The State of Retrieval-Augmented Generation (RAG): An Archival Review of Foundational Papers, Advanced Architectures, and Systemic Evaluation

I. Foundational Principles of Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) is a critical architectural paradigm that has profoundly shifted the landscape of Large Language Model (LLM) deployment, particularly in knowledge-intensive applications. Fundamentally, RAG enhances generative LLMs by dynamically integrating information retrieval techniques, compelling the model to consult a specified external knowledge base before formulating a response.[1] This process effectively merges the linguistic coherence derived from the LLM's static, parametric knowledge (weights) with verifiable, dynamic, non-parametric knowledge stored externally.[3]

The seminal work defining this field was published in 2020 by Lewis et al., which introduced the term RAG.[2] The original framework proposed two distinct generative models that wrestled with the complexity of integrating discrete retrieval decisions into the continuous, autoregressive generation process.[4] The **RAG-Token** model was designed as a simpler, standard autoregressive sequence-to-sequence generator where the retrieval influences the transition probability, $p_{\theta}(y_i|x, y_{1:i-1})$, allowing for the use of a conventional beam decoder. The **RAG-Sequence** model, however, aimed to calculate the marginal likelihood, $p(y|x)$, over all retrieved documents z. Since $p(y|x)$ does not decompose into a conventional per-token likelihood, this necessitated a computationally intensive approach known as "Thorough Decoding," where beam search was run for *each* retrieved document z.[4] This early architectural challenge, particularly the complexity of decoding and scoring in RAG-Sequence, foreshadowed the later need for sophisticated reranking and modular orchestration methods to better integrate and prioritize retrieved knowledge.[5] The initial RAG implementation used a substantial knowledge source: a single December 2018 Wikipedia

dump, segmented into 21 million disjoint 100-word chunks, indexed via FAISS utilizing a Hierarchical Navigable Small World approximation for fast retrieval.[4]

RAG's Role in Mitigating Hallucination and Managing Knowledge Cut-off

RAG's primary function is to address the core deficit of standalone LLMs: the tendency to produce responses that are plausible but ungrounded in facts, commonly referred to as hallucination.[1] This factual grounding is particularly vital in high-stakes domains, such such as legal or medical queries, where describing non-existent policies or recommending non-existent legal cases carries severe consequences.[2]

Beyond mitigating factual errors, RAG provides significant operational benefits, particularly concerning knowledge management and cost efficiency. RAG allows LLMs to access dynamic, up-to-date, and domain-specific information that was not available in their static training data.[2] This is crucial for fast-evolving industries; RAG models have demonstrated the ability to reduce outdated responses by 15 to 20 percent compared to traditional LLMs, emphasizing its necessity in environments with high data dynamism.[8] Furthermore, RAG presents a critical economic leverage point by eliminating the need to retrain massive LLMs whenever the knowledge base requires updating, resulting in substantial savings on computational and financial costs.[2] This capability allows smaller, open-source models augmented with RAG to achieve competitive performance against much larger, proprietary models, democratizing advanced AI deployment for organizations with constrained resources.[8]

II. The Taxonomy of RAG Architectures and Flow Dynamics

The maturity of RAG has necessitated an evolution from rigid pipelines to flexible, composable architectures. This maturation is characterized by a progression from simple to modular systems.[9]

Evolution from Naive RAG to Modular Systems

The earliest implementation, **Naive RAG**, involved a single, straightforward pass of retrieval followed by generation. This quickly evolved into **Advanced RAG** systems, which began employing optimized retrieval strategies and incorporating structured data use. The current state-of-the-art is encapsulated by **Modular RAG**, which defines a highly flexible, task-specific paradigm that permits independent control and dynamic orchestration of system components.[6]

The Three-Tiered Structure of Modular RAG

Modular RAG provides a unified structural approach, organizing the RAG system into a clear, hierarchical design [6]:

1. **Modules (Top Tier):** These represent the high-level, orchestratable stages of the pipeline, such as Indexing, Retrieval, and Generation. They function as large, independent units.
2. **Sub-modules (Mid-Tier):** These handle more specific tasks within a module, such as Query Transformation, Document Loading, or Post-retrieval Reranking.
3. **Operators (Granular Tier):** These are the basic functional units within the sub-modules, executing atomic actions like embedding calculation, similarity search, or conditional logic checks.[6]

Advanced Architectural Patterns and RAG Flow Orchestration

The arrangement and execution order of these modules and operators define the **RAG Flow**, which serves as the blueprint for expressing complex RAG methodologies.[6] This framework shifts the system from a static pipeline to an Agentic system blueprint capable of dynamically selecting the optimal knowledge acquisition strategy.

