The American Journal of Engineering and Technology
ISSN 2689-0984 Volume 08 - 2026

Comparative Analysis of RAG Algorithms and LLM Fine-Tuning Methods for Domain-Specific Search Tasks

Kapil Verma
Software Engineer, Google, Mountain View, CA, USA

Received: 12 Jan 2026 | Received Revised Version: 18 Feb 2026 | Accepted: 21 Mar 2026 | Published: 04 Apr 2026

Volume 08 Issue 04 2026 | Crossref DOI: 10.37547/tajet/Volume08Issue04-03

Abstract

The article examines the comparative properties of Retrieval-Augmented Generation algorithms and large-language-model fine-tuning methods in the context of domain-specific search tasks with a high cost of error. The aim is to identify operating regimes in which RAG and fine-tuning differentially affect the accuracy of top-ranked results, the evidential quality of answers, and the safety of handling sensitive data. The relevance of the study is driven by the rapid growth of industrial domain-specific search systems that must simultaneously ensure knowledge updatability, strict citation-based verifiability, and regulatory discipline. The novelty lies in the fact that the comparison is conducted not in the abstract form of RAG versus fine-tuning, but at the level of individual pipeline components and from the perspective of operational trade-offs: it is shown that retrieval and ranking form a truth scaffold and a channel for knowledge refresh, whereas fine-tuning acts as a delicate regulator of format, terminology, and epistemic precision without resolving the problem of obsolescence in parametric representations. The article concludes in favor of hybrid schemes that combine hybrid retrieval, reranking, and strict citation rules with lightweight, parameter-efficient model adaptations, thereby enabling reproducible, controllable, and scalable operation of domain-specific search systems. The article is intended for researchers in information retrieval, engineers of applied RAG systems, and practitioners deploying generative models in high-risk domains.

Keywords: domain-specific search, Retrieval-Augmented Generation, LLM fine-tuning, hybrid retrieval, reranking

Cite This Article: Verma, K. (2026). Comparative Analysis of RAG Algorithms and LLM Fine-Tuning Methods for Domain-Specific Search Tasks. The American Journal of Engineering and Technology, 8(4), 32–40. https://doi.org/10.37547/tajet/Volume08Issue04-03

Introduction

Domain-specific search is fundamentally more complex than general web search, not because of data volume but because of the density of meaning and the cost of error: queries often rely on narrow terminology, internal abbreviations, document versioning, and context that is not expressed literally. Under such conditions, a system must generalize without losing evidential grounding and must transfer behavior to new parts of the collection where formulations change, and the topical distribution diverges from the training data. This is clearly visible in cross-domain evaluations of search systems: in a heterogeneous benchmark of 18 collections, robust lexical methods remain a strong baseline, while more complex neural variants often achieve gains at the price of substantially higher computational load (Thakur et al., 2021).

From this, follow the key requirements of domain-specific search: high precision in the top-ranked results, citability and verifiable reliance on sources, temporal and

version-aware freshness, and secure handling of sensitive data, adhering to regulatory constraints. For generative systems, this implies that a fluent answer is insufficient: what is needed is an answer that can be decomposed into supporting fragments and reproduced on repeated runs. At the same time, even architectures that inject retrieved fragments into the answer generation process remain vulnerable to leakage from the retrieval corpus and to privacy attacks if access control and filtering mechanisms are not embedded into the design (Zeng et al., 2024).

This article compares two families of solutions and their combinations: the retrieval-augmented generation approach, in which the model answers based on retrieved documents, and the fine-tuning approach, in which the model's behavior is modified (ranging from supervised instruction tuning to preference-based training and parameter-efficient adaptations). Surveys on retrieval-augmented generation show that this class of methods systematically addresses the problem of knowledge updating and improves reliability via external context, whereas fine-tuning more often improves answer discipline, style, robustness to frequent query patterns, and instruction following, but does not substitute for access to up-to-date data (Gao et al., 2023).

In what follows, domain-specific search is understood to encompass scenarios ranging from navigation in internal knowledge bases and procedure lookup to questions about standard requirements and the alignment of fragments from multiple documents; query types range from short find-the-definition prompts to composite requests such as compare two solutions and state the constraints. To keep the analysis operational, the task is decomposed into three stages: candidate retrieval from the collection, ranking to select the most useful fragments, and answer generation grounded in the selected context. In real deployments, this inevitably confronts constraints on latency, computational cost, confidentiality, and compliance with sectoral requirements: for example, fine-tuning via low-rank adaptations reduces the storage and update cost of specialized model variants, and large instruction datasets demonstrate that high-quality behavioral tuning can deliver notable gains on multiple tasks without a proportional increase in computational expenditure (Hu et al., 2021).

