Svedberg Open
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

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



Research Paper Open Access

Personalized Retrieval-Augmented Generation For Human-AI Collaborative Data Analysis

Santosh Reddy Muthyala

Independent Researcher, USA.

Abstract
Retrieval-Augmented Generation (RAG) has become the dominant approach for grounding large language model outputs in verified, domain-specific knowledge, reducing hallucinated responses by over 60 percent relative to non-retrieval baselines [12]. Current RAG systems for enterprise analytics, however, treat every user identically: a senior data engineer and a business analyst asking the same question receive the same retrieval, ranking, and generated response, despite needing fundamentally different things. This article argues that personalization is not an optional refinement but a structural requirement for analytics RAG, since the right answer in data analysis depends on who is asking, what they already know, and what they intend to do with the result. We propose a personalized RAG architecture built on three components: a User Profile Store combining explicit and implicit signals, a Multi-Index Retrieval System routing queries across specialized schema, metric-definition, past-query, documentation, and example indices, and a Personalized Generation Layer that adapts response format and depth to the individual analyst. Drawing on personalized-search literature showing 10 to 31 percent MAP improvements from personalized re-ranking [14], and on the author's own experience building a comparable multi-domain retrieval-augmented analytics assistant in production, we argue this architecture is both theoretically grounded and practically achievable. We close with a proposed evaluation methodology for personalized analytics RAG, an area with no existing standardized benchmark.

Keywords: retrieval-augmented generation; personalization; human-AI collaboration; enterprise analytics; information retrieval; multi-index architecture.

1. Introduction

Retrieval-Augmented Generation grounds large language model outputs in external, verified knowledge rather than relying solely on parametric memory. The approach, introduced by Lewis et al. [1], retrieves relevant documents from a knowledge base and injects them into the model's context before generation. The effect on factual reliability is substantial: retrieval augmentation has been shown to reduce hallucinated responses by over 60 percent relative to non-retrieval baselines, with the effect more pronounced still on out-of-distribution topics, where knowledgeability gains reach 85 percent compared to 70 percent for in-distribution content [12].

What current RAG systems do not do is distinguish between the people asking the questions. The same query, show me revenue trends, triggers identical retrieval and generation regardless of whether it comes from a data engineer who wants a SQL statement, a business analyst who wants a narrative with charts, or a product manager who thinks in terms of organizational KPIs and historical baselines. This uniformity matters more for data analysis than for most other RAG applications, because in analytics the correctness of an answer is inseparable from its audience: an explanation of a retention metric that is appropriately rigorous for a data scientist, complete with confidence intervals and cohort definitions, is simply the wrong answer for an executive who needs one sentence and a trend arrow.

The stakes of getting this wrong are not merely stylistic. When a system consistently returns responses miscalibrated to a user's actual expertise and intent, the friction compounds over repeated interactions. A data engineer forced to parse a narrative explanation to extract the underlying query logic wastes time on every single query; an executive shown a raw SQL statement disengages entirely. Analytics teams evaluating whether to adopt an AI assistant are, in practice, evaluating whether the tool respects the diversity of roles inside their

organization, and a generic RAG deployment answers that question poorly by construction, independent of how capable the underlying language model is.

This article makes the case that personalization is a structural requirement, not a convenience feature, for analytics RAG, and proposes an architecture for delivering it. Section 2 positions this argument against three adjacent literatures: general RAG systems, personalized information retrieval, and RAG for text-to-SQL. Section 3 examines why analysts are not a homogeneous user population and reviews the quantitative evidence for personalization's impact. Section 4 presents the proposed architecture across three components: a user profile store, a multi-index retrieval system, and a personalized generation layer, and grounds the architecture in a comparable production system the author has built. Section 5 discusses the resulting shift in human-AI collaboration dynamics and the trust mechanisms personalization supports. Section 6 addresses cold-start, privacy, and scalability challenges, and proposes an evaluation methodology for a research area that currently has none.

2. Related Work and Contribution

Personalized RAG for analytics sits at the intersection of three established research areas, and it is worth being precise about what each area has already established and where this article's contribution actually lies.

