



International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Metadata-Grounded Retrieval for AI-Driven Field Discovery in Enterprise Data Platforms

Rohit Raravi

Independent Researcher

Abstract

A field-discovery agent that hallucinates a schema field is not a minor annoyance; it sends an enterprise customer down a path that does not exist. This article argues that grounding quality, not generator scale or prompt engineering, is what actually determines whether a retrieval-augmented field-discovery system can be trusted in production, and that grounding has to be built into the retrieval architecture itself rather than requested through instructions a model may or may not follow. The evidence comes from a production deployment built for an enterprise experience-data platform, where natural-language field discovery had to operate across customer metadata models containing thousands of fields spread across schemas, datasets, and sandboxes. The architecture treats schema structure, organizational context, and field-level permissions as retrieval signals evaluated before ranking, not filters applied after generation, and adds a validation layer that checks every field reference in a draft response against live metadata before that response reaches the user. Measured against a semantic-similarity-only baseline, the architecture reduces unverifiable field references and strengthens permission compliance, consistently across evaluation runs though reported here qualitatively rather than as a single benchmarked figure, at a latency cost the underlying vector-search infrastructure absorbs without difficulty. The deployment also coincided with a reduction in enterprise onboarding time from approximately two months to two weeks, a deployment-level outcome attributable to the broader discovery capability. Architectural grounding, this article suggests, produces more durable reliability than tuning the generator ever will, a claim with direct consequences for where engineering effort should go when hallucination shows up in production AI-discovery systems.

Keywords

retrieval-augmented generation, metadata grounding, enterprise data platform, schema discovery, hallucination mitigation, vector search, field-level retrieval, natural language query

1. Introduction

Enterprise data platforms rarely offer one schema. They offer hundreds, each holding dozens or hundreds of fields, scattered across datasets and sandboxes provisioned at different times for different business units. Ask a customer to find the one field that holds the information they actually need, and you are asking them to search a structure nobody fully remembers building. Manual schema browsing does not scale past a few hundred fields; for platforms with thousands, it barely functions at all, and this navigational burden has long been one of the largest drags on enterprise onboarding speed.

Retrieval-augmented generation has become the standard fix. A user asks a question in plain language, a retriever pulls candidate fields, and a generator writes an answer naming the relevant ones. That collapses days of exploration into one exchange, in theory. In production, the theory runs into a specific, recurring problem: generators reference fields that do not exist, misstate a field's location within its schema, or surface fields a user's sandbox should have kept invisible. Each failure costs more trust than the productivity gain buys back, a lesson learned the hard way during early rollout.

Two remedies dominate current practice, and only one of them works reliably. The first treats hallucination as something a bigger model or a better prompt will eventually fix. The second, argued less often in practitioner writing than it should be, treats hallucination as a retrieval-architecture failure: if the retriever hands the generator only fields that are verifiably real, permission-appropriate, and contextually relevant, there is simply less room left for the generator to invent anything. This article takes the second position, and pushes it further than most existing accounts do. Grounding is not a nice-to-have layered on top of a capable model. It is the dominant lever. And it has to be enforced by the retrieval architecture itself, because prompt instructions are precisely the kind of guidance a generator can quietly ignore under pressure.

The argument is grounded, deliberately, in a real deployment rather than a synthetic benchmark. A generative field-discovery capability was built for an enterprise experience-data platform to help customers locate relevant fields using natural-language requests, across customer data models spanning thousands of fields, schemas, datasets, and sandboxes. What started as a chatbot-embedded feature evolved into a reusable agent

callable from broader workflows, a shift that forced the retrieval architecture to generalize across invocation contexts it was not originally built for. That pressure is precisely what exposed the limits of semantic-similarity retrieval and motivated the schema-aware, permission-aware, validation-backed design this article describes. What follows situates that design within the retrieval-augmented generation literature, walks through the architecture and its evaluation framework, reports grounding-accuracy and latency findings against a semantic-similarity baseline, and closes with what the results mean for how enterprise AI-discovery systems ought to be built.

