Universal Library of Innovative Research and Studies Research Article

ISSN: 3065-0003 | Volume 3, Issue 1

Open Access | PP: 57-64

DOI: https://doi.org/10.70315/uloap.ulirs.2026.0301008

Building Trustworthy AI for Enterprise Support: An Empirical Study of a RAG-Based Architectural Framework

Udit Joshi, Kapil Verma

Product Manager and Software Engineer, Google, 1600 Amphitheater Parkway, Mountain View, CA 94043.
ORCID iD: 10009-0006-9276-5255, 20009-0007-0413-6596

Abstract

The paper considers an approach to engineering a trustworthy enterprise support service based on a Retrieval-Augmented Generation (RAG) architecture in a high-load ticket-handling environment. The study's relevance stems from the widespread deployment of generative solutions in contact centers and users' growing sensitivity to hallucinations, stale data, and unpredictable system behavior. The objective of the research is to obtain an empirical assessment of a full-scale RAG architecture for internal support and to identify engineering decisions that critically determine its reliability. Across five experimental series, empirical data were obtained for key metrics: MRR@10 for retrieval quality, overall refusal and correct-refusal rates, the proportion of confident yet factually incorrect answers, p50/p90/p99 end-to-end latency, the time required to incorporate a document into the index, and the share of interactions involving outdated information. The scientific contribution of the work lies in analyzing a RAG system as an integrated engineering artifact studied on a real-world corporate corpus. It is shown that semantic document segmentation yields a substantial improvement in retrieval quality over fixed-size chunking. That dense semantic search dramatically outperforms sparse keyword-based search, including in terms of the system's ability to refuse correctly under knowledge scarcity and to reduce dangerous hallucinations. It is established that the generative component is the dominant source of latency. In contrast, a hybrid indexing strategy that combines streaming delta-indexing of critical documents with nightly re-indexing of the entire corpus enables maintaining both knowledge freshness and operational resilience. The paper is intended for researchers and practitioners designing scalable and trustworthy enterprise support systems built on large language models.

Keywords: *Enterprise Support, Retrieval-Augmented Generation, Semantic Search, Trustworthy AI.*

INTRODUCTION

As service operations are progressively digitized, text and voice assistants, agent-assist solutions, or auto-reply assistants are mainstreamed into customer contact centers and internal service functions. Evidence suggests that these solutions markedly change customer expectations and behavior, as well as user error sensitivity, when interacting with automated service systems. This is especially true in high-stakes service contexts, such as medical and legal services [1]. Large language models are known to suffer from so-called hallucinations, in which the system confidently produces factually incorrect statements, a phenomenon well documented in the general natural language processing literature and in applied domains, including medicine and other safety-critical areas [2]. In B2B and B2C support settings, such errors lead to direct trust erosion, additional operational costs (repeat tickets, escalations, manual verification), and long-term reputational risks. Consequently, controllability and predictability of system behavior become central prerequisites for industrial deployment.

One key direction for mitigating hallucinations and improving the reliability of generative systems is the use of retrieval-augmented architectures, which leverage an external knowledge store rather than relying solely on its parametric memory. In such approaches, the generative component is responsible for linguistic realization and abstraction. At the same time, a specialized search module retrieves relevant facts, policies, histories of past tickets, and other contextual elements from the corporate corpus, which are then explicitly passed into the model prompt [3]. This grounding mechanism is fundamental in corporate scenarios, where most relevant knowledge resides in internal repositories, evolves continuously, and is subject to strict access control, regulatory compliance, and traceability constraints that are

Citation: Udit Joshi, Kapil Verma, "Building Trustworthy AI for Enterprise Support: An Empirical Study of a RAG-Based Architectural Framework", Universal Library of Innovative Research and Studies, 2026; 3(1): 57-64. DOI: https://doi.org/10.70315/uloap.ulirs.2026.0301008.

markedly more stringent than in mass consumer applications. In parallel, related approaches are developing that rely on structured knowledge graphs, which likewise show promise in reducing hallucinations by explicitly linking answers to formalized facts [4].

