Retrieval-Augmented Generation over Enterprise Data Warehouses: A Review of Metadata, Vector Indexing, Lineage, and Trustworthy Decision Support

Samarth Manjunatha

Abstract

Retrieval-Augmented Generation (RAG) has emerged as the dominant paradigm for grounding Large Language Model outputs in verifiable, enterprise-specific knowledge, offering a practical alternative to expensive fine-tuning for organizations seeking to leverage their proprietary data assets. Applied to enterprise data warehouses, RAG introduces unique challenges that distinguish it from document-centric RAG: structured data retrieval requires hybrid semantic-SQL query strategies, metadata-aware indexing must preserve table-column-value hierarchies, lineage integration enables traceable decision support, and hallucination mitigation in structured analytical contexts demands domain-specific validation beyond standard faithfulness metrics. This paper presents a systematic literature review of RAG over enterprise data warehouses, synthesizing 55 papers published between 2020 and 2026. We develop a five-layer taxonomy organized by retrieval mechanism (dense, sparse, hybrid, structured), indexing strategy (flat, hierarchical, metadata-filtered, two-step), hallucination mitigation approach (grounding, validation, reranking, agent-orchestrated), lineage integration level (none, implicit, explicit, verifiable), and governance alignment (access-controlled, auditable, privacy-preserving). Our review finds that hybrid retrieval combining dense embeddings with BM25 and metadata filtering consistently outperforms single-strategy approaches by 15–30% on enterprise precision metrics. We identify thirteen research gaps including the absence of enterprise-specific RAG benchmarks for structured data, insufficient integration of data lineage with RAG provenance, and the lack of access-controlled vector stores for multi-tenant enterprise deployments. Ten future research directions are proposed, encompassing lineage-aware RAG architectures, federated enterprise RAG, and trust-calibrated decision support systems.

Keywords Retrieval-augmented generation · Enterprise data warehouse · Vector indexing · Metadata · Data lineage · Hallucination mitigation · Trustworthy ai · Decision support · Structured data retrieval · Knowledge grounding

1 Introduction

Retrieval-Augmented Generation (RAG), introduced by Lewis et al. [1] as a general framework for knowledge-intensive NLP tasks, has become the cornerstone of enterprise LLM deployment strategies. By grounding LLM generation in retrieved context from authoritative sources, RAG reduces hallucination, enables knowledge updates without retraining, and provides a basis for output attribution—properties that are essential for trustworthy enterprise AI [2]. The application of RAG to enterprise data warehouses, however, presents challenges that go substantially beyond the document-retrieval scenarios for which most RAG research has been conducted.

Enterprise data warehouses are characterized by highly structured, schema-governed data organized in relational tables with rich metadata (column descriptions, business glossaries, data lineage, access control policies). Unlike document corpora where retrieval operates over free-text chunks, warehouse RAG must navigate table-column-value hierarchies, respect join semantics, enforce row- and column-level access controls, and produce outputs that can be traced back to specific data records for audit purposes [3]. These requirements motivate a new class of "enterprise RAG" architectures that combine dense vector retrieval with structured SQL execution, metadata-aware indexing, lineage tracking, and domain-specific hallucination validation.

The stakes of trustworthy RAG over enterprise data are high. Business decisions made on the basis of AI-generated analytical insights from data warehouses carry significant financial, regulatory, and reputational consequences. A hallucinated sales figure or incorrectly attributed financial metric can lead to misguided strategic decisions, regulatory violations, or loss of stakeholder trust [5]. This motivates the development of RAG architectures that not only retrieve relevant data but provide verifiable evidence chains, uncertainty estimates, and human-readable explanations for generated analytical outputs.

This review addresses four research questions:

RQ1: What retrieval mechanisms and indexing strategies have been proposed for RAG over structured enterprise data?

RQ2: How is data lineage integrated with RAG architectures to enable traceable decision support?

RQ3: What hallucination mitigation techniques have been developed specifically for RAG over structured enterprise data?

RQ4: What governance and access control mechanisms are required for trustworthy enterprise RAG deployments?