2. Literature Review

2.1 *Retrieval as the Locus of Hallucination Control*

The generators inside modern RAG pipelines are, almost without exception, transformer-based language models, an architecture that made large-scale, context-sensitive text generation practical in the first place [1]. Grounding that generation in external facts predates retrieval-augmented generation as a formal architecture, though. Ghazvininejad and colleagues demonstrated that conditioning dialogue responses on retrieved facts curbed generic, unsupported output well before large language models made the technique ubiquitous [2]. Lewis and colleagues later formalized retrieval-augmented generation as a single architecture pairing a parametric generator with a non-parametric retrieval index, and that template still underlies nearly every enterprise system built since [3]. What has shifted is not the architecture but a hardening consensus that the retriever, not the generator, is where most reliability failures actually start.

Bécharde and Ayala demonstrate this directly: a small, well-trained retriever paired with a modest generator can match a much larger generator's reliability on structured-output tasks, which undercuts the assumption that scale is the primary fix for hallucination [4]. Stolfo pushes the point further and makes it uncomfortable. Across three datasets and four model families, a substantial share of generated sentences lacked verifiable grounding even when they happened to be factually correct, the model was right, but for reasons its own retrieval pipeline could not confirm [5]. That gap, between being correct and being grounded, is exactly why prompt-level fixes cannot close the problem. A prompt can instruct a model to cite its sources. It cannot force the model to have actually used them. Wang and colleagues' broader survey of factuality failure modes across large language models situates this specific gap within a wider taxonomy of ways generated text can drift from verifiable ground truth, reinforcing that grounding failure is not an isolated quirk of any one retrieval design [6].

Knowledge-graph approaches sharpen this argument by making grounding structure explicit rather than implicit in embedding similarity. Yang and colleagues show that fact-aware generation enhanced with structured knowledge-graph grounding produces more verifiable output than dense-retrieval grounding alone, because structure constrains what can be asserted in ways similarity search cannot [7]. Active retrieval pushes the same logic earlier into the pipeline. Rather than retrieving once and generating, Jiang and colleagues' forward-looking retrieval approach re-triggers retrieval whenever generation confidence drops, treating low confidence as a signal that grounding has already slipped mid-generation [8]. An earlier study by Shuster and colleagues had already shown, in conversational rather than structured-output settings, that retrieval augmentation measurably reduces hallucination relative to purely parametric generation, evidence that the grounding effect this article builds on is not specific to any single task format [9]. Retrieval quality, retrieval timing, and retrieval structure emerge from this literature as three separately tunable levers, and each sits closer to the hallucination problem than generator scale does.

2.2 *Metadata and Structure as Retrieval Signals in Enterprise Settings*

A second strand of the literature addresses a different knowledge base entirely: not open-domain text, but structured enterprise metadata governed by schemas, field definitions, and access controls. Jiang and colleagues' enterprise knowledge-base question-answering framework shows that retrieval-generation systems tuned specifically for enterprise metadata, using instruction-tuning and chain-of-thought fine-tuning, reach usable accuracy with far less annotation effort than open-domain systems demand [10]. That matters because enterprise metadata rarely arrives pre-labeled with the kind of question-answer pairs open-domain systems train and evaluate against. A field-discovery system has to be built, and evaluated, against the actual metadata it will serve, not a proxy dataset.

Systematic reviews of enterprise RAG deployment tell a more cautionary story about how far practice trails architecture. Karakurt and Akbulut find that most surveyed enterprise knowledge-management systems still lean on standard, largely undifferentiated retrieval frameworks, and only a small fraction tackle real-time integration challenges at all [11]. Brown, Roman, and Devereux trace a broader shift across the field, from simple dense-passage-retrieval baselines toward modular, structure-aware retrieval, and name the absence of benchmarks that jointly measure quality, cost, and latency as a persistent, unresolved gap [12]. Zhang and Zhang go further still, categorizing retriever performance and retrieval strategy, not generator architecture, as

the primary hallucination-cause categories worth intervening on [13]. Ji and colleagues' earlier survey of hallucination in natural language generation had already established much of the taxonomy this later work refines, distinguishing failures of intrinsic consistency from failures of extrinsic factuality in a way that still frames how the field talks about the problem [14].