Materials and Methodology

The study materials comprise a corpus of 14 works selected according to the criterion of direct applicability to domain-specific search and to the comparison of RAG and fine-tuning: a cross-domain robustness benchmark used to fix a strong baseline for lexical search and the cost of complicating neural methods (Thakur et al., 2021); a survey of Retrieval-Augmented Generation as a class of architectures that separates updatable collection knowledge from parametric model knowledge (Gao et al., 2023); and a set of articles on key nodes of the RAG pipeline, segmentation/chunking (Merola & Singh, 2025), hybrid fusion of relevance signals (Bruch et al., 2024), reranking (Pradeep et al., 2022), and multi-step retrieval (Jiang et al., 2024). The fine-tuning block includes parameter-efficient adaptation methods that are particularly important for domain deployment and behavioral versioning (Hu et al., 2021), as well as works showing how LLMs can be used as instruments of query expansion, thereby increasing recall while introducing an additional error channel (Jagerman et al., 2023). To ensure operational honesty, that is, answer reproducibility and verifiable grounding in sources, additional materials are used on reducing hallucinations in structured outputs by enforcing contextual grounding (Ayala & Bechard, 2024), and on privacy risks and leakage in RAG as a consequence of access to the retrieval corpus and weak context control (Zeng et al., 2024), which directly intersects with domain-specific requirements for security and access control.

The methodology is structured as a comparative, modularly decomposed analysis in which the objects being compared are families of algorithms within the three stages of domain-specific search. The first stage is candidate retrieval, the second is ranking/reranking, and the third is answer generation with explicit source attribution. This decomposition enables distinguishing root causes of quality degradation from the effects of merely cosmetic improvements in model behavior (Gao et al., 2023). Within the retrieval stage, sparse, dense, and hybrid schemes are compared with an emphasis on the precision@k versus computational cost trade-off; hypotheses concerning the robustness of the baseline are examined with reference to the cross-domain BEIR picture (Thakur et al., 2021), and hybridization quality is treated as a function of the chosen fusion rule and the query distribution (Bruch et al., 2024). For reranking and generation, latency and context-length constraints are fixed, and verifiability criteria are introduced: citation correctness and the coverage of answer statements by sources serve as practical proxies for evidentiality

(Pradeep et al., 2025; Ayala & Bechard, 2024). Finally, to align the comparison with fine-tuning, an operational perspective is adopted: fine-tuning is interpreted as a mechanism for stabilizing format, terminology, and answer discipline under inevitable parametric knowledge obsolescence; privacy risks and attacks on the retrieval layer are explicitly incorporated to ensure that the comparison remains valid for real-world domain deployments rather than only for laboratory metrics (Hu et al., 2021; Zeng et al., 2024).

Results and Discussion

The baseline architecture of systems in which generation is augmented by retrieval is constructed as a pipeline whose stages are tightly interdependent: first, documents are cleaned, normalized, and indexed; then, on a query, a set of candidates is retrieved; after that, more precise reranking is performed; a size-limited context is assembled; and only then is an answer produced, which additionally undergoes checks for source grounding and citation correctness. This scheme is described as a canonical pattern for applied datasets and evaluation tracks in source-grounded generative answering, where the inputs and outputs of the retrieval and reranking modules are explicitly specified (Pradeep et al., 2025).

A critical engineering detail of such a pipeline is document segmentation prior to indexing: excessively large segments dilute the relevance signal and suppress useful details, whereas excessively small segments disrupt semantic coherence and impair recall. Comparative studies of segmentation show that methods that aim to preserve the global semantic contour improve the coherence of the retrieved context at the expense of increased computational cost (Merola & Singh, 2025). Metadata (date, version, source, access level) functions not as cosmetic additions but as a control mechanism: filtering by time and version prevents the mixing of document revisions and helps align freshness with domain requirements, which is directly connected to the introductory thesis about the high cost of error and the need for reproducible source verification.