Despite the rapid progress of retrieval-augmented architectures, most existing work focuses either on conceptual descriptions of such systems or on isolated components, while leaving underexplored the impact of concrete engineering decisions on system behavior in production. Contemporary surveys emphasize that evaluating such hybrid pipelines is complicated by the need to jointly measure search quality, generation faithfulness, robustness to data staleness, and a spectrum of operational characteristics, including end-to-end response latency [3]. At the same time, there is little empirical evidence on how exactly the document chunking strategy, the choice of retrieval component (dense semantic search versus classical sparse keyword-based indexing), and the indexing and freshness-maintenance policy affect retrieval metrics, the model's ability to refuse responses when information is missing, and the temporal characteristics of the whole pipeline [5].

To assess the practical value of retrieval-augmented architectures, they need to be considered not in isolation but in comparison with established enterprise support practices. These include scenario-driven dialog systems based on hard-coded rules and decision trees, traditional keyword search engines, and purely generative assistants without access to internal organizational data. Research on trust in conversational agents shows that simple rule-based chatbots or those layered on top of classical search often suffer from a gap between expected and actual behavior. In contrast, generative models without a corporate context tend to either hallucinate or produce overly generic responses [1]. Against this backdrop, retrieval-augmented architectures are viewed as candidates for a new standard. However, systematic comparison of their characteristics with these baseline approaches under high load and in complex internal corpora remains limited.

The present study aims to construct and empirically evaluate a retrieval-augmented support architecture for a large enterprise environment with a high volume of tickets. Several interrelated questions are addressed. First, how does the document chunking strategy influence the quality of semantic search, as measured by ranking metrics, and can more fine-grained segmentation increase the likelihood that a relevant fragment appears among the top results. Second, how critical is the choice of retrieval component type, specifically, how do systems based on sparse keyword indices compare with systems using dense representations for semantic matching between queries and fragments. Third, in what way does retrieval quality affect the model's ability to refuse to answer when relevant information is absent from the corpus, which is particularly important for reducing dangerous hallucinations. Fourth, which elements of the pipeline determine end-to-end response latency in an interactive dialog setting, and where the primary bottlenecks actually lie. Finally, fifth, how should corpus updates and re-indexing be organized so as to maintain freshness without severely degrading performance or introducing temporal inconsistencies between document versions.

The paper contributes to the discussion of these questions in several ways. A concrete retrieval-augmented support architecture is described, tailored for internal ticket management, and treated as a full-fledged object of empirical investigation rather than merely as a conceptual schema. Based on a series of ablation-style experiments, the influence of different corpus segmentation strategies, retrieval component types, and refusal-handling configurations on retrieval quality and final answers is quantified using metrics recommended in recent surveys on RAG-system evaluation. Additionally, the latency structure across pipeline stages is analyzed, and a hybrid approach to indexing and corpus updates is proposed, combining streaming processing of critical changes with batch re-indexing. Finally, by comparing the proposed architecture with existing enterprise support practices, the study formulates a set of practical recommendations for designing reliable and scalable systems that narrow the trust gap between users and intelligent agents within an organization.

MATERIALS AND METHODOLOGY

The system under consideration, deployed within an internal enterprise support perimeter, serves several interrelated classes of tasks: issues related to the information infrastructure, product incidents and requests, and typical HR cases grounded in internal policies, benefits, and regulations. On this corpus, three modes of exploitation are formed: asynchronous matching of new tickets to previously resolved cases to accelerate and standardize support; online hints for human operators based on the latest messages or ticket descriptions in the form of relevant knowledge-base articles and response templates; as well as automation of labeling, prioritization, and routing of tickets, where the operator acts as a verifier.

The input corpus aggregates knowledge-base articles and operational documentation, historical tickets and incidents, transcripts of support-channel dialogs, and internal documents (policies, instructions, technical manuals). These undergo normalization, removal of system-level formatting, deduplication, filtering of personal and sensitive data using regular expressions and entity recognition, and subsequent segmentation into semantic fragments that serve as the minimal units for search and generation.