2.1 RAG Systems

The RAG paradigm itself is well established. Lewis et al.'s foundational formulation [1] and subsequent comprehensive surveys [10] cover the core architecture in depth. A family of advanced variants has emerged to address specific weaknesses in the basic retrieve-then-generate pattern. Self-RAG lets the model decide when retrieval is necessary rather than retrieving unconditionally on every query, using reflection tokens generated during training to signal when external context would help [6]. Adaptive-RAG classifies incoming queries by complexity and selects a retrieval strategy accordingly, avoiding unnecessary retrieval overhead on simple factual questions while escalating to multi-step retrieval for genuinely complex queries [7]. Corrective RAG adds a layer that assesses the quality of retrieved passages and triggers a fallback, such as a web search, when the knowledge base retrieval proves insufficient [8]. GraphRAG builds a knowledge graph from the underlying corpus and uses hierarchical community summaries to support synthesis across multiple documents, an approach that outperforms naive vector retrieval specifically on queries requiring this kind of cross-document synthesis [9].

None of these systems, despite their sophistication, adapts to who is asking. They optimize retrieval and generation for an undifferentiated average user, treating the question posed by a first-time visitor and a domain expert as equivalent inputs to be resolved through the same retrieval and generation pipeline. This is a deliberate simplification in the design of each of these systems, not an oversight, since their contribution lies elsewhere, in when to retrieve, how to assess quality, or how to structure the knowledge base. But the simplification means none of them addresses the problem this article is concerned with.

2.2 Personalized Information Retrieval

Personalized search has a substantially longer research history than RAG itself, predating the current generation of large language models by well over a decade. Dou et al. demonstrated meaningful mean average precision gains from personalized re-ranking strategies in web search, with the magnitude of improvement varying considerably by query type and ambiguity [14]. White et al. showed that domain expertise measurably shapes search behavior, with expert searchers issuing shorter, more specific queries and engaging more productively with deeper technical content than novice searchers presented with the same material [13]. Bennett et al. modeled how short- and long-term interaction signals combine to drive personalization quality over time, establishing that both recent session behavior and longer historical patterns contribute independently to personalization accuracy [15]. The LaMP benchmark established an evaluation methodology specifically for personalized text generation with large language models, covering tasks including headline generation, email completion, and citation identification [5].

This body of work is directly relevant to the architecture proposed here, and several of its mechanisms, personalized re-ranking in particular, are incorporated directly into Section 4. But its focus is general personalization: the artifacts these systems retrieve and rank are web pages, emails, or generic documents, and the personalization signal being learned is a general preference model rather than anything specific to the demands of analytics workflows. The artifacts an analytics RAG system must retrieve, database schemas, organizational metric definitions, past query history, differ categorically from web pages, and the artifacts it must generate, SQL statements, narrative interpretations, executive summaries, differ categorically from the writing-assistance or search-result-ranking tasks this literature typically addresses.

2.3 RAG for Analytics and Text-to-SQL

Text-to-SQL systems increasingly use retrieval to improve generation accuracy. DAIL-SQL retrieves similar worked SQL examples to guide generation, using the retrieved examples as in-context demonstrations for the generating model [11]. Related multi-agent approaches to text-to-SQL extend this pattern further, decomposing the generation task across multiple specialized reasoning steps. None of this work personalizes retrieval or generation based on the individual analyst asking the question; a junior analyst and a senior data engineer submitting the identical natural-language query receive the identical retrieved examples and the identical generated SQL, regardless of whether the junior analyst would benefit from a more heavily annotated query or the senior engineer would prefer a terser one.

2.4 What Is New

Against this background, this article makes five contributions not covered by the literature surveyed above. First, an explicit argument that personalization is a structural, not optional, requirement specifically for analytics RAG, grounded in the observation that analyst expertise, domain, and intent fundamentally determine what constitutes a useful response, a case no prior work makes specifically for the data-analysis setting. Second, an analyst persona framework formalizing how different analyst types require different retrieval strategies, response formats, and depth levels, an explicit formalization the general personalization literature acknowledges as a concern but does not construct for this domain. Third, a multi-index retrieval architecture purpose-built for the specific knowledge types analytics workflows depend on: schemas, metric definitions, past queries, documentation, and curated examples, each requiring a distinct retrieval strategy. Fourth, a personalized re-ranking and generation pipeline integrating existing component techniques, re-ranking, adaptive generation, into a single analytics-specific system, where the novelty lies in the integration and domain-specific routing rather than in any individual component. Fifth, a proposed evaluation methodology for personalized analytics RAG, an intersection for which no standardized benchmark currently exists.