Error sources in such systems are conveniently grouped by their locus, because the symptom hallucination is often a consequence rather than a cause. At the retrieval level, quality degradation most often arises when the needed fragment does not appear in the top ranks due to lexical mismatch, query ambiguity, or incorrect segmentation, forcing the generator to fill gaps with conjecture. At the reranking level, biases emerge when the model prefers stylistically similar but substantively secondary fragments, and during context assembly, additional noise is introduced by duplicates, conflicting statements, and the gluing together of misaligned sources, especially hazardous in the presence of multiple versions of the same document.

Finally, at the generation level, typical errors include misaligned citations and spurious confidence, when an answer appears formally well-justified but cannot in fact be derived from the provided fragments. Studies devoted to reducing hallucinations via retrieval show that the quality of contextual grounding depends not only on the presence of retrieved fragments but also on how the model is compelled to use them when constructing structured outputs and citations (Ayala & Bechard, 2024). Verification at the end of the pipeline should therefore be treated as a separate quality-control module: it does not repair poor retrieval but enables detection of discrepancies between context and conclusions, thereby localizing the cause of degradation, an important property for subsequent comparison with fine-tuning, where error is often masked by stylistic change rather than by improved evidentiality. The Retrieval-Augmented Generation Pipeline Stages are shown in Figure 1.

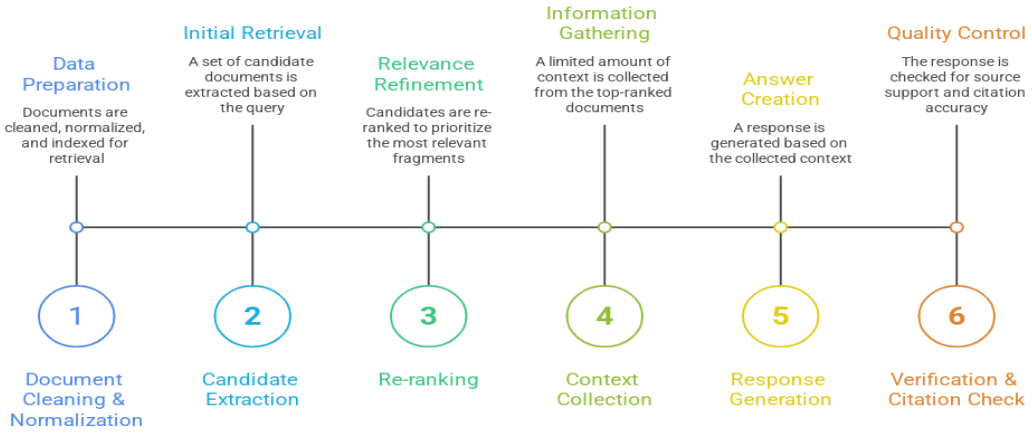

Fig. 1. Retrieval-Augmented Generation Pipeline Stages

Sparse retrieval, whose most prominent representative is based on the BM25 formula, relies on literal matches and term statistics; consequently, in domain collections, it often performs particularly well when queries contain codes, legal formulations, or stable proper names. Its weakness manifests under paraphrasing and synonymy, when meaning remains close but the surface form diverges. It is noteworthy that even within this classical family, there remains room for learning: it has been shown that sparse query representations can be automatically expanded and reweighted so as to improve quality while preserving speed, meaning that the boundary between hand-crafted rules and learning is less rigid here than it may seem (Chen & Wiseman, 2023). Alternatively, dense retrieval maps the query and document fragments to the same embedding space and computes their similarity, particularly when a domain has terminological variants or requires indirect information. Dense retrieval methods for this task suffer from domain shifts, and the robustness of existing methods has been measured mainly through surveys and systematizations of dense retrieval based on pretrained language models (Zhao et al., 2024).

Hybrid retrieval can therefore naturally combine the two, without forcing the relevance signal into one or the other: lexical precision is preserved for hard tokens, for example, while vector similarity ranks semantically-relevant candidates that would not have been ranked highly otherwise. This means that the question is not whether to mix, but how to mix. The score distributions resulting from fusion functions (e.g., linear score combinations, reciprocal-rank fusion (Bruch et al., 2024)) differ from the query/document distributions being indexed and queried, justifying separate treatment. In hybrid schemes, the role of reranking becomes especially pronounced: even a strong initial selection often leaves a candidate pool that is too broad and noisy, and an additional model jointly evaluating query–fragment pairs then shifts the system toward higher top-k precision, which is crucial for subsequent generation that would otherwise lean on arbitrary context snippets (Pradeep et al., 2022).

