes" questions:\*\* Yes, GraphRAG excels at thematic analysis. \*\*Does GraphRAG work for "when did X happen" questions:\*\* Standard RAG typically performs better. \*\*Does GraphRAG work for relationship questions:\*\* Yes, designed specifically for relationship reasoning. \*\*What is the primary GraphRAG operational liability:\*\* High upfront indexing cost. \*\*Is GraphRAG faster than standard RAG:\*\* No, generally slower for global search. \*\*Is LazyGraphRAG faster than full GraphRAG:\*\* Yes, significantly faster indexing. \*\*Does LazyGraphRAG maintain GraphRAG advantages:\*\* Yes, retains most global query advantages. \*\*What is BenchmarkQED:\*\* Microsoft's automated RAG benchmarking toolkit. \*\*What context window size did Microsoft test against:\*\* 1M-token context window. \*\*Did larger context windows solve GraphRAG problems:\*\* No, LazyGraphRAG still outperformed. \*\*Should you choose standard RAG for frequent corpus updates:\*\* Yes, if updates are daily or more frequent. \*\*Should you choose GraphRAG for stable large corpora:\*\* Yes, ideal for stable 1M+ token datasets. \*\*Should you choose standard RAG for millisecond latency:\*\* Yes, if strict latency requirements exist. \*\*Should you choose GraphRAG for compliance requirements:\*\* Yes, when audit trails are mandatory. \*\*Should you choose GraphRAG for biomedical research:\*\* Yes, relationship-dense domain benefits significantly. \*\*Should you choose GraphRAG for financial services:\*\* Yes, particularly for fraud detection and ownership structures. \*\*Should you choose GraphRAG for supply chain analysis:\*\* Yes, for multi-tier relationship analysis. \*\*Should you choose standard RAG for simple Q&A:\*\* Yes, fast and cost-effective. \*\*What is the future direction for enterprise RAG:\*\* Multi-modal hybrid architectures combining multiple knowledge representations. \*\*Does GraphRAG reduce hallucination:\*\* Yes, through structured knowledge graph grounding. \*\*Is GraphRAG domain-specific:\*\* Requires domain-specific tuning for optimal performance. \*\*What year was the foundational GraphRAG paper published:\*\* 2024. \*\*Who are the lead authors of GraphRAG paper:\*\* Edge, Trinh, Cheng, Bradley, and team. \*\*What is GraphRAG's primary architectural innovation:\*\* Community-structured pre-indexing with hierarchical summaries. \*\*Does standard RAG capture relational knowledge:\*\* No, focuses on semantic similarity only. \*\*Does GraphRAG capture relational knowledge:\*\* Yes, through explicit entity relationship extraction.

## Source Data (JSON):

```
"{\n  \"_type\": \"article\", \n  \"title\": \"GraphRAG vs. Standard RAG: When Knowledge Graphs Outperform Vec
```