The predominant RAG Flow patterns supported by this modular architecture include [6]:

- **Linear:** The sequential execution familiar to Naive RAG (Query $\rightarrow$ Retrieval $\rightarrow$ Generation).
- **Conditional:** Allows the flow to diverge based on criteria (e.g., checking the confidence score of retrieved documents before proceeding, or deciding to stop retrieval early).
- **Branching:** Executes multiple paths in parallel, useful for combining different retrieval

modalities (e.g., running sparse and dense retrieval simultaneously).
- **Looping (Iterative):** Supports recursive processes, such as iteratively re-ranking or generating intermediate results that refine subsequent retrieval steps.

These dynamic flows enable the implementation of advanced architectures, particularly those focused on multi-step reasoning. **KRAGEN (Knowledge Retrieval Augmented Generation Engine)** uses a structured, graph-of-thoughts prompting approach, leveraging the Branching/Conditional flow to decompose complex queries into subproblems and retrieve relevant knowledge subgraphs.[5] Similarly, **LQR (Layered Query Retrieval)** implements hierarchical planning specifically to manage complex multi-hop questions.[5] Another critical development is **R2AG (Retrieval information into RAG)**, which utilizes a Looping flow to recursively re-rank retrieval candidates *during* the generation process, dynamically updating the evidence based on the evolving partial answer state.[5]

A crucial area of optimization occurs upstream in the pre-retrieval phase. **Query Optimization** techniques, such as query rewriting, reformulate the user's initial input into a form more effective for knowledge retrieval.[12] Research, including RQ-RAG, focuses on training models to decompose and disambiguate complex queries, often proving vital because factual errors can frequently originate from ambiguous or poorly formulated user queries themselves.[12]

III. Optimization of the Retrieval Component: Advanced Strategies

The efficacy of a RAG system is inextricably linked to the quality and speed of its context retrieval mechanism. Architectural choices in this domain often involve a trade-off between semantic depth and keyword precision.

Comparative Analysis of Retrieval Modalities

Retrieval systems are broadly classified based on their underlying mechanism:

- **Sparse Retrieval** methods, such as BM25, rely on lexical overlap and keyword density. These approaches are highly interpretable and effective for queries demanding exact matches or specific proper nouns.[13]
- **Dense Retrieval** methods, such as DPR, leverage transformer models to generate

vector embeddings, thereby capturing the semantic intent and contextual relationships of the query and documents. Dense retrievers have shown strong performance in Open Domain Question Answering (ODQA), with DPR achieving a top-1 accuracy of 50.17% on the NQ dataset in certain analyses.[14]

Hybrid Retrieval Systems and Fusion Techniques

Neither sparse nor dense retrieval is universally superior; dense models struggle with precise keyword-dependent queries, while sparse models lack semantic context.[13] **Hybrid Retrieval** combines both dense vector search and traditional keyword methods to leverage their complementary strengths.[13]

This combined approach yields measurable and significant gains in retrieval performance. In an evaluation comparing an optimized hybrid RAG system incorporating SPLADE sparse vectors against a dense-only baseline, the hybrid system demonstrated superior ranking precision. It elevated the **Mean Reciprocal Rank (MRR) by 18.5%** (from 0.410 to 0.486).[15] This substantial MRR improvement is critical, as it indicates a much higher probability that the single most relevant document is ranked in the top position, directly translating to a better user experience and reduced LLM burden. Furthermore, the hybrid model improved **Recall@5 by 7.2%** (from 0.655 to 0.702), demonstrating a more consistent ability to surface the correct answer within the top five results.[15]

However, the pursuit of maximum factual grounding through hybrid methods introduces a clear economic and performance trade-off. The improved accuracy of the optimized hybrid approach incurred a significant latency increase of approximately 201 milliseconds per query, representing a **24.5% penalty**.[15] This establishes a quantifiable constraint: high accuracy via hybrid search sacrifices query speed. It is also observed that rigorous tuning of the sparse/dense weighting is non-negotiable; an initial configuration that aggressively favored keyword matches (sparse boost $= 3.0$) underperformed the dense-only baseline, highlighting the necessity of achieving a precise balance.[15]