Query expansion and multi-variant retrieval add another degree of freedom to the pipeline: instead of a single query, several semantic projections of the user's intent are constructed, after which results are aggregated, increasing recall and reducing dependence on a single formulation. In recent years, this step has been increasingly delegated to large language models, which generate clarifications or pseudo-documents acting as a guiding bridge between the query and the collection; empirical work shows that such generation can improve retrieval in both lexical and vector systems, although it also creates a risk of introducing extraneous details, which necessitates careful verification (Jagerman et al., 2023). Reranking, in turn, can be implemented not only by specialized pairwise models but also by the large language model itself operating in relevance-judgment mode; however, this sharpens the latency–quality trade-off, which in domain-specific search almost always has to be resolved by limiting the number of candidates and enforcing strict context economy (Pradeep et al., 2022).

Iterative retrieval schemes, where the system issues several reformulated queries, retrieves the new entities, and logs the search path, are particularly useful for multi-

piece queries with relevant information stored in different documents. Modern versions reduce the need to preserve context and invoke the LLM when generating intermediate queries and filtering retrieved pieces of evidence in favor of more resource-efficient approaches (Jiang et al., 2024). In contrast to structural blindness in segmentation, knowledge-graph-based systems aim to identify relationships between fragments by linking fragments and their statements. They therefore enable grouping passages that share information and are logically related fragments, supporting both coherence and the diversity of context. Within this line of work, schemes have been proposed in which the graph defines paths between fragments, thereby improving the consistency of the material retrieved for subsequent generation (Zhu et al., 2025). A Compact Overview of Retrieval Strategies for RAG is shown in Table 1.

Table 1. Compact Overview of Retrieval Strategies for RAG

Approach	Best for	Main pitfall
Sparse (BM25)	codes, legal phrases, exact names	weak on paraphrases/synonyms
Dense (vectors)	semantic match, indirect queries	domain drift, data sensitivity
Hybrid (sparse + dense)	balanced precision/recall	fusion needs tuning
Reranking (cross-encoder/LLM)	top-k precision for RAG	latency, needs small candidate set
Multi-query expansion (often via LLM)	higher recall, wording robustness	off-topic/hallucinated queries
Iterative retrieval	multi-hop questions	extra steps + context control
Knowledge-graph aided	coherent, linked evidence	graph build/maintenance cost

In contrast with retrieval-augmented generation, fine-tuning allows the system to change how it generates text. Fine-tuning is useful when a domain system needs to produce output in a specific format, consistently use particular definitions, conform to a specific style, provide structured justification for its decisions, or avoid generating superfluous details. At the same time, it is a poor substitute for access to current information, because knowledge encoded in parameters inevitably becomes outdated, and attempts to fill gaps more often lead to conjecture than to verifiable citations.

Supervised learning on labeled examples that specify the desired reaction to a query is typically used as a disciplinary tool: the model learns not only to answer correctly but also to answer in the prescribed manner. This manifest itself in robust instruction following, the ability to preserve domain terminology without redundant explanations, a fixed template for conclusions and warnings, and controlled refusal behavior when data are insufficient. A downside is that the method's effectiveness depends on the abundance and diversity of the current examples; it can also lead to limited flexibility in the model, with predictions even when the input contains no information about the answer.

Preference-based optimization shifts the goal from correct versus incorrect to a ranking of behavioral

variants. It learns to prefer answers that are more useful, careful, and cautious, and reject those that are not verifiably true but rhetorically convincing. This leads to more precise levels of confidence, completeness, and self-critique. It also allows the model to learn to avoid speculation, which is penalized, and to increase its verifiability, which is encouraged. The weakness here is conceptual: preference is not truth, and under poorly designed criteria, the model can learn to appear reliable without becoming more reliable in substance. It can increase plausibility rather than strengthen factual grounding.

Parameter-efficient fine-tuning methods introduce small sets of additional parameters without rewriting the entire weight matrix, thereby making domain adaptation inexpensive, fast, and manageable. This is convenient in organizations that must maintain several domain-specific behavioral variants and rapidly roll back between versions. This results in a large collection of adapters atop a single model, each with its own data format, vocabulary, and answer style. If the domain adaptation requires such diverse representations, a single lightweight adaptation may not be optimal, and one must again trade-off between ease of deployment and depth of domain adaptation.