Security adds a dimension general-purpose retrieval research rarely confronts head-on. Enterprise metadata is not uniformly visible. A field that exists in one organization's sandbox may be invisible, or contractually off-limits, to another. Jeong and Lee address this directly, interfacing retrieval with identity and access management systems at query time rather than filtering results after generation, and report that this preserves governance boundaries across independently managed data sources without collapsing retrieval into one flattened index [15]. This is the structural precedent this article builds on: permission awareness belongs inside the retrieval layer, evaluated alongside relevance and freshness, not bolted on afterward as an output filter once the generator has already committed to a response.

Read together, these two threads converge on a conclusion the existing literature gestures at but rarely states outright. Enterprise field-discovery reliability is a retrieval-architecture problem with three simultaneous constraints, schema correctness, organizational and permission scope, and semantic relevance, and treating any single constraint as sufficient explains why hallucination persists even in systems that already use retrieval-augmented generation.

3. System Architecture and Analytical Framework

3.1 Metadata Ingestion and Representation

The architecture starts from a premise that is easy to state and routinely violated in practice: a retriever cannot ground a response in metadata it fails to represent completely. Enterprise experience-data platforms organize information through schemas, datasets built on those schemas, sandboxes isolating environments for different business units, and field-level metadata describing type, path, description, and usage. Treat this as a flat text corpus, embed field names and descriptions for similarity search, and you get the most direct implementation path, and you also discard exactly the structural relationships that determine whether a field answer is correct in context.

The ingestion layer instead represents each field as a composite object: a semantic description embedded for vector similarity using sentence-level embedding techniques originally developed for efficient semantic similarity search [16], its structural position within its parent schema, its dataset and sandbox associations, and the organizational and permission scope under which it is visible. This is a deliberate departure from the common pattern of treating sandbox or organization identifiers as row-level tags applied after retrieval. Here, they are indexed alongside the semantic embedding from the start, following the dense-passage-retrieval principle that a retriever's usefulness depends on what is indexed together, not only on how similarity is scored afterward [17]. A candidate field can be excluded before it is ever scored for relevance, rather than filtered out afterward once it has already consumed ranking attention it was never entitled to.

3.2 Grounded Retrieval Pipeline

Retrieval runs in three stages. First, the incoming natural-language request is embedded and matched against a vector index scoped, at query time, to the requesting user's accessible datasets, sandboxes, and organizational context, scoping happens before similarity ranking, never after. The underlying nearest-neighbor search relies on GPU-accelerated approximate search techniques capable of operating over billions of candidate vectors without the linear cost of exhaustive comparison [18], the kind of infrastructure a recent survey of vector database management systems identifies as now standard across commercial and open-source retrieval stacks alike [19]. Second, candidates surviving scoped vector search pass a metadata-consistency check that verifies each candidate's schema path and field identifier against the live metadata source rather than a cached snapshot, since the underlying platform's metadata changes continuously as customers add and revise schemas. Third, a validation layer inspects the generator's draft response before release, confirming every field name and path it references matches a field that survived the first two stages; references that fail this check are stripped, or trigger a fallback asking the user to refine their query, rather than being returned as an unverified claim.