Enhancements and Optimizations in Reranking

Reranking serves as a necessary post-retrieval processing step, refining the initially retrieved document set to improve precision and combat the "noise funnel" effect often generated by the initial retrieval phase.[5]

- **Adaptive Reranking:** Methods that dynamically adjust the number of documents reranked based on the complexity of the query are gaining traction. **RLT (Ranked List Truncation)** has been shown to improve MRR and nDCG metrics while concurrently reducing retrieval noise by **15%**.[5]
- **Integrated Retrieval-Generation:** Architectures are moving away from sequential steps toward integrated processes. **RankRAG** fine-tunes a language model to jointly score documents and generate answers, achieving a **7.8% improvement in MRR@10** while reducing latency.[5]
- **Confidence Calibration:** Research into Confidence-Calibrated RAG systems emphasizes that the way retrieved documents are ordered and presented in the prompt structure significantly affects the model's output certainty, highlighting that simplistic retrieval is insufficient to guarantee high confidence alongside factual accuracy.[5]

IV. RAG Performance, Scaling, and Economic Trade-offs

Retrieval-Augmented Generation has redefined the economic viability of smaller LLMs, establishing a new dynamic in the scaling debate. RAG eliminates the core economic burden of massive LLM retraining, making it an inherently cost-effective solution for continuous knowledge management.[8]

The RAG Effect: Augmenting Small LLMs vs. Large Baselines

RAG acts as a critical capability wrapper, allowing smaller, resource-efficient, open-source models to achieve a performance parity that competes with larger, proprietary architectures.[8] This effectively democratizes access to high-performing generative AI. Systematic research comparing RAG-Augmented smaller LLMs (e.g., TinyLlama, Mistral 7B) against their larger baseline counterparts (e.g., Llama 1 13B) demonstrates this leverage point.[17]

Quantifying Performance Gains in Knowledge-Intensive Tasks

The overall findings confirm that RAG-Augmentation significantly improves baseline LLM

performance across both lexical and semantic metrics, proving particularly beneficial for knowledge-intensive applications.[17] However, the magnitude of the gains reveals RAG's specific function as a factual constraint mechanism:

- **Lexical Similarity Metrics (BLEU and ROUGE):** RAG yields dramatic gains in lexical precision due to the retrieval of verbatim, contextually rich phrases, ensuring high n-gram overlap. RAG-Augmented Llama 3.1 8B, for example, demonstrated a remarkable **216.90% improvement in BLEU scores** and **145.10% gains in ROUGE-L**.[17] RAG-Augmented Mistral 7B also saw a 60.50% improvement in BLEU.[17] These figures illustrate RAG's strength in enforcing accurate factual word choice and structural adherence to source material.
- **Semantic Similarity Metrics (BERT-based metrics):** Semantic improvements, conversely, tend to be modest.[17] For instance, RAG-Augmented Mistral 7B achieved an **8.4% improvement in BERT Recall and F1 scores** over the larger Llama 1 13B baseline.[17] This discrepancy suggests that while RAG excels at factual injection and surface-level accuracy, the LLM still relies primarily on its parametric knowledge for deep semantic coherence and reasoning.

The analysis confirms that RAG-Augmented Mistral 7B serves as an **effective trade-off** compared to the larger baseline Llama 3.1 8B, providing comparable utility at a reduced operational cost.[17] More convincingly, RAG-Augmented Mistral 7B was found to **outperform** the un-augmented Llama 1 13B in *both* lexical and semantic similarity scores, validating RAG as the critical augmentation strategy for competitive performance in resource-constrained environments.[17]

V. Systemic Evaluation, Metrics, and Benchmarking

The evaluation of RAG systems is fundamentally complicated by their hybrid architecture and reliance on dynamically retrieved, external knowledge.[1] A comprehensive evaluation must assess the quality of retrieval, the factual grounding of generation, and the end-to-end integration.[18]

A Unified Evaluation Process (Auepora) and Core Metrics

The challenge of RAG evaluation led to the development of structured analytical frameworks, such as **A Unified Evaluation Process of RAG (Auepora)**.[1] Auepora organizes analysis

across three dimensions—Target, Dataset, and Metric—to guide the development and comparison of RAG benchmarks, acknowledging that most current benchmarks are incomplete in their coverage.[1]