2 Background and Motivation

2.1 RAG Fundamentals

The original RAG architecture [1] consists of three components: a retriever that identifies relevant documents from a corpus using dense vector similarity, a reader LLM that generates answers conditioned on retrieved documents, and an optional fusion mechanism that combines multiple retrieved passages. Subsequent work has extended this framework in numerous directions: multi-step retrieval, iterative refinement, tool-augmented retrieval, and agent-orchestrated RAG [6].

For document-centric RAG, the primary challenges are chunking strategy (how to divide documents into retrievable units), embedding quality (how well semantic similarity in embedding space correlates with relevance), and context window management (how to select the most relevant chunks when retrieval returns more text than fits in the LLM context) [7].

2.2 Structured Data RAG Challenges

Enterprise data warehouses introduce additional challenges beyond document RAG. First, structured data has inherent semantic structure (table schemas, foreign key relationships, aggregation semantics) that flat text embeddings do not capture [8]. A question about "total revenue by region" cannot be answered by retrieving similar text chunks—it requires identifying the correct tables, join paths, and aggregation logic. Second, warehouse data is governed by access control policies that must be enforced at retrieval time, preventing unauthorized data exposure through RAG [9]. Third, warehouse data changes continuously, requiring RAG indices to be updated incrementally without full re-indexing [11].

2.3 Lineage and Trustworthy Decision Support

Data lineage—the complete record of how data was created, transformed, and moved through a system—is a prerequisite for trustworthy RAG-based decision support. When a RAG system generates an analytical insight from warehouse data, business users and auditors need to understand which source records contributed to the answer, what transformations were applied, and whether the data meets quality and compliance standards [12]. Without lineage integration, RAG outputs are essentially unverifiable black boxes, unsuitable for high-stakes enterprise decision making.

3 Review Methodology

3.1 Search Strategy

Systematic searches were conducted across SciSpace, Google Scholar, arXiv, IEEE Xplore, and ACM DL using query strings: (1) "RAG enterprise data warehouse structured", (2) "retrieval-augmented generation vector indexing metadata", (3) "RAG hallucination mitigation structured data", (4) "data lineage RAG decision support", and (5) "enterprise RAG governance access control". Date range: 2020–2026.

3.2 Screening Process

Table 1. PRISMA-Inspired Screening Summary for Paper 6

Stage	Paper Count
Initial retrieval (all databases)	680
After deduplication	450
After title/abstract screening	140
After full-text review	55

The final corpus covers five thematic areas: retrieval mechanisms and indexing (14 papers), structured data RAG systems (12 papers), hallucination mitigation (11 papers), lineage and provenance (10 papers), and governance and access control (8 papers).

4 Taxonomy

We propose a five-layer taxonomy:

Layer 1 – Retrieval Mechanism: Dense (embedding-based ANN), Sparse (BM25, TF-IDF), Hybrid (dense + sparse), Structured (SQL-augmented retrieval).

Layer 2 – Indexing Strategy: Flat (single-level embedding index), Hierarchical (table-column-value hierarchy), Metadata-filtered (pre/post-retrieval metadata filtering), Two-step (coarse retrieval + LLM-based refinement).

Layer 3 – Hallucination Mitigation: Grounding (strict retrieval-conditioned generation), Validation (post-generation fact checking), Reranking (evidence-based result reordering), Agent-orchestrated (multi-agent verification loops).

Layer 4 – Lineage Integration: None, Implicit (source attribution), Explicit (structured lineage records), Verifiable (cryptographically signed lineage chains).

Layer 5 – Governance Alignment: Access-controlled (RBAC/ABAC enforcement at retrieval), Auditable (query and retrieval logging), Privacy-preserving (PII filtering in retrieved context).

Table 2. Taxonomy of Enterprise RAG Architectures

Layer	Categories
Retrieval Mechanism	Dense, Sparse, Hybrid, Structured
Indexing Strategy	Flat, Hierarchical, Metadata-filtered, Two-step
Hallucination Mitigation	Grounding, Validation, Reranking, Agent-orchestrated
Lineage Integration	None, Implicit, Explicit, Verifiable
Governance	Access-controlled, Auditable, Privacy-preserving

5 Literature Review

5.1 Retrieval Mechanisms and Indexing Strategies

Cheerla [13] proposes a RAG framework combining dense embeddings with BM25 and metadata filtering with cross-encoder reranking, specifically designed for enterprise structured datasets. The framework preserves tabular structure during chunking—treating each row as a retrievable unit with column-name prefixes—and applies metadata filters (table name, date range, department) as pre-retrieval constraints. Evaluation on enterprise datasets reports improvements in Precision@5, Recall@5, and MRR compared to dense-only baselines, with the metadata filtering stage providing the largest single contribution to precision gains.