Continuing pretraining on a large corpus of domain text does not significantly affect the formatting of returned answers. Its effect can be described as favoring linguistic substrates that underlie understanding. Such substrates include the ability to recognize unusual word types, regular phrases, common patterns of articulation (e.g., headings and sections), and latent connections within professional domains (e.g., a system of definitions and caveats). This form of adaptation can improve retrieval or generation performance by modifying the perception of the domain language. However, at a much higher cost, less controllable, and with the risk of over-specialization through domain jargon, it is always a possibility unless domain-specific data is sufficiently balanced with general data, and degradation on external tasks is controlled.

The comparison between retrieval-augmented generation and fine-tuning is conveniently framed in terms of operational and scientific verifiability criteria. Retrieval-augmented generation has advantages in knowledge freshness and controllability. Newly added documents can be used as soon as their collection and related indexes are updated, and the ability to show the source and localize errors in the retrieval and ranking parts is preserved. Fine-tuning has an advantage in behavioral stability. It improves consistency and can program formatting instructions for a specific domain. However, it is less amendable to rapid factual updates, is vulnerable to memory shadow, in which the model recalls sensitive snippets of the training data, and models may also become overconfident in outdated formulations. These limitations can result in divergent control regimes during deployment, necessitating monitoring of retrieval and segmentation quality, and the correctness of metadata in retrieval-augmented generation, and using careful dataset versioning, behavioral drift tests, and reproducible training pipelines during fine-tuning. Comparison of RAG vs. Fine-Tuning by criteria is presented in Table 2.

Table 2. Comparison of RAG vs. Fine-Tuning by criteria

Criterion	Retrieval-Augmented Generation (RAG)	Model Fine-Tuning
Knowledge freshness & updatability	High; updates via documents and indexes	Low; updates require new training
Source control & reduced hallucination	Strong when citations and context-grounding checks are enforced	Can improve caution/style, but doesn't guarantee source grounding without external context
Cost & latency	Higher per-query cost due to retrieval/ranking; latency depends on pipeline steps	Cost shifted to training; inference can be cheaper and faster

Risks	Data leakage via retrieval/access control; noisy or wrong passages; segmentation errors	Overfitting; knowledge staleness; behavior drift; possible retention of sensitive data
Operations	Needs index/metadata monitoring and retrieval/citation quality control	Needs disciplined training, strict data versioning, and behavior regression tests
Reproducibility	Usually higher: searches can be repeated and sources shown	Harder: depends on training runs, data, and optimization settings

In applied domain-specific search systems, the most robust regime is one in which retrieval and generation form a truth scaffold, and fine-tuning acts as a behavioral regulator. The logic is straightforward: first, the system must find relevant material and make it accessible to the model; then the model must handle that material in a disciplined manner, without source substitution and without stylistic improvisation. A typical pattern looks as follows: retrieval functions as the primary mechanism for freshness and verifiability, while lightweight domain adaptation locks in the answer format, terminology, and citation rules, turning the same pipeline into a predictable interface for diverse user groups.

A separate class of combined solutions targets not the generator but the retrieval and reranking modules themselves, because this is where the root causes of many downstream errors often arise. Fine-tuning the retriever is usually designed to pull the representations of queries and relevant fragments closer together while pushing non-relevant ones apart; fine-tuning the reranker improves the model's sensitivity to subtle domain signals, such as prioritizing normative formulations over explanatory text. This shifts the geometry of candidates: the generator receives a cleaner context and is less compelled to compensate for gaps. An important production detail is that retriever and reranker fine-tuning should rely on real query logs and access rules; otherwise, the system begins to optimize for laboratory query–document pairs and loses robustness on naturally occurring formulations.

Query routing becomes necessary when the task spectrum is broad, ranging from quick navigational queries to composite questions that require multi-step search and fragment comparison. In the simplest regime, one-step retrieval and answering on a short context suffice, ensuring minimal latency and cost. In the more complex regime, the system triggers a sequence of clarifications, repeated retrievals, and contradiction checks, but this regime should be reserved for cases with clear signs of complexity; otherwise, the pipeline degenerates into an expensive mechanism that overthinks when a simple definition would suffice. In production, it is therefore useful not to complicate everything at once, but to separate flows: a fast path for brief queries and an extended path for queries that exhibit compositionality, ambiguity, or explicit comparison requirements.