3. The Case for Personalization in Analytics RAG

3.1 Analysts Are Not a Homogeneous Group

Enterprise data analysts vary along dimensions that directly determine what a RAG system should retrieve and how it should present results. Table 1 summarizes these differences across three representative analyst types.

Table 1. Analyst Persona Dimensions

Dimension	Senior Data Engineer	Business Analyst	Product Manager
Query style	Short, precise, technical	Verbose, natural language	Business-framed questions
Expected output	SQL query, brief explanation	Narrative with visualizations	KPI summary with trends
Schema knowledge	Deep - knows table structures	Moderate - knows key tables	Minimal - thinks in metrics
Retrieval need	Precise: specific table docs	Broad: context and examples	Targeted: metric definitions
Appropriate depth	Technical, terse	Explanatory, contextual	Executive, actionable

A senior data engineer issues short, precise, technical queries and expects a SQL statement with a brief explanation, drawing on deep schema knowledge accumulated over years of working directly with the underlying tables. A business analyst issues more verbose, natural-language queries and expects a narrative response with visualizations, drawing on moderate familiarity with the key tables relevant to their domain but little interest in join logic or column-level detail. A product manager frames questions in business terms and expects a KPI summary with trend context, typically without direct schema knowledge at all, thinking instead in terms of the metrics that define their product's success.

A generic RAG system receiving the query shows me revenue trends cannot distinguish among these three users, and so retrieves the same documents and generates the same response regardless of who asked. White et al.'s empirical work on domain expertise and search behavior supports this distinction directly: expert users issue measurably shorter, more specific queries and engage more productively with deeper technical content,

while novice users benefit from broader context and more explanatory framing [13]. The gap this creates in a generic system is not symmetric: a novice user shown expert-level technical detail is confused, while an expert user shown novice-level explanatory framing is merely inefficient, wasting time scanning past content they do not need. Both failure modes erode the perceived value of the system, but they erode it differently, and a personalization strategy needs to address both directions rather than optimizing only for one.

3.2 Quantitative Evidence for Personalization

The case for personalization rests on more than intuition. The LaMP benchmark, the most established evaluation framework for personalized text generation with large language models, demonstrates that retrieving from user-specific history meaningfully outperforms generic retrieval across multiple generation tasks [5], though the precise magnitude varies by task and should be verified against the specific downstream application rather than treated as a universal constant. In the personalized-search literature more broadly, Dou et al. found substantial mean average precision gains, ranging from 10 to 31 percent depending on query type, from incorporating personalized re-ranking signals [14]. Bennett et al.'s modeling work found that a meaningful share of the maximum achievable personalization benefit is realized within the first several dozen user interactions, suggesting the cold-start period, while real, is bounded rather than indefinite [15].

Industry case studies from production RAG deployments, published through engineering blogs and conference presentations rather than peer review, offer directional but less rigorously validated evidence in the same direction: reported improvements in answer relevance and reductions in follow-up clarification queries when systems incorporate user role and team context into retrieval. These figures should be read as indicative of the general direction of the effect in production systems rather than as precise, reproducible benchmarks, since they were not independently validated under controlled experimental conditions and may reflect favorably selected configurations. Treating peer-reviewed and industry-reported evidence with different levels of confidence is itself methodologically important here: the peer-reviewed literature establishes that the underlying effect (personalization improves retrieval and generation quality) is real and reproducible, while the industry evidence suggests the effect persists at meaningful magnitude in production analytics contexts specifically, without establishing a precise, generalizable number.

4. Architecture of a Personalized RAG Data Platform

The architecture proposed here rests on three components: a User Profile Store, a Multi-Index Retrieval System, and a Personalized Generation Layer.

4.1 User Profile Store

The User Profile Store captures analyst profiles through two signal types. Explicit signals, collected once and updated periodically, include declared role and domain, primary data domains of interest, technical proficiency, and preferred response format. Implicit signals, learned continuously from interaction, include query complexity patterns, dwell time on retrieved passages, a documented proxy for relevance in the broader information-retrieval literature [21], copy-paste behavior indicating which parts of a response were actually useful, query reformulation patterns, and the specific tables, columns, and metrics an analyst most frequently accesses.