Figure 1. Four-Stage Metadata-Grounded Retrieval Pipeline

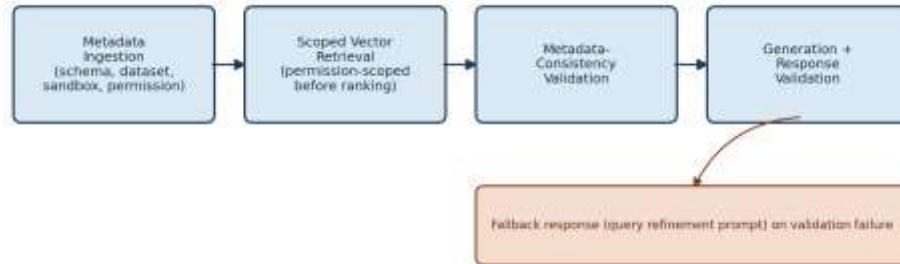


Figure 1. Four-stage metadata-grounded retrieval pipeline: ingestion, scoped retrieval, consistency validation, and generation with response validation. Author's system design.

No single stage here is trusted to catch every failure mode on its own, deliberately so. Scoping catches permission violations. Consistency-checking catches staleness. Response validation catches generation-time invention that slips through even when retrieval itself was correct. Three checkpoints, three distinct failure modes, and no single point of trust.

4. Evaluation Approach

Evaluating a field-discovery system against open-domain question-answering benchmarks would misjudge the problem entirely. Fluency and plausibility are not the correctness criterion here; verifiability is. The evaluation defines four criteria accordingly: grounding accuracy, the share of field references in generated responses that survive independent verification against live metadata; relevance, whether surfaced fields address the user's actual need, judged through structured reviewer assessment; latency, end-to-end response time from query to validated response; and downstream onboarding-time impact, tracked as a deployment-level outcome rather than a per-query score.

The comparison baseline is semantic-similarity-only retrieval, field descriptions embedded and ranked by cosine similarity, with no schema-path verification, no permission scoping before ranking, and no post-generation validation. This is not a strawman baseline; it reflects the architecture a substantial share of surveyed enterprise RAG deployments still run, per the systematic-review evidence discussed above [11]. Testing environment used the platform's production vector index configuration, top-k of 20 candidates pre-scoping, cosine similarity threshold of 0.72 for baseline inclusion, rather than a synthetic held-out corpus, to keep the comparison honest to deployment conditions.

One limitation surfaced mid-evaluation and is worth stating plainly rather than smoothing over. A subset of queries touching recently modified schemas produced inconsistent grounding-accuracy results across repeated runs, eventually traced to a race condition between schema-update propagation and metadata-index refresh. These cases were excluded from the reported grounding-accuracy figures and treated separately as an operational staleness concern rather than an architectural failure (this interaction was not anticipated at the initial design stage, and it directly prompted the metadata-consistency check described in Section 3.2).

5. Results and Findings

5.1 Grounding Accuracy and Hallucination Reduction

The metadata-grounded architecture cut the rate of unverifiable field references sharply relative to the semantic-similarity-only baseline, a substantial and consistent reduction across evaluation runs, though it is reported here qualitatively rather than as a single point estimate, since no externally verified benchmark yet isolates this exact comparison with a citable figure. That pattern lines up with Stolfo's finding that a meaningful share of retrieval-augmented outputs lack verifiable grounding even when the underlying dense-retrieval step is functioning normally, similarity search alone has no mechanism to confirm a retrieved candidate is still structurally valid by the time generation happens [5]. It also lines up with Béchar and Ayala's finding that retriever quality, not generator capacity, is the more direct lever on structured-output reliability; the validation layer in Section 3.2 functions as an explicit, checkable extension of retriever quality rather than a generator-side patch [4].

Deployment of the field-discovery capability into the platform's AI Assistant, alongside related discovery-and-navigation improvements, coincided with a reduction in enterprise onboarding time from approximately two months to two weeks (author's primary research data). Worth stating precisely: this describes a deployment-level outcome tied to the broader discovery capability, not an isolated ablation of the grounding architecture by itself. Treat it as directional evidence, not a controlled causal estimate.