Safeguards in such systems should not merely decorate the answer but compel the model to be epistemically honest. Citation here serves as an interface between retrieval and generation: if a citation is not anchored to a specific fragment, it does not protect against fabrication but only simulates evidential support. Equally important is a strategy for correct non-knowledge: the model must explicitly state that the retrieved context does not warrant a conclusion and either propose query refinement or enumerate missing data. In practice, this is implemented via coverage checks: the answer is decomposed into statements, and each statement must have a supporting fragment; if no support is found, the system should downgrade confidence or refrain from formulating the statement rather than completing it.

Quality evaluation in domain-specific search should start with separate measurement of retrieval and generation; otherwise, it is impossible to determine where in the system quality is being lost. For retrieval, the crucial metrics are those that reflect the probability of correct fragments appearing in the top ranks, the quality of shallow-depth ranking, and coverage metrics indicating how often the correct source is retrieved. For generations, important metrics have concerned contextual grounding, citation accuracy, and user utility, where utility should be interpreted not as a subjective impression but as the ability of an answer to resolve the

task without additional clarification rounds or hidden assumptions. It is important to remember that high textual smoothness is not a quality metric: in domain systems, it can sometimes mask errors.

Experimental design typically combines automated evaluation on a fixed set of frozen queries with human assessment and live-traffic testing, since each procedure detects different classes of defects. Automated evaluation enables rapid comparison of retriever, reranker, and context-assembly variants, but it is less effective at detecting user-interaction failures and cases where the correct answer depends on pragmatic aspects of the query. Human assessment reveals failures in explainability and citation correctness, but is expensive and slow, and is therefore best used for calibration and for control in critical scenarios. Live traffic testing can provide better predictions of latency and system utility, but it generally requires more careful risk management to avoid issues when processing private or critical data.

In a 2-dimensional projection, the axes would be the stability of domain knowledge, the necessity of citation-based evidentiality, the availability of training data, and the acceptable cost of the system. If domain knowledge is unstable and evidentiary requirements must be met with citations, a retrieval pipeline with strong guardrails and fine-tuning for formatting and behavior is a reasonable choice. If the domain is stable and queries are fairly standard and well labeled, fine-tuning can work (specially to reduce the number of dialogue steps and produce more consistent output). Obsolescence is more likely in this case, but it is also easier to alleviate through retrieval. The baseline system that is described generally consists of an initial hybrid retrieval, a lightweight reranking model, and good segmentation, metadata, and citation heuristics. It then continues to query routing, multi-step search for edge cases, and selective fine-tuning of individual components once sufficient data has amassed, and the quality bottleneck has been identified.

Conclusion

Domain-specific search imposes on generative systems requirements that in general-purpose search may be viewed as nice improvements, but here become conditions of viability: high precision among the top results, reproducible source-based verification, careful temporal and version-aware freshness, and secure, regulation-compliant handling of sensitive data. Within this frame, the key thesis of the article takes on the character of a methodological imperative: the value of an answer lies not in its fluency but in its decomposability into supporting fragments, and even Retrieval-Augmented Generation remains vulnerable, from segmentation errors and lexical mismatch to leakage via the retrieval corpus, if metadata filtering, access control, and final grounding checks are not embedded into the pipeline. In this perspective, hallucination is interpreted not as a mysterious weakness of the generator but as a symptom of a breakdown in the chain of retrieval, reranking, context assembly, and citation attachment, where each node introduces its own type of bias and, crucially, admits localization of the degradation's root cause.

The comparison of retrieval strategies establishes that domain relevance rarely collapses into a single signal: sparse methods perform strongly on codes, legal formulations, and exact names but fail under paraphrase; dense vector schemes elevate indirect matches and better tolerate terminological variability but are sensitive to domain drift and data quality; hybrid methods promise balance but require meaningful rank fusion and benefit in particular from reranking, which pushes the system toward high shallow-depth precision, a critical property for subsequent generation. Against this background, query expansion, multi-variant and iterative retrieval, and graph-based enhancements appear not as fashionable embellishments but as attempts to stabilize the completeness and coherence of evidence for complex, composite queries; their cost lies in latency, context control, and the need for verification, because additional generation of intermediate queries can just as easily become a bridge to relevant material as a channel for injecting extraneous details.