The cold-start problem, in which new users have no interaction history to draw on, is addressed through role-based default profiles and organizational-hierarchy inheritance, where a new team member's profile begins as their team's aggregate pattern rather than a blank slate. This mitigation is consistent with the broader recommender-systems literature on cold-start users, where collaborative signals from related users have been shown to meaningfully improve outcomes for individuals with no history of their own [23].

4.2 Multi-Index Retrieval System

Rather than a single undifferentiated vector index, the proposed system maintains specialized indices for distinct knowledge types, summarized in Table 2.

Table 2. Multi-Index Retrieval System

Index	Contents	When Used
Schema Index	DDL statements, column descriptions, join documentation	Schema lookups (e.g. which tables have revenue data)
Metric Dictionary	Organizational metric definitions, computation logic, ownership	Metric questions (e.g. how is DAU defined)

Past Query Index	Team and individual query history with results	How-to questions (e.g. how was churn computed last quarter)
Documentation Index	Analytical reports, runbooks, data dictionaries	Exploratory questions, background context
Example Index	Curated SQL examples for common analytical patterns	SQL generation tasks

A routing layer directs each incoming query to the appropriate index or combination of indices based on detected intent. A query asking which tables contain revenue data routes to the Schema Index; a query asking how daily active users are defined routes to the Metric Dictionary; a query asking how churn was computed last quarter routes to the Past Query Index. Many real analyst queries require combining signals from more than one index simultaneously, for instance a query that needs both the correct metric definition and a similar prior query demonstrating how that definition was implemented in SQL, and the routing layer must support weighted combination across indices rather than a strict single-index dispatch.

After routing, a personalized re-ranking layer re-scores retrieved candidates using a weighted combination of three signals, expressed as score equals alpha times semantic similarity plus beta times user relevance plus gamma times recency, where the weights alpha, beta, and gamma are tuned per deployment to reflect how much a given organization wants to prioritize topical match, individual user history, or content freshness. The user-relevance term draws on the analyst's historical interaction with similar content: documents a given user has previously found useful are boosted for that user specifically, while documents from domains the user has consistently ignored are demoted, without affecting the ranking any other user sees for the same underlying query. This draws directly on the personalized re-ranking literature, which has demonstrated meaningful retrieval-quality gains from incorporating exactly this kind of user-relevance signal [14]. Retrieval methods underlying this layer can draw on dense, sparse, or hybrid approaches: dense bi-encoder and late-interaction models such as ColBERT achieve strong passage-retrieval quality through contextualized token-level matching that outperforms single-vector dense retrieval on standard passage-ranking benchmarks [3], while hybrid approaches that combine dense and sparse signals are increasingly the production standard because they capture complementary strengths, dense retrieval for semantic similarity and sparse retrieval for precise lexical matching on domain-specific terminology such as table and column names, that neither approach alone reliably provides. A cross-encoder re-ranking pass applied after initial candidate retrieval, conditioned on the user's profile embedding rather than the query alone, provides a further refinement step at modest additional latency cost, since cross-encoders can attend jointly to the query, the candidate passage, and user-specific features in a way a bi-encoder's independently-computed embeddings cannot.

4.3 Personalized Generation Layer

The generation layer adapts response style and depth to the user profile. Technical users receive a SQL query first with a brief explanation, expressed in precise terminology and referencing specific tables and columns. Business users receive a narrative interpretation first, with the underlying SQL available on expansion, framed in business language that connects data patterns to business drivers. Executive users receive a one-sentence summary with a trend indicator, with supporting detail available on request. Response verbosity itself adapts continuously: users who consistently disregard verbose explanations receive increasingly concise responses over time, while users who frequently ask follow-up clarification questions receive more detailed initial responses.

The system improves through three feedback mechanisms operating together. Explicit feedback, in the form of simple approval or rejection signals on generated answers, trains both the retriever and the user profile directly. Implicit feedback, drawn from copy-paste behavior, follow-up queries, and session abandonment, provides a continuous signal without requiring any additional user effort. Active learning, where the system asks a clarifying question rather than guessing when retrieval ambiguity is high, has been shown to improve answer quality specifically for genuinely ambiguous queries, where guessing incorrectly is more costly than the minor friction of an additional clarifying exchange [16].