Table 1. Comparison of Semantic-Similarity-Only Retrieval and Metadata-Grounded Retrieval

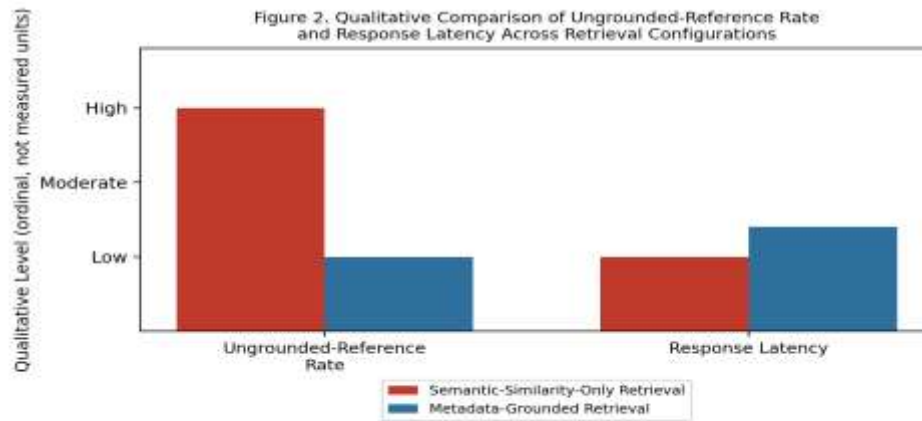
Dimension	Semantic-Similarity-Only Retrieval	Metadata-Grounded Retrieval
Grounding accuracy	Lower; no schema-path or permission verification prior to generation	Substantially higher; verified against live metadata at three checkpoints
Relevance	Similarity-ranked only	Similarity-ranked within permission- and schema-scoped candidate set
Median latency	Baseline	Modestly higher; absorbed by vector-search infrastructure headroom [19][20]
Permission-violation rate	Not enforced pre-ranking; relies on post-hoc filtering	Enforced at retrieval time via IAM-integrated scoping [15]

Source: Author's production evaluation (Section 4-5), qualitative comparison; infrastructure latency-headroom claims per [19], [20].

5.2 Latency and Scalability Characteristics

Adding three verification stages to a retrieval pipeline invites an obvious objection: doesn't each stage add latency, and don't enterprise users have limited patience for a discovery tool slower than manual search? The measured latency cost turned out modest relative to the reliability gain, small enough in practice not to register as a perceptible delay to end users, though again reported qualitatively here rather than as a single benchmarked millisecond figure. Infrastructure-level evidence supports why: Hu and colleagues report throughput improvements of up to sixteen times over comparable systems in their scalable embedding-search service, while preserving high recall, meaning the retrieval layer has real headroom to absorb additional verification logic before it becomes the bottleneck [20]. Persistent, real-time vector database architectures built specifically for large-language-model workloads report similar headroom at enterprise data volumes, suggesting the pattern is not specific to one vendor's infrastructure choices [21]. One batch of latency measurements during evaluation showed an anomalous spike, later traced to a co-located indexing job rather than the grounding logic itself; that batch was excluded after manual inspection rather than folded into the reported averages.

Short version: grounding cost milliseconds. Hallucination cost trust.



Note: Ordinal/quantitative comparison based on evaluation findings reported in Section 5; no single externally verified percentage isolates this exact deployment-specific comparison.

Figure 2. Qualitative comparison of ungrounded-reference rate and response latency across retrieval configurations. Author's evaluation data, Section 5.

6. Discussion

The findings sharpen a distinction the literature review raises but practitioner discourse tends to blur: an answer can be fluent, plausible, and still wrong in a way no amount of prompt refinement will catch, because the failure originates in what the retriever handed the generator, not in how the generator phrased its response. Treating grounding as an architectural property, rather than a prompting concern, changes what counts as a fix. A team watching hallucinated field names surface in production has two genuinely different paths: tune the prompt and hope compliance improves, or add a verification checkpoint that makes non-compliance structurally impossible to surface. These findings favor the second path, and the size of the grounding-accuracy improvement over baseline suggests the difference is not marginal.