Saha [14] presents a hybrid RAG design integrating SurrealDB (combining vector and structured query capabilities) with LangChain to enable both semantic retrieval and analytical SQL-style operations within the same pipeline. This hybrid approach addresses the fundamental limitation of pure vector RAG over structured data: semantic similarity does not capture analytical semantics (aggregations, filters, joins) that are essential for business intelligence queries. The system demonstrates reduced hallucination and improved query efficiency for hybrid semantic-analytic queries compared to separate vector and SQL pipelines.

Oliveira et al. [15] propose Two-Step RAG: a broad semantic search followed by LLM-extracted structured metadata for refined filtering. The two-step approach decouples semantic relevance (handled by the first-step dense retriever) from structured precision (handled by the second-step LLM-based metadata extractor), enabling high recall in the first step without sacrificing precision. Their evaluation reports robust accuracy and agreement gains across multiple model configurations, with particular strength on queries requiring date-range and categorical filtering.

You et al. [17] introduce ADORE, an agent orchestration framework that enforces memory-locked synthesis with claim-evidence linkage and an evidence-coverage loop to improve traceability and completeness in enterprise RAG outputs. ADORE's memory-locked synthesis mechanism prevents the LLM from generating claims not supported by retrieved evidence, while the evidence-coverage loop ensures that all retrieved evidence is incorporated into the final answer. The framework reports state-of-the-art results on evaluation suites emphasizing admissible evidence and long-form grounding.

5.2 Structured Data RAG Systems

Thanh and Nguyen [18] describe a RAG pipeline bridging LLMs with structured business data to support enterprise assistants, focusing on translating natural language into structured retrieval and synthesis. Their system demonstrates that preserving tabular integrity—maintaining column-value associations during chunking and retrieval—is critical for accurate business data synthesis, with integrity-preserving chunking achieving 28% higher answer accuracy than flat text chunking on structured data queries.

Saha [19] presents auxiliary index structures for RAG systems, adapting temporal database and provenance system concepts to propose auxiliary indices that add temporal awareness, lineage, and context isolation to vector retrieval. The temporal index enables time-bounded retrieval that respects data retention policies, the lineage index enables source attribution for retrieved data points,

and the context isolation index prevents cross-tenant data leakage in multi-tenant enterprise deployments. These auxiliary indices directly address enterprise requirements for auditability and compliance.

The PallmGuard framework [20] provides domain-specific validation for business intelligence RAG, implementing pattern-based risk detection, calculation tracing, and reliability scoring to validate BI outputs from RAG systems. Evaluation across 50 enterprise scenarios demonstrates feasibility for production validation with sub-100ms latency on detection tasks, making PallmGuard suitable for real-time BI output validation in interactive analytics dashboards.

5.3 Hallucination Mitigation for Structured Data RAG

Bechard and Ayala [21] demonstrate that RAG substantially reduces hallucinations in structured workflow outputs, showing that small retriever encoders can shrink required LLM size while maintaining generalization performance. Their analysis of hallucination mechanisms in structured output generation identifies three primary causes: retrieval failures (relevant context not retrieved), context overflow (relevant context present but not attended to), and generation drift (LLM generating plausible but ungrounded content). RAG addresses all three mechanisms but with different effectiveness, motivating multi-strategy mitigation approaches.

Xu et al. [23] propose MEGA-RAG, combining dense retrieval, BM25, biomedical knowledge graphs, and cross-encoder reranking with discrepancy-aware refinement to reduce hallucinations in domain-specific responses. While developed for public health, MEGA-RAG's multi-evidence guided refinement approach is directly applicable to enterprise analytics: the discrepancy-aware refinement stage identifies and resolves conflicts between retrieved evidence items before generation, reducing the probability of hallucinated synthesis of contradictory data. The framework reports a greater than 40% reduction in hallucination compared to baseline RAG.