RAG evaluation relies heavily on a specialized set of quantifiable metrics:

Retrieval Performance Metrics

These metrics quantify how effectively the system accesses and prioritizes the external context [19]:

- **Contextual Recall:** Determines the completeness of the retrieved set, measuring whether the retrieved context contains all the necessary information required to generate the ideal output.[19] This is often assessed using recall@k metrics against a ground truth.[20]
- **Contextual Relevancy:** Measures the purity of the retrieved context, assessing how relevant the top-K documents are to the input query.[19]

Generative Performance (Groundedness) Metrics

These metrics focus on the final output, particularly its adherence to the source material:

- Faithfulness (Groundedness): This is the paramount metric for RAG integrity. It measures the factual consistency of the response with the retrieved context.19 A perfect score requires that every claim made in the response be supported by the retrieved context.22 Faithfulness is computed as:

  $$Faithfulness Score = \frac{Number\ of\ claims\ in\ the\ response\ supported\ by\ the\ retrieved\ context}{Total\ number\ of\ claims\ in\ the\ response}$$
- **Answer Relevancy:** Measures the utility of the generated response, ensuring it aligns with the intent and subject matter of the initial query.[19] Auepora defines this as the relationship between the final Response and the original Query.[1]

Survey of Key RAG Evaluation Frameworks and Benchmarks

The necessity for nuanced evaluation has driven the community towards **LLM-as-a-judge** methods, which, despite adding cost and variability, capture the nuance of relevancy and faithfulness better than purely computation-based scores.[18]

- **RAGAs:** A widely adopted open-source framework utilizing LLM-as-a-judge methodologies to evaluate key metrics like Faithfulness and Context Relevance.[18]
- **ARES:** Designed specifically for stress-testing retrieval systems using adversarial examples, while also providing metrics for context relevance and faithfulness.[18]
- **Domain Specialization:** The field's maturity is evidenced by the emergence of highly specialized benchmarks, such as **LegalBench-RAG**, which is tailored to legal question-answering where factual misrepresentation results in critical compliance failures.[7] This specialization underscores that cross-domain generalization remains a significant challenge, requiring domain-specific tuning of the RAG system.[23]

VI. Critical Challenges and Mitigation Strategies

While RAG substantially mitigates knowledge deficits, the systems are not immune to failure modes, especially when handling complex reasoning or noisy context.[7] Advanced research has begun to categorize LLM hallucinations based on their origin to develop targeted solutions.[24]

A Taxonomy of LLM Hallucinations: Knowledge-Based vs. Logic-Based Errors

Current research proposes a taxonomy distinguishing between two fundamental types of hallucination [24]:

1. **Knowledge-Based Hallucinations:** Errors arising from factual inconsistency with the retrieved knowledge base or real-world data. RAG's external grounding capability is the direct mitigation strategy for this type of error.[24]
2. **Logic-Based Hallucinations:** Errors that persist even when correct facts are provided, stemming from deficits in the LLM's internal reasoning, inference chain, or complex logical processing.[24] RAG itself focuses on factual grounding and does not inherently resolve these internal reasoning flaws.[26]

RAG-Based Mitigation Paradigms

The RAG pipeline can mitigate these errors through strategic retrieval philosophies and context management:

- **Precise Retrieval vs. Broad Retrieval:** The design choice involves balancing high precision (Precise Retrieval, aiming for minimal, relevant context) against high recall (Broad Retrieval, aiming to ensure the fact is present but risking the introduction of retrieval noise).[12] Retrieval noise, ranking errors, or poor document ordering (a confidence calibration issue) can, in themselves, induce hallucination.[5]
- **Structured Knowledge Integration:** Utilizing structured approaches like **Knowledge Graph RAG (KG-RAG)** significantly improves accuracy in complex domains. By modeling relationships (e.g., between legal entities or statutes), KG-RAG reduces factual inconsistency caused by semantic ambiguity, achieving superior performance in domain-specific QA.[12]
- **Pre-Retrieval Optimization:** Crucially, query rewriting approaches (like RQ-RAG) enhance retrieval precision by reformulating ambiguous user input, addressing hallucinations that originate from the query stage itself.[12]