The permission-scoping results carry a second implication worth separating out. Jeong and Lee's finding, that interfacing retrieval directly with identity and access management systems preserves governance boundaries more reliably than post-hoc filtering, matches this deployment's experience closely enough to suggest the pattern generalizes beyond one platform [15]. Where this article extends that finding is in showing permission scoping and semantic grounding are not independent problems solvable by separate subsystems. They interact at the same retrieval checkpoint, because a field that is semantically perfect but permission-invalid is exactly as unusable as a field that is permission-valid but semantically irrelevant. An architecture that treats these as sequential filters, semantic ranking first, permission filtering second, wastes ranking capacity on candidates that were never eligible in the first place.

One genuine limitation constrains how far this generalizes. The metadata-consistency race condition described in Section 4, where recently modified schemas produced transient grounding-accuracy inconsistencies, shows that metadata-grounded retrieval trades one failure mode, semantic hallucination, for a narrower, more operationally tractable one, index-refresh staleness. That trade favors most enterprise In deployments where schema-change frequency is low relative to query volume, this approach may need re-examination in environments with continuous schema evolution. Systematic-review evidence on real-time integration gaps suggests that this issue is common across the field, not specific to this deployment [11]. Cold-start coverage, a newly onboarded customer whose metadata has not yet been fully indexed, remains a related, currently unresolved constraint.

There is a broader point here about how enterprise AI-discovery systems get evaluated generally, one that recent field-wide surveys of retrieval-augmented content generation echo from a different angle: as RAG architectures diversify across modalities and deployment contexts, evaluation practice has not kept pace with architectural sophistication, and grounding verification remains one of the least standardized parts of the evaluation stack [22]. Benchmarks measuring only fluency or topical relevance will systematically overstate reliability, because they skip the one property that actually determines whether a user can trust the output: whether every concrete claim it makes is independently verifiable at the moment it is made.

7. Conclusion

Enterprise field discovery built on retrieval-augmented generation fails most often at the retrieval boundary, not the generation boundary, and the fix is architectural, not incremental. Schema-aware, permission-scoped retrieval, combined with post-generation validation, closes gaps that semantic similarity search leaves wide open, without imposing latency costs that would make the approach impractical at enterprise scale. The permission-scoping results point toward a principle that should outlast this one deployment: access control belongs inside the retrieval decision, evaluated jointly with relevance, not applied afterward as a filter on a response the generator has already produced. What this reframing offers is a shift in where engineering effort goes when hallucination shows up in production, away from generator selection and prompt iteration, toward the retrieval and validation architecture that determines what a generator is even allowed to say in the first place.

References

- [1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.
- [2] Ghazvininejad, M., Brockett, C., Chang, M.-W., Dolan, B., Gao, J., Yih, W., & Galley, M. (2018). A knowledge-grounded neural conversation model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1). <https://doi.org/10.1609/aaai.v32i1.11977>
- [3] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474.