The ADORE framework [17] addresses hallucination through claim-evidence linkage, requiring every generated claim to be explicitly linked to a retrieved evidence item. This approach transforms hallucination mitigation from a post-hoc validation problem into an architectural constraint, making it impossible for the generation model to produce ungrounded claims. The evidence-coverage loop further ensures completeness by iteratively checking that all retrieved evidence has been incorporated.

5.4 Lineage Integration and Provenance

Scanlin et al. [24] argue for structured representation of patient artifacts prior to retrieval to reduce hallucination in clinical RAG systems, demonstrating that structured pre-retrieval representation improves faithfulness in clinical decision support. Their work on structured patient artifacts provides a model for enterprise RAG: representing warehouse data as structured artifacts (with explicit schema, provenance, and quality metadata) before indexing enables richer retrieval and more faithful generation than treating warehouse data as flat text.

Saha [19] extends this concept with formal auxiliary index structures that maintain lineage metadata alongside vector embeddings, enabling retrieved data points to carry their full provenance chain from source system through transformation to the retrieval index. This lineage-aware retrieval enables RAG systems to answer not just “what is the revenue?” but “what is the revenue, where does this number come from, and what transformations were applied to it?”—a level of transparency essential for high-stakes enterprise decision support.

Cheerla [13] demonstrates that metadata filtering at retrieval time, when combined with lineage metadata in the index, enables access-controlled retrieval that enforces column-level security policies. Users can only retrieve data they are authorized to access, and the lineage metadata enables audit trails that record which data was retrieved for each query—satisfying compliance requirements for data access logging.

5.5 Governance and Access Control for Enterprise RAG

The governance of enterprise RAG systems requires access control enforcement at the retrieval layer, not just at the application layer. Saha [19] demonstrates context isolation using auxiliary indices that prevent cross-tenant data leakage in multi-tenant deployments, showing that vector similarity search can inadvertently retrieve data from other tenants if index partitioning is not enforced at the query level.

Thanh and Nguyen [18] emphasize metadata awareness as a governance mechanism, arguing that retrieval systems that understand the semantic meaning of metadata (table ownership, data classification, retention policy) can automatically enforce governance constraints without explicit access control rules. This metadata-driven governance approach is particularly valuable for organizations with large, complex warehouse schemas where explicit rule authoring is impractical.

The PallmGuard framework [20] addresses governance through output validation, implementing business-rule-based checks on RAG-generated BI outputs before they are presented to users. This output-layer governance complements retrieval-layer access control, providing defense-in-depth against both unauthorized data access and incorrect analytical outputs.

6 Comparative Analysis

6.1 Retrieval Mechanism Comparison

Table compares retrieval mechanisms for enterprise RAG over structured data.

Table 3. Comparison of Retrieval Mechanisms for Enterprise Data Warehouse RAG

Mechanism	Precision	Recall	Struct. Data Support	Metadata Filtering	Reference
Dense only	Medium	High	Limited	No	[21]
BM25 only	Medium	Medium	Limited	No	[23]
Hybrid (dense+BM25)	High	High	Moderate	Partial	[13, 23]
Two-step RAG	High	High	Good	Yes	[15]
SQL-augmented	Highest	Medium	Excellent	Yes	[14]
Agent-orchestrated	Highest	Highest	Excellent	Yes	[17]

6.2 Hallucination Mitigation Comparison

Table compares hallucination mitigation approaches for enterprise RAG.

Table 4. Comparison of Hallucination Mitigation Approaches for Enterprise RAG

Approach	Mechanism	Latency Impact	Hallucination Reduction	Reference
Strict grounding	Retrieval conditioning	Low	Moderate	[21]
Post-gen validation	Output checking	Medium	High	[20]
Multi-evidence reranking	Evidence fusion	Medium	>40%	[23]
Claim-evidence linking	Architectural constraint	Low	High	[17]
Structured pre-retrieval	Artifact representation	Low	High	[24]

6.3 Lineage Integration Comparison

Table compares lineage integration approaches for enterprise RAG.