Fine-tuning, by contrast, changes not access to knowledge but the dynamics of model behavior: the discipline of instruction following, the stability of terminology, the formatting of justifications, confidence management, and the capacity to not know correctly. Supervised training on examples consolidates how one ought to answer; preference optimization calibrates degrees of usefulness and caution; parameter-efficient adaptations turn domain specialization into a manageable library of versions; and continued pretraining deepens command of the domain language, yet none of these trajectories eliminates the fundamental problem of parametric knowledge obsolescence or the risk that rhetorical neatness will mask substantive unreliability. The operational conclusion of the article is therefore

formulated as a division of labor: retrieval and ranking constitute the truth scaffold and the mechanism of knowledge refresh, while fine-tuning acts as a behavioral regulator; together with query routing, strict citation rules, coverage checks for answer statements, and separate evaluation of pipeline modules, such a hybrid approach yields not merely answers but a controllable, auditable, and reproducible domain-specific search system.

References

1. Ayala, O., & Bechard, P. (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*, *6*, 228–238. https://doi.org/10.18653/v1/2024.naacl-industry.19
2. Bruch, S., Gai, S., & Ingber, A. (2024). An Analysis of Fusion Functions for Hybrid Retrieval. *ACM Transactions on Information Systems*, *42*(1), 1–35. https://doi.org/10.1145/3596512
3. Chen, X., & Wiseman, S. (2023). BM25 Query Augmentation Learned End-to-End. *ArXiv*. https://doi.org/10.48550/arxiv.2305.14087
4. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., & Wang, H. (2023, December 18). *Retrieval-Augmented Generation for Large Language Models: A Survey*. ArXiv. https://doi.org/10.48550/arXiv.2312.10997
5. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., & Chen, W. (2021). LoRA: Low-rank adaptation of large language models. *ArXiv*. https://doi.org/10.48550/arxiv.2106.09685
6. Jagerman, R., Zhuang, H., Qin, Z., Wang, X., & Bendersky, M. (2023, May 5). *Query Expansion by Prompting Large Language Models*. ArXiv. https://doi.org/10.48550/arXiv.2305.03653
7. Jiang, Z., Sun, M., Liang, L., & Zhang, Z. (2024). Retrieve, Summarize, Plan: Advancing Multi-hop Question Answering with an Iterative Approach. *ArXiv*. https://doi.org/10.48550/arxiv.2407.13101
8. Merola, C., & Singh, J. (2025). Reconstructing Context: Evaluating Advanced Chunking Strategies for Retrieval-Augmented Generation. *ArXiv*. https://doi.org/10.48550/arXiv.2504.19754
9. Pradeep, R., Liu, Y., Zhang, X., Li, Y., Yates, A., & Lin, J. (2022). Squeezing Water from a Stone: A Bag of Tricks for Further Improving Cross-Encoder Effectiveness for Reranking. *Lecture Notes in Computer Science*, *13185*, 655–670. https://doi.org/10.1007/978-3-030-99736-6_44
10. Pradeep, R., Thakur, N., Sharifymoghaddam, S., Zhang, E., Nguyen, R., Campos, D., Craswell, N., & Lin, J. (2025). Ragnarök: A Reusable RAG Framework and Baselines for TREC 2024 Retrieval-Augmented Generation Track. *Advances in Information Retrieval: 47th European Conference on Information Retrieval*, 132–148. https://doi.org/10.1007/978-3-031-88708-6_9
11. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A. K., & Gurevych, I. (2021). BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. *ArXiv*. https://doi.org/10.48550/arxiv.2104.08663
12. Zeng, S., Zhang, J., He, P., Liu, Y., Xing, Y., Xu, H., Ren, J., Chang, Y., Wang, S., Yin, D., & Tang, J. (2024). The Good and The Bad: Exploring Privacy Issues in Retrieval-Augmented Generation (RAG). *Findings of the Association for Computational Linguistics: ACL 2022*, 4505–4524. https://doi.org/10.18653/v1/2024.findings-acl.267
13. Zhao, W. X., Liu, J., Ren, R., & Wen, J.-R. (2024). Dense Text Retrieval Based on Pretrained Language Models: A Survey. *ACM Transactions on Information Systems*, *42*(4), 1–60. https://doi.org/10.1145/3637870
14. Zhu, X., Xie, Y., Liu, Y., Li, Y., & Hu, W. (2025). Knowledge Graph-Guided Retrieval Augmented Generation. *ArXiv*. https://doi.org/10.48550/arxiv.2502.06864