Synergistic Integration: RAG and Reasoning Enhancement

To combat logic-based hallucinations, RAG must be synergistically integrated with reasoning enhancement approaches.[24] These techniques improve the LLM's capacity for inference and complex task execution over the retrieved context:

- **CoT (Chain-of-Thought):** Improves internal reasoning by demanding that the LLM articulate its step-by-step logic.[12]
- **Tool-Augmented Reasoning:** Enhances capabilities by allowing the LLM to call external utilities, such as specialized databases or code execution tools, to solve complex logical sub-tasks.[12]
- **Symbolic Reasoning:** Leverages structured data and formal logic to ensure the generated output is consistent with formal rules.[12]

The ultimate solution for comprehensive reliability is the integration of RAG, Reasoning Enhancement, and their coordination within **Agentic Systems**, offering a unified framework to

address the full spectrum of knowledge-based and logic-based failures.[24]

VII. Frontiers of RAG Research: Future Directions

The advanced research trajectory for RAG focuses on expanding the architectural complexity, improving internal efficiency, and integrating non-textual data.

Multimodal RAG Systems (MM-RAG)

The shift toward **Multimodal RAG (MM-RAG)** involves integrating retrieval and generation across multiple modalities, including text, tables, images, and video.[27] This enables novel applications, such as socio-political and economic research that analyzes combined textual and visual data sources.[28] However, MM-RAG development is hindered by two key issues: the lack of unified frameworks capable of sophisticated reasoning across *more than two modalities*, and scalability limitations due to the requirement for separate training pipelines for new data types.[27] Furthermore, current evaluation benchmarks primarily focus on single- or dual-modality tasks, creating a significant evaluation gap for truly cross-modal systems.[27]

Advanced Knowledge Integration and Distillation Techniques

Standard RAG faces fundamental limitations regarding high inference cost for large contexts and an inability to synthesize and store **global document information**—the holistic meaning of a corpus.[29] This is particularly challenging in low-data or specialized domains.

Research has proposed solving this by modularizing knowledge through **Deep Context Distillation (DCD)**. This involves training document-level **Knowledge Modules (KMs)**, which are lightweight, parameter-efficient LoRA modules designed to store document knowledge.[29] KMs are trained using DCD to simulate the internal hidden states and logits of a teacher model that has the document in context.[30] This approach efficiently integrates general document understanding without incurring the high inference cost of passing massive context windows repeatedly. The synergy is clear: RAG provides fast, precise external lookup, while KMs efficiently store the distilled, global essence of a document set internally, creating a more

robust, low-latency, two-tiered knowledge architecture.[29]

Agentic RAG Systems and Next-Generation Orchestration

The trend in RAG architecture is a move towards intelligent, adaptive agents capable of self-directed knowledge acquisition. The structured definition of **Modular RAG** components and the flexibility of **RAG Flow** patterns provide the essential framework for this development.[6] Architectures like KRAGEN, which use graph-of-thoughts prompting for multi-hop reasoning, exemplify this shift.[5] Future systems must balance this growing intelligence with efficiency; work on **Sparse Context Selection** highlights the necessity of efficient sparse reformulations to maintain optimal recall without sacrificing query speed, ensuring that complexity does not introduce unacceptable latency.[5]

VIII. Archival Summary of Foundational and State-of-the-Art RAG Literature

Table 6: Summary of Foundational and State-of-the-Art RAG Literature (Archival Reference)

Paper/Framework	Authors/Source	Year	Core Contribution	Source Reference
Retrieval-Augmented Generation	Lewis et al.	2020	Foundational paper defining RAG-Sequence and RAG-Token models.	[4]
Retrieval-Augmented Generation for NLP: A Survey	Wu et al.	2024	Comprehensive review of techniques, training, and	[31]

			applications.	
Evaluation of RAG: A Survey (Auepora)	Gao et al. / Zhao et al.	2024	Structured framework for evaluation (Target, Dataset, Metric).	[1]
Mitigating Hallucination Survey	Li et al.	2025	Taxonomy of Knowledge-based vs. Logic-based hallucinations; RAG/Reasoning synergy.	[24]
Training Plug-and-Play Knowledge Modules with Deep Context Distillation	(Microsoft Research)	2025	Introduction of Knowledge Modules (KMs) for dynamic knowledge integration.	[29]
Modular RAG and RAG Flow	(OpenRAG)	2024	Three-tiered architecture and orchestration patterns (linear, conditional, looping).	[6]
KRAGEN / LQR	Matsumoto et al. / Huang et al.	2024	Architectures for multi-hop reasoning and query decomposition.	[5]