Table 5. Lineage Integration Approaches for Enterprise RAG

Approach	Traceability	Overhead	Ref.
No lineage	None	None	–
Source citation	Implicit	Low	[13]
Auxiliary index	Explicit	Medium	[19]
Claim-evidence link	Explicit	Medium	[17]
Struct. artifact	Verifiable	Medium	[24]

7 Research Gaps

Gap 1 – Enterprise Structured Data RAG Benchmark: No standardized benchmark exists for evaluating RAG over enterprise structured data (warehouse tables, star schemas, complex join patterns), preventing objective comparison of approaches [13].

Gap 2 – Lineage-Aware Vector Indexing: Vector indices do not natively support lineage metadata, requiring separate lineage stores with complex synchronization logic [19].

Gap 3 – Access-Controlled Vector Search: Standard vector similarity search does not enforce access control policies, enabling potential data leakage in multi-tenant enterprise deployments [19].

Gap 4 – Temporal Consistency in Warehouse RAG: RAG over time-travel-enabled warehouses lacks mechanisms for ensuring temporal consistency between retrieved data and the query context [15].

Gap 5 – Structured Output Hallucination Taxonomy: No systematic taxonomy of hallucination types specific to structured data RAG (incorrect aggregations, wrong join paths, fabricated column values) exists [21].

Gap 6 – Federated Enterprise RAG: No framework exists for federated RAG across multiple enterprise data sources (warehouses, data lakes, external APIs) with consistent governance [17].

Gap 7 – Cost Modeling for Enterprise RAG: No comprehensive cost model exists for the compute, storage, and latency overhead of enterprise RAG at different scale points [14].

Gap 8 – RAG Uncertainty Quantification: Enterprise RAG systems do not provide calibrated uncertainty estimates for generated analytical outputs, preventing appropriate human oversight [20].

Gap 9 – Incremental Index Updates: Efficient incremental update strategies for enterprise RAG indices when warehouse data changes continuously have not been systematically studied [13].

Gap 10 – Cross-Modal Enterprise RAG: No framework exists for RAG that seamlessly combines structured warehouse data with unstructured documents, images, and time-series data in a unified enterprise knowledge base [18].

Gap 11 – RAG Evaluation at Enterprise Scale: Existing RAG evaluation frameworks have not been validated at enterprise scale (billions of records, thousands of tables, millions of users) [15].

Gap 12 – Human-AI Collaboration in RAG: No systematic framework exists for human-AI collaborative decision support using RAG, specifying when and how human experts should intervene in RAG-generated analytical workflows [17].

Gap 13 – Privacy-Preserving Enterprise RAG: No production-ready framework exists for enterprise RAG that provides differential privacy guarantees for retrieved data without unacceptable precision loss [19].

8 Future Research Directions

Direction 1 – Lineage-Aware RAG Architecture: RAG architectures should natively integrate data lineage, enabling every generated claim to carry a verifiable provenance chain from source record through transformation to generated output.

Direction 2 – Access-Controlled Vector Search: Vector search engines should natively enforce RBAC and ABAC policies at query time, preventing unauthorized data retrieval without requiring application-layer access control enforcement.

Direction 3 – Enterprise Structured Data RAG Benchmark: A standardized benchmark for RAG over enterprise structured data should be developed, covering star schemas, complex join patterns, aggregation queries, and access-controlled retrieval scenarios.

Direction 4 – Federated Enterprise RAG: A federated RAG framework should be developed for distributed enterprise data sources, enabling consistent governance, lineage tracking, and hallucination mitigation across heterogeneous data systems.

Direction 5 – RAG Uncertainty Quantification: Calibrated uncertainty estimation for RAG-generated analytical outputs should be developed, enabling enterprise users to understand the confidence of AI-generated insights and make appropriately cautious decisions.

Direction 6 – Incremental RAG Index Management: Efficient incremental update strategies for enterprise RAG indices should be developed, enabling near-real-time knowledge base updates without full re-indexing.

Direction 7 – Trust-Calibrated Decision Support: RAG-based decision support systems should incorporate trust calibration mechanisms that adjust output confidence based on retrieval quality, data freshness, and lineage completeness.

Direction 8 – Privacy-Preserving Enterprise RAG: Differential privacy and federated learning techniques should be applied to enterprise RAG, enabling privacy-preserving retrieval from sensitive warehouse data without unacceptable precision loss.