4.4 Validation Through Production Experience

The architecture proposed above is not a purely theoretical construction. The author has built and operated a comparable system in production: a personalized retrieval-augmented platform spanning four distinct data domains, specifically SQL query history, dashboards, real-time analytics, and code notebooks, at a billion-user-scale technology platform, transforming what had been a generic internal analytics agent into a context-aware assistant used across the organization. This experience directly informed the multi-index design proposed in

Section 4.2: maintaining separate, purpose-specific indices for these four domains, rather than a single undifferentiated index, proved necessary in practice because the retrieval and ranking signals that matter for a SQL-history lookup differ categorically from those that matter for a dashboard reference or a real-time analytics query. A single shared index tuned for one domain's retrieval characteristics systematically underperformed on the others, which is the practical, production-tested motivation behind the specialized-index design this article proposes as a general architecture rather than a domain-specific workaround.

5. Impact on Human-AI Collaboration

5.1 From Tool to Collaborative Partner

Generic RAG positions the AI system as a search engine: the analyst asks, the system retrieves, the model answers. Personalized RAG shifts this dynamic toward genuine collaboration. A system that understands an analyst's domain, current project, expertise level, and preferred working style can anticipate follow-up questions based on typical workflow patterns, proactively surface related documentation the analyst has not yet requested, calibrate how prominently it flags uncertainty depending on the user's expertise in the relevant domain, and adapt iteratively within a single session as early interactions inform later responses. None of these capabilities is available to a system that treats every incoming query identically, since each depends on maintaining and reasoning over a persistent model of who the user is and what they have already been shown.

5.2 Trust Through Transparency and Personalization

Whether analysts adopt an AI tool or abandon it after the initial novelty wears off depends substantially on trust, and two findings from the literature are directly relevant here. Providing source citations alongside generated answers measurably increases user trust, though the effect is more nuanced than a simple monotonic relationship: a controlled study found that citations increase self-reported trust even when the citations themselves are effectively random, while participants who actually verified the citations they were shown reported lower trust afterward [19]. This suggests citation-based trust in generated analytics answers should be built on genuinely relevant, verifiable sources rather than citation presence alone, since sophisticated users, and enterprise analysts are typically sophisticated users, will eventually check.

Separately, work on miscalibrated AI confidence shows that communicating a system's confidence calibration helps users detect when a system is unreliable, but this communication can itself reduce trust and push users toward under-reliance rather than appropriately calibrated reliance [22]. This is a genuinely asymmetric finding worth dwelling on: the naive assumption that more transparency about uncertainty straightforwardly improves outcomes does not hold, and a personalized analytics RAG system needs to calibrate not just its retrieval and generation to the user, but how and when it communicates its own confidence, since the same disclosure that helps one user detect an unreliable answer may push another user toward ignoring correct answers unnecessarily.

Personalization contributes to trust through a distinct mechanism from either citation or confidence signaling: when a system consistently responds at the right level of technical depth and in the right format, users develop confidence that the system understands their specific working context, which reduces interaction friction and increases willingness to rely on it for progressively more complex tasks. This mechanism compounds over time in a way citation-based trust does not: a single well-cited but poorly-formatted answer does little to build confidence in a business analyst who has to parse through SQL syntax to find their answer, while a consistently well-formatted answer, even without extensive citation, builds the specific kind of confidence that predicts continued adoption.

6. Challenges and Future Directions

6.1 Cold-Start Problem

New users present no interaction history to personalize against. Practical mitigations include role-based default profiles as reasonable starting points, organizational inheritance so new team members begin from their team's aggregate pattern, collaborative-filtering approaches that bootstrap recommendations from similar users' patterns, an approach with demonstrated value for cold-start users in the broader recommender-systems literature [23], and progressive profiling, where the system explicitly asks preference questions during early interactions to accelerate profile construction. The cold-start window, while real, appears to be bounded rather than open-ended based on the personalization-modeling literature [15].

6.2 Privacy and Data Sensitivity

Storing detailed interaction histories raises legitimate concerns. User profiles containing query patterns and behavioral signals constitute personal data under frameworks such as GDPR and CCPA, requiring right-to-delete capabilities and defined retention limits. Analyst queries may also reference confidential business information, requiring encrypted, access-controlled profile storage. Differential privacy, applying calibrated noise to protect individual query patterns per the foundational framework of Dwork et al. [4], offers one mitigation path, though it necessarily trades off against personalization fidelity, and the precise magnitude of that trade-off in an analytics RAG context specifically remains an open empirical question rather than a settled figure.