IX. Conclusions and Recommendations

Retrieval-Augmented Generation has evolved from a theoretical mechanism for knowledge grounding into a mature, modular architecture that defines the cutting edge of reliable LLM deployment. The foundational success of RAG is rooted in its proven ability to act as a **Factual Constraint Mechanism**, yielding high lexical precision gains, and its capacity as an **Economic Enabler** that allows smaller, resource-efficient LLMs to compete with larger baselines.[17]

However, the field is defined by critical trade-offs that guide future research: the need to balance the accuracy gains of hybrid retrieval ($+18.5\%$ MRR) against the associated latency penalty ($+24.5\%$ query time) [15], and the necessity of shifting RAG from a purely knowledge-grounding tool to an orchestrated system that integrates reasoning enhancement (CoT) to address pervasive logic-based hallucinations.[24]

For researchers starting an archive on RAG systems, the data suggests the following strategic focus:

1. **Prioritize Unified Evaluation Frameworks:** Resources dedicated to frameworks like Auepora and RAGAs are essential for establishing standardized definitions of critical metrics, particularly Faithfulness and Answer Relevancy, which capture the core RAG value proposition.[1]
2. **Document Architectures that Bridge the Retrieval-Generation Boundary:** Focus on papers detailing advanced, integrated RAG systems such as RankRAG and R2AG [5], which move beyond the sequential pipeline model and demonstrate dynamic, collaborative retrieval and generation.
3. **Establish Domain-Specific Benchmarking:** Given that cross-domain generalization is a persistent challenge, the archive should catalog domain-specific benchmarks (e.g., LegalBench-RAG, MedRAGBench) to facilitate research into domain-specific tuning and KG-RAG solutions.[12]
4. **Track Efficiency and Knowledge Distillation:** Include resources detailing Deep Context Distillation and Knowledge Modules [29] as these methods address the high inference cost and context window limitations inherent in traditional RAG, positioning them as essential components of future, efficient, two-tiered knowledge architectures.

Works cited

1. Evaluation of Retrieval-Augmented Generation: A Survey, accessed November 9, 2025, https://arxiv.org/pdf/2405.07437
2. Retrieval-augmented generation - Wikipedia, accessed November 9, 2025,

https://en.wikipedia.org/wiki/Retrieval-augmented_generation
3. What is Retrieval Augmented Generation (RAG)? - Databricks, accessed November 9, 2025, https://www.databricks.com/glossary/retrieval-augmented-generation-rag
4. Retrieval-Augmented Generation for Knowledge ... - NIPS papers, accessed November 9, 2025, https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf
5. Retrieval-Augmented Generation: A Comprehensive Survey of Architectures, Enhancements, and Robustness Frontiers - arXiv, accessed November 9, 2025, https://arxiv.org/html/2506.00054v1
6. Modular RAG: Transforming RAG Systems into LEGO-like Reconfigurable Frameworks, accessed November 9, 2025, https://arxiv.org/html/2407.21059v1
7. Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools - Daniel E. Ho - Stanford University, accessed November 9, 2025, https://dho.stanford.edu/wp-content/uploads/Legal_RAG_Hallucinations.pdf
8. RAG vs Traditional LLMs: Key Differences - Galileo AI, accessed November 9, 2025, https://galileo.ai/blog/comparing-rag-and-traditional-llms-which-suits-your-project
9. Advancements in RAG: A Comprehensive Survey of Techniques and Applications | by Sahin Ahmed, Data Scientist | Medium, accessed November 9, 2025, https://medium.com/@sahin.samia/advancements-in-rag-a-comprehensive-survey-of-techniques-and-applications-b6160b035199
10. Modular RAG using LLMs: What is it and how does it work? | by Sahin Ahmed, Data Scientist, accessed November 9, 2025, https://medium.com/@sahin.samia/modular-rag-using-llms-what-is-it-and-how-does-it-work-d482ebb3d372
11. Modular RAG and RAG Flow: Part II | by OpenRAG - Medium, accessed November 9, 2025, https://medium.com/@OpenRAG/modular-rag-and-rag-flow-part-ii-77b62bf8a5d3
12. Mitigating Hallucination in Large Language Models (LLMs): An Application-Oriented Survey on RAG, Reasoning, and Agentic Systems - arXiv, accessed November 9, 2025, https://arxiv.org/html/2510.24476v1
13. Integrate sparse and dense vectors to enhance knowledge retrieval in RAG using Amazon OpenSearch Service | AWS Big Data Blog, accessed November 9, 2025, https://aws.amazon.com/blogs/big-data/integrate-sparse-and-dense-vectors-to-enhance-knowledge-retrieval-in-rag-using-amazon-opensearch-service/
14. From Retrieval to Generation: Comparing Different Approaches - arXiv, accessed November 9, 2025, https://arxiv.org/html/2502.20245v1
15. Hybrid RAG: Boosting RAG Accuracy - Research AIMultiple, accessed November 9, 2025, https://research.aimultiple.com/hybrid-rag/
16. Hybrid Retrieval-Augmented Generation (RAG) Systems with Embedding Vector Databases, accessed November 9, 2025,