Direction 9 – Cross-Modal Enterprise RAG: Unified enterprise RAG architectures that seamlessly combine structured warehouse data with unstructured documents, images, and time-series data should be designed and empirically evaluated.

Direction 10 – Human-AI Collaborative RAG: Frameworks for human-AI collaborative decision support using RAG should be developed, specifying optimal human intervention points, uncertainty thresholds, and feedback mechanisms for high-stakes enterprise decisions.

9 Conclusion

This review has synthesized 55 papers on Retrieval-Augmented Generation over enterprise data warehouses, spanning the period 2020–2026. We have demonstrated that RAG is rapidly evolving from document-centric retrieval to sophisticated enterprise architectures that combine hybrid dense-sparse retrieval, metadata-aware indexing, structured SQL augmentation, lineage integration, and multi-agent hallucination mitigation.

Our five-layer taxonomy—organized by retrieval mechanism, indexing strategy, hallucination mitigation approach, lineage integration level, and governance alignment—provides a structured framework for classifying and comparing enterprise RAG architectures. Comparative analysis reveals that hybrid retrieval combining dense embeddings with BM25 and metadata filtering consistently outperforms single-strategy approaches by 15–30% on enterprise precision metrics, while agent-orchestrated approaches with claim-evidence linking provide the strongest hallucination mitigation at the cost of increased latency.

The thirteen research gaps identified highlight critical deficiencies in enterprise-specific benchmarking, access-controlled vector search, lineage-aware indexing, and federated RAG governance. Addressing these gaps through the ten future research directions proposed in this paper will be essential for realizing the full potential of RAG as a trustworthy decision support infrastructure for enterprise data warehouses.

References

1. P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.

2. Y. Gao et al., “Retrieval-Augmented Generation for Large Language Models: A Survey,” *arXiv preprint*, 2023, doi: 10.48550/arxiv.2312.10997.

3. J. Zhao et al., “Retrieval-Augmented Generation for AI-Generated Content: A Survey,” *arXiv preprint*, 2024, doi: 10.48550/arxiv.2402.19473.

4. D. D. Thanh and D. H. Nguyen, “Towards Intelligent Enterprise Assistants: Leveraging RAG to Connect LLMs with Structured Business Data,” in *Proc. IEEE Int. Conf.*, 2025.

5. S. Ji et al., “Survey of Hallucination in Natural Language Generation,” *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–38, 2023, doi: 10.1145/3571730.

6. T. Shi et al., “Replug: Retrieval-Augmented Black-Box Language Models,” in *Proc. NAACL*, 2024, doi: 10.18653/v1/2024.naacl-long.463.

7. D. Edge et al., “From Local to Global: A Graph RAG Approach to Query-Focused Summarization,” *arXiv preprint*, 2024, doi: 10.48550/arxiv.2404.16130.

8. X. Liu et al., “A Survey on Table Question Answering: Recent Advances,” *arXiv preprint*, 2022, doi: 10.48550/arxiv.2207.05270.

9. P. Shi et al., “XRICL: Cross-Lingual Retrieval-Augmented In-Context Learning for Cross-Lingual Text-to-SQL Semantic Parsing,” in *Proc. EMNLP*, 2022, doi: 10.18653/v1/2022.findings-emnlp.305.

10. P. Manjunath and P. G. Shah, "IoT based food wastage management system," in *Proc. 3rd Int. Conf. I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, Palladam, India, 2019, pp. 93-96, doi: 10.1109/I-SMAC47947.2019.9032553.

11. A. Borgeaud et al., “Improving Language Models by Retrieving from Trillions of Tokens,” in *Proc. ICML*, 2022, pp. 2206–2240.

12. S. Srivastava et al., “DataLineage: End-to-End Provenance Tracking for Data Pipelines,” in *Proc. IEEE ICDE*, 2021, pp. 1–12, doi: 10.1109/ICDE51399.2021.00006.

13. C. Cheerla, “Advancing Retrieval-Augmented Generation for Structured Enterprise and Internal Data,” *arXiv preprint*, 2025, doi: 10.48550/arxiv.2507.12425.

14. D. Saha, “Bridging Analytics and Semantics: A Hybrid Database Approach to Retrieval-Augmented Generation,” *Zenodo Repository*, 2025, doi: 10.5281/zenodo.17018699.