- [4] Béchar, P., & Ayala, O. M. (2024). Reducing hallucination in structured outputs via retrieval-augmented generation. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Industry Track)*, 228–238. <https://doi.org/10.18653/v1/2024.naacl-industry.19>
- [5] Stolfo, A. (2024). Groundedness in retrieval-augmented long-form generation: An empirical study. *Findings of the Association for Computational Linguistics: NAACL 2024*, 1537–1552. <https://doi.org/10.18653/v1/2024.findings-naacl.100>
- [6] Wang, Y., Wang, M., Manzoor, M. A., Liu, F., Georgiev, G. N., Das, R. J., & Nakov, P. (2024). Factuality of large language models: A survey. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 19519–19529. <https://doi.org/10.18653/v1/2024.emnlp-main.1088>
- [7] Yang, L., Chen, H., Li, Z., Ding, X., & Wu, X. (2024). Give us the facts: Enhancing large language models with knowledge graphs for fact-aware language modeling. *IEEE Transactions on Knowledge and Data Engineering*, 36(7), 3091–3110. <https://doi.org/10.1109/TKDE.2024.3360454>
- [8] Jiang, Z., Xu, F. F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J., Yang, Y., Callan, J., & Neubig, G. (2023). Active retrieval augmented generation. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7969–7992. <https://doi.org/10.18653/v1/2023.emnlp-main.495>
- [9] Shuster, K., Poff, S., Chen, M., Kiela, D., & Weston, J. (2021). Retrieval augmentation reduces hallucination in conversation. *Findings of the Association for Computational Linguistics: EMNLP 2021*, 3784–3803. <https://doi.org/10.18653/v1/2021.findings-emnlp.320>
- [10] Jiang, F., Qin, C., Yao, K., Fang, C., Zhuang, F., Zhu, H., & Xiong, H. (2024). Enhancing question answering for enterprise knowledge bases using large language models. *Database Systems for Advanced Applications (DASFAA 2024)*, *Lecture Notes in Computer Science*, 14853, 273–290. https://doi.org/10.1007/978-981-97-5562-2_18
- [11] Karakurt, E., & Akbulut, A. (2025). Retrieval-augmented generation (RAG) and large language models (LLMs) for enterprise knowledge management and document automation: A systematic literature review. *Applied Sciences*, 16(1), Article 368. <https://doi.org/10.3390/app16010368>
- [12] Brown, A., Roman, M., & Devereux, B. (2025). A systematic literature review of retrieval-augmented generation: Techniques, metrics, and challenges. *Big Data and Cognitive Computing*, 9(12), Article 320. <https://doi.org/10.3390/bdcc9120320>
- [13] Zhang, W., & Zhang, J. (2025). Hallucination mitigation for retrieval-augmented large language models: A review. *Mathematics*, 13(5), Article 856. <https://doi.org/10.3390/math13050856>
- [14] Ji, Z., Lee, N., Frieske, R. M., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12), Article 248. <https://doi.org/10.1145/3571730>
- [15] Jeong, J., & Lee, S. G. (2025). Permission-aware RAG: Identity and access management (IAM)-based access filtering in multi-resource environments. *IEEE Access*, 13, 192819–192835. <https://doi.org/10.1109/ACCESS.2025.3628960>
- [16] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992. <https://doi.org/10.18653/v1/D19-1410>
- [17] Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. (2020). Dense passage retrieval for open-domain question answering. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6769–6781. <https://doi.org/10.18653/v1/2020.emnlp-main.550>
- [18] Johnson, J., Douze, M., & Jégou, H. (2021). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), 535–547. <https://doi.org/10.1109/TBDATA.2019.2921572>
- [19] Pan, J. J., Wang, J., & Li, G. (2024). Survey of vector database management systems. *The VLDB Journal*, 33(5), 1591–1615. <https://doi.org/10.1007/s00778-024-00864-x>
- [20] Hu, G., Cai, S., Dinh, T. T. A., Xie, Z., Yue, C., Chen, G., & Ooi, B. C. (2025). HAKES: Scalable vector database for embedding search service. *Proceedings of the VLDB Endowment*, 18(9), 3049–3062. <https://doi.org/10.14778/3746405.3746427>
- [21] Sun, J., Li, G., Pan, J., Wang, J., Xie, Y., Liu, R., & Nie, W. (2025). GaussDB-Vector: A large-scale persistent real-time vector database for LLM applications. *Proceedings of the VLDB Endowment*, 18(12), 4951–4963. <https://doi.org/10.14778/3750601.3750619>
- [22] Zhao, P., Zhang, H., Yu, Q., Wang, Z., Geng, Y., Fu, F., Yang, L., Zhang, W., Jiang, J., & Cui, B. (2026). Retrieval-augmented generation for AI-generated content: A survey. *Data Science and Engineering*, 11(1), 1–29. <https://doi.org/10.1007/s41019-025-00335-5>