https://www.researchgate.net/publication/390326215_Hybrid_Retrieval-Augmented_Generation_RAG_Systems_with_Embedding_Vector_Databases
17. Retrieval-Augmented Generation vs. Baseline LLMs: A Multi-Metric ..., accessed November 9, 2025, https://www.mdpi.com/2078-2489/16/9/766
18. RAG Evaluation: Metrics and Benchmarks for Enterprise AI Systems - Label Your Data, accessed November 9, 2025, https://labelyourdata.com/articles/llm-fine-tuning/rag-evaluation
19. RAG Evaluation Metrics: Assessing Answer Relevancy, Faithfulness, Contextual Relevancy, And More - Confident AI, accessed November 9, 2025, https://www.confident-ai.com/blog/rag-evaluation-metrics-answer-relevancy-faithfulness-and-more
20. A complete guide to RAG evaluation: metrics, testing and best practices - Evidently AI, accessed November 9, 2025, https://www.evidentlyai.com/llm-guide/rag-evaluation
21. accessed November 9, 2025, https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/faithfulness/#:~:text=The%20Faithfulness%20metric%20measures%20how,supported%20by%20the%20retrieved%20context.
22. Faithfulness - Ragas, accessed November 9, 2025, https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/faithfulness/
23. RAGBench: Explainable Benchmark for Retrieval-Augmented Generation Systems - arXiv, accessed November 9, 2025, https://arxiv.org/html/2407.11005v1
24. [2510.24476] Mitigating Hallucination in Large Language Models (LLMs): An Application-Oriented Survey on RAG, Reasoning, and Agentic Systems - arXiv, accessed November 9, 2025, https://arxiv.org/abs/2510.24476
25. Mitigating Hallucination in Large Language Models (LLMs): An Application-Oriented Survey on RAG, Reasoning, and Agentic Systems | Request PDF - ResearchGate, accessed November 9, 2025, https://www.researchgate.net/publication/397006643_Mitigating_Hallucination_in_Large_Language_Models_LLMs_An_Application-Oriented_Survey_on_RAG_Reasoning_and_Agentic_Systems
26. RAG Hallucination: What is It and How to Avoid It, accessed November 9, 2025, https://www.k2view.com/blog/rag-hallucination/
27. Multimodal Retrieval-Augmented Generation: Unified Information Processing Across Text, Image, Table, and Video Modalities - ACL Anthology, accessed November 9, 2025, https://aclanthology.org/anthology-files/anthology-files/pdf/magmar/2025.magmar-1.5.pdf
28. A Multimodal Framework Embedding Retrieval-Augmented Generation with MLLMs for Eurobarometer Data - MDPI, accessed November 9, 2025, https://www.mdpi.com/2673-2688/6/3/50
29. Training Plug-and-Play Knowledge Modules with Deep Context Distillation | OpenReview, accessed November 9, 2025, https://openreview.net/forum?id=ghyyHZYORi
30. Training Plug-and-Play Knowledge Modules with Deep Context Distillation -

Microsoft, accessed November 9, 2025, https://www.microsoft.com/en-us/research/publication/training-plug-and-play-knowledge-modules-with-deep-context-distillation/
31. Retrieval-Augmented Generation for Natural Language ... - arXiv, accessed November 9, 2025, https://arxiv.org/pdf/2407.13193?