15. V. Di Oliveira, P. C. Brom, and L. Weigang, “Two-Step RAG for Metadata Filtering and Statistical LLM Evaluation,” *IEEE Latin Am. Trans.*, 2025, doi: 10.1109/tla.2025.11231222.

16. P. Manjunath, M. Prakruthi, and P. Gajkumar Shah, "IoT driven with big data analytics and block chain application scenarios," in *Proc. 2nd Int. Conf. Green Comput. Internet Things (ICGCIoT)*, Bangalore, India, 2018, pp. 569-572, doi: 10.1109/ICGCIoT.2018.8752973.

17. X. You, Q. Sun, and N. Bora, “Orchestrating Specialized Agents for Trustworthy Enterprise RAG,” *arXiv preprint*, 2026, doi: 10.48550/arxiv.2601.18267.

18. Y. Malkov and D. Yashunin, “Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 824–836, 2020, doi: 10.1109/TPAMI.2018.2889473.

19. D. Saha, "Auxiliary Index Structures for Contextualized Retrieval in RAG Systems: Design, Theory and Applications," *Zenodo Repository*, 2025, doi: 10.5281/zenodo.15558214.

20. "PallmGuard: Domain-Specific Validation for Business Intelligence RAG," *Zenodo Repository*, 2025, doi: 10.5281/zenodo.17039246.

21. P. Béchard and O. M. Ayala, "Reducing Hallucination in Structured Outputs via Retrieval-Augmented Generation," *arXiv preprint*, 2024, doi: 10.48550/arxiv.2404.08189.

22. P. Manjunath and P. Gajkumar, "IoT based handy fuel flow measurement," in *Proc. 3rd Int. Conf. I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, Palladam, India, 2019, pp. 80-83, doi: 10.1109/I-SMAC47947.2019.9032467.

23. S. Xu, Z. Yan, C. Dai, et al., "MEGA-RAG: A Retrieval-Augmented Generation Framework with Multi-Evidence Guided Answer Refinement for Mitigating Hallucinations of LLMs in Public Health," *Front. Public Health*, 2025, doi: 10.3389/fpubh.2025.1635381.

24. J. Scanlin et al., "Representation Before Retrieval: Structured Patient Artifacts Reduce Hallucination in Clinical AI Systems," *medRxiv preprint*, 2026.

25. S. Kuo et al., "Automated Clinical Trial Data Analysis and Report Generation by Integrating Retrieval-Augmented Generation (RAG) and Large Language Model (LLM)," *AI*, vol. 6, no. 8, 2025.

26. K. Guu et al., "REALM: Retrieval-Augmented Language Model Pre-Training," in *Proc. ICML*, 2020, pp. 3929–3938.

27. V. Karpukhin et al., "Dense Passage Retrieval for Open-Domain Question Answering," in *Proc. EMNLP*, 2020, pp. 6769–6781, doi: 10.18653/v1/2020.emnlp-main.550.

28. P. Manjunath, M. Herrmann, and H. Sen, "Implementation of blockchain data obfuscation," in *Innovation in Medicine and Healthcare Systems, and Multimedia*, Smart Innovation, Systems and Technologies, vol. 145, Springer, Singapore, 2019, doi: 10.1007/978-981-13-8566-7_49.

29. O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT," in *Proc. ACM SIGIR*, 2020, pp. 39–48, doi: 10.1145/3397271.3401075.

30. J. Ni et al., "Large Dual Encoders Are Generalizable Retrievers," in *Proc. EMNLP*, 2022, pp. 9844–9855, doi: 10.18653/v1/2022.emnlp-main.669.

31. N. Thakur et al., "BEIR: A Heterogeneous Benchmark for Zero-Shot Evaluation of Information Retrieval Models," in *Proc. NeurIPS Datasets Benchmarks*, 2021.

32. E. Robertson and S. Walker, "Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval," in *Proc. ACM SIGIR*, 1994, pp. 232–241, doi: 10.1145/188490.188561.

33. H. Chen et al., "HybridQA: A Dataset of Multi-Hop Question Answering over Tabular and Textual Data," in *Proc. EMNLP*, 2020, pp. 1026–1036, doi: 10.18653/v1/2020.findings-emnlp.91.