6.3 Scalability

Maintaining per-user profiles and retrieval state at enterprise scale requires deliberate architecture choices: lightweight, user-specific adapters rather than full model copies per user, a shared base index with personalized re-ranking layers to avoid duplicating the entire document corpus per user, and incremental index maintenance for user-specific query history that avoids full re-indexing on every new interaction. Personalized re-ranking introduces additional latency overhead, which for interactive analytics with tight end-to-end latency budgets requires optimized re-ranking models and aggressive caching of user profile representations.

6.4 A Proposed Evaluation Methodology

No standardized benchmark exists for personalized analytics RAG specifically; LaMP [5] establishes methodology for general personalization tasks but nothing tailored to data-analysis workflows. This article proposes an evaluation methodology spanning three aspects of personalization quality.

Retrieval personalization quality can be measured through personalized ranking metrics computed against user-specific relevance judgments rather than generic relevance, since two users asking an identical question may have genuinely different relevant passages depending on their domain and expertise; through a role-appropriate retrieval rate measuring what fraction of retrieved documents match the user's expertise level; and through a diversity measure tracking whether the system still surfaces unexpected but relevant information or has collapsed into a filter bubble of familiar-domain content.

Generation personalization quality can be measured through format appropriateness, whether the response format matches the user's preferred format, evaluated by human judges or an LLM-as-judge protocol against the user's profile; through depth calibration, whether technical depth is appropriate for the user's expertise, evaluated through satisfaction surveys or through the proxy metric of follow-up clarification rate; and through terminology adaptation, whether the system's vocabulary matches the user's own historical usage.

Longitudinal personalization quality can be measured through a learning curve tracking personalization quality across increasing interaction counts, through adaptation speed when a user's role or domain changes, and through stale-profile detection measuring the delay between an actual shift in user behavior and a corresponding update to their profile.

A test suite adequate to this methodology should include a substantial number of analyst profiles spanning multiple roles and domains, each with a meaningful history of prior interactions, alongside a large pool of test queries carrying per-profile relevance judgments, including both role-specific queries where different users should receive different responses and role-neutral queries where all users should receive the same factual answer, to confirm personalization does not introduce bias into purely factual responses.

6.5 Open Research Questions

Several questions remain genuinely open. No standardized benchmark yet exists for this specific intersection of personalization and analytics. The filter-bubble risk, where personalization that always retrieves familiar content prevents analysts from discovering relevant information outside their usual domain, requires balancing exploitation of known preferences against exploration of unexpected but relevant material, an unresolved design tension. Nearly all published personalization evaluations cover single sessions or a small number of sessions; no study has tracked personalization quality over months of sustained real-world use, leaving open how profile accuracy evolves, whether personalization quality plateaus or continues improving, and whether user behavior itself changes as the system adapts. Finally, analysts work across text, SQL, charts, and dashboards, and personalizing consistently across all of these modalities, knowing which users prefer tabular output versus visualizations, SQL versus a higher-level query language, remains largely unexplored.

7. Conclusion

Generic RAG systems deliver the same response regardless of who is asking, a design choice that leaves substantial value on the table for data analysis workflows specifically, where the correctness of an answer is inseparable from the analyst's context, expertise, and intent. Personalized RAG addresses this by integrating user profiles into query understanding, retrieval, re-ranking, and generation, delivering the right level of detail, the right format, and the right context for each individual analyst. The architecture proposed here, multi-index retrieval with personalized re-ranking, adaptive generation, and continuous feedback loops, is not merely a theoretical proposal: a comparable system built and operated in production, spanning four distinct analytics data domains, demonstrates that this architecture is achievable incrementally on top of existing RAG infrastructure rather than requiring a wholesale redesign. The challenges of cold start, privacy, and scale are real but appear bounded rather than intractable based on the evidence surveyed here. The deeper research frontier is building personalization that adapts consistently across the full spectrum of analytical work, SQL generation, metric interpretation, experimentation analysis, and visualization, calibrated continuously to each analyst's evolving expertise.

Author Contributions
Conceptualization, methodology, investigation, writing (original draft, review and editing): the sole author.

Conflicts of Interest
The author declares no conflict of interest.