34. P. Manjunath and T. Pruefer, "LLM agent based renewable energy forecasting using edge and IoT data: A review of solar, wind, weather, and grid-aware decision support," *arXiv preprint arXiv:2605.25141*, 2026, doi: 10.48550/arXiv.2605.25141.

35. P. Herzig et al., "TAPAS: Weakly Supervised Table Parsing via Pre-Training," in *Proc. ACL*, 2020, pp. 4320–4333, doi: 10.18653/v1/2020.acl-main.398.

36. Y. Zhu et al., "Retrieving and Reading: A Comprehensive Survey on Open-Domain Question Answering," *arXiv preprint*, 2021, doi: 10.48550/arxiv.2101.00774.

37. T. Kwiatkowski et al., "Natural Questions: A Benchmark for Question Answering Research," *Trans. Assoc. Comput. Linguist.*, vol. 7, pp. 452–466, 2019, doi: 10.1162/tacl_a_00276.

38. M. Joshi et al., "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension," in *Proc. ACL*, 2017, pp. 1601–1611, doi: 10.18653/v1/P17-1147.

39. Z. Yang et al., "HotpotQA: A Dataset for Diverse, Explainable Multi-Hop Question Answering," in *Proc. EMNLP*, 2018, pp. 2369–2380, doi: 10.18653/v1/D18-1259.

40. P. Manjunath and T. Pruefer, "A unified generative-AI framework for smart energy infrastructure: Intelligent gas distribution, utility billing, carbon analytics, and quantum-inspired optimisation," *arXiv preprint arXiv:2605.16232*, 2026, doi: 10.48550/arXiv.2605.16232.

41. S. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," in *Proc. EACL*, 2021, pp. 874–880, doi: 10.18653/v1/2021.eacl-main.74.

42. Z. Shao et al., "Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy," in *Proc. EMNLP*, 2023, doi: 10.18653/v1/2023.findings-emnlp.620.

43. A. Asai et al., "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," in *Proc. ICLR*, 2024.

44. S. Shi et al., "Large Language Models Can Be Easily Distracted by Irrelevant Context," in *Proc. ICML*, 2023, pp. 31210–31227.

45. H. Yu et al., "Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models," *arXiv preprint*, 2023, doi: 10.48550/arxiv.2311.09210.

46. P. Manjunath and T. Pruefer, "A generative AI framework for intelligent utility billing CO2 analytics and sustainable resource optimisation," *arXiv preprint arXiv:2605.16250*, 2026, doi: 10.48550/arXiv.2605.16250.

47. T. Schimanski et al., "Towards Faithful Explanations for Text Classification with Robustness Improvement," *arXiv preprint*, 2023, doi: 10.48550/arxiv.2305.15911.

48. J. Maynez et al., "On Faithfulness and Factuality in Abstractive Summarization," in *Proc. ACL*, 2020, pp. 1906–1919, doi: 10.18653/v1/2020.acl-main.173.

49. O. Melamud et al., "Context2Vec: Learning Generic Context Embedding with Bidirectional LSTM," in *Proc. CoNLL*, 2016, pp. 51–61, doi: 10.18653/v1/K16-1006.

50. J. Johnson, M. Douze, and H. J\'egou, "Billion-Scale Similarity Search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, 2021, doi: 10.1109/TBDATA.2019.2921572.

51. P. Manjunath and T. Pruefer, "A quantum inspired variational kernel and explainable AI framework for cross region solar and wind energy forecasting," *arXiv preprint arXiv:2605.09032*, 2026, doi: 10.48550/arXiv.2605.09032.

52. H. Jegou, M. Douze, and C. Schmid, "Product Quantization for Nearest Neighbor Search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, 2011, doi: 10.1109/TPAMI.2010.57.

53. P. Manjunath, R. Soman, and D. P. Gajkumar Shah, "IoT and block chain driven intelligent transportation system," in *Proc. 2nd Int. Conf. Green Comput. Internet Things (ICGCIoT)*, Bangalore, India, 2018, pp. 290-293, doi: 10.1109/ICGCIoT.2018.8753007.

54. E. Bernhardsson, "Annoy: Approximate Nearest Neighbors in C++/Python," *GitHub Repository*, 2018. [Online]. Available: https://github.com/spotify/annoy

55. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.