Declaration on the Use of Generative AI
The author used Claude (Anthropic, claude-sonnet model), accessed July 2026, to assist with manuscript drafting and language refinement only, with no use in data analysis or code generation. The author takes full responsibility for the accuracy, originality, and integrity of all content presented in this manuscript.

References

[1] P. Lewis, E. Perez, A. Piktus, et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, 2020. arXiv:2005.11401.
[2] V. Karpukhin, B. Oguz, S. Min, et al., "Dense Passage Retrieval for Open-Domain Question Answering," in Proc. EMNLP, 2020. arXiv:2004.04906.
[3] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT," in Proc. SIGIR, 2020. DOI: 10.1145/3397271.3401075.
[4] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," in Proc. TCC, 2006.
[5] A. Salemi, S. Mysore, M. Bendersky, and H. Zamani, "LaMP: When Large Language Models Meet Personalization," arXiv:2304.11406, 2023.
[6] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," in Proc. ICLR, 2024. arXiv:2310.11511.
[7] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. C. Park, "Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity," in Proc. NAACL, 2024. DOI: 10.18653/v1/2024.naacl-long.389.
[8] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, "Corrective Retrieval Augmented Generation," arXiv:2401.15884, 2024.
[9] D. Edge, H. Trinh, N. Cheng, et al., "From Local to Global: A Graph RAG Approach to Query-Focused Summarization," arXiv:2404.16130, 2024.
[10] Y. Gao, Y. Xiong, X. Gao, et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv:2312.10997, 2024.
[11] D. Gao, H. Wang, Y. Li, et al., "Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation," arXiv:2308.15363, 2023.
[12] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston, "Retrieval Augmentation Reduces Hallucination in Conversation," in Findings of EMNLP, 2021. DOI: 10.18653/v1/2021.findings-emnlp.320.
[13] R. W. White, S. T. Dumais, and J. Teevan, "Characterizing the Influence of Domain Expertise on Web Search Behavior," in Proc. WSDM, 2009.
[14] Z. Dou, R. Song, and J. Wen, "A Large-scale Evaluation and Analysis of Personalized Search Strategies," in Proc. WWW, 2007. DOI: 10.1145/1242572.1242651.

[15] P. N. Bennett, R. W. White, W. Chu, et al., "Modeling the Impact of Short- and Long-Term Behavior on Search Personalization," in Proc. SIGIR, 2012.
[16] H. Zamani, S. T. Dumais, N. Craswell, P. Bennett, and G. Lueck, "Generating Clarifying Questions for Information Retrieval," in Proc. WWW, 2020.
[17] R. Nogueira and K. Cho, "Passage Re-ranking with BERT," arXiv:1901.04085, 2019.
[18] E. Kamalloo, A. Jafari, X. Zhang, N. Thakur, and J. Lin, "HAGRID: A Human-LLM Collaborative Dataset for Generative Information-Seeking with Attribution," arXiv:2307.16883, 2023.
[19] H. Ding, M. Facciani, A. Poudel, et al., "Citations and Trust in LLM Generated Responses," in Proc. AAAI, 2025. DOI: 10.1609/aaai.v39i22.34550.
[20] S. Mysore, Z. Lu, M. Wan, L. Yang, B. Sarrafzadeh, S. Menezes, T. Baghaee, E. Barajas Gonzalez, J. Neville, and T. Safavi, "PEARL: Personalizing Large Language Model Writing Assistants with Generation-Calibrated Retrievers," arXiv:2311.09180, 2023.
[21] Y. Kim, A. Hassan, R. W. White, and I. Zitouni, "Modeling Dwell Time to Predict Click-Level Satisfaction," in Proc. WSDM, 2014.
[22] J. Li, Y. Yang, R. Zhang, Q. V. Liao, T. Song, Z. Xu, and Y. Lee, "Understanding the Effects of Miscalibrated AI Confidence on User Trust, Reliance, and Decision Efficacy," arXiv:2402.07632, 2024.
[23] X. Zhang, Z. Kuang, Z. Zhang, F. Huang, and X. Tan, "Cold & Warm Net: Addressing Cold-Start Users in Recommender Systems," in Proc. DASFAA, Lecture Notes in Computer Science, vol. 13946, Springer, 2023, pp. 532-543. DOI: 10.1007/978-3-031-30678-5_40.