The architecture rests on two interrelated pipelines, shown in Figure 1: an asynchronous corpus-preparation pipeline and a real-time query-handling pipeline.

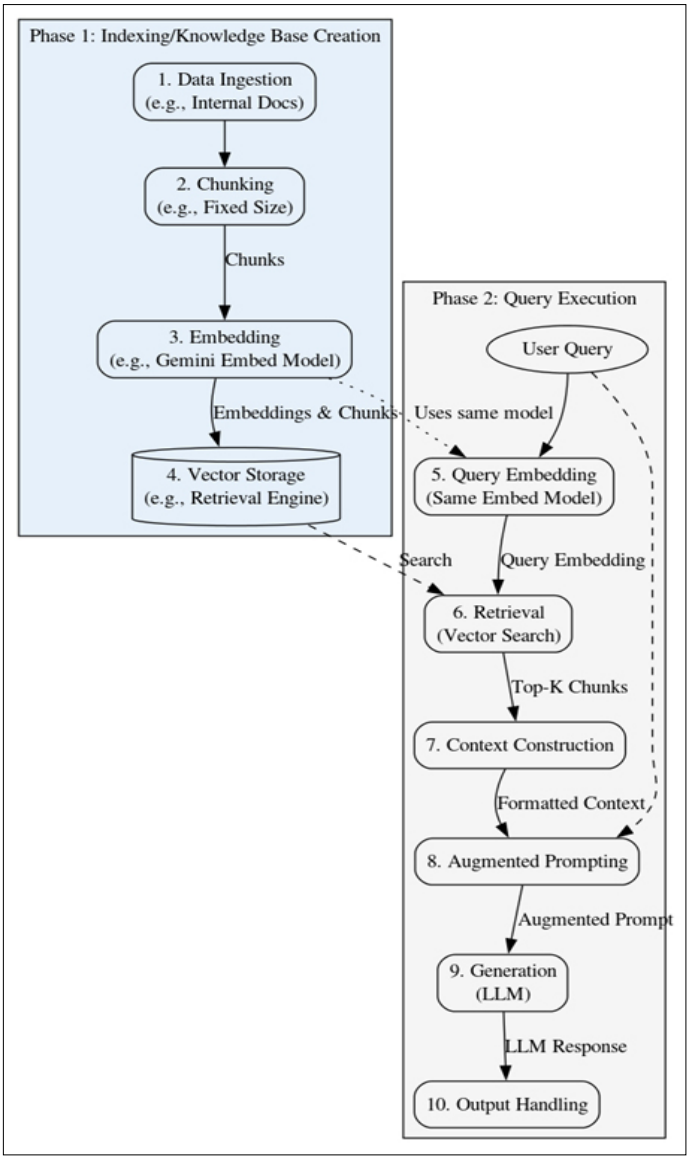

Figure 1. Core Flow Diagram: The RAG-Powered Decision Support Architecture

In the first pipeline, documents are ingested from various storage systems via connectors, split into fragments using fixed-size windows with overlap and by semantic structure (headings, paragraphs, logical sections), and a vector representation is computed for each fragment using a state-of-the-art text-embedding model within a hybrid-memory paradigm. The tuple vector, text, and metadata (identifier, source, timestamp, version) are then stored in a vector database that supports nearest-neighbor search by cosine similarity. In the second pipeline, the incoming query is normalized, encoded by the same model, and employed to retrieve the closest fragments under configurable recall-latency trade-offs. These fragments are used to assemble a context with explicit source indications; an augmented prompt is then constructed for a general-purpose large language model with strict constraints: reliance exclusively on the provided context, mandatory refusal when insufficient data is provided, and a requirement to include references to the utilized fragments. The answer is returned to the operator together with a recommended next-step action and a source panel enriched with usefulness assessments, stale-information flags, and operator edits, which serve as feedback signals for subsequent re-evaluation and fine-tuning of system components.

The experimental part of the study is organized around a series of controlled modifications to individual pipeline components. In the first group of experiments, the impact of document chunking strategy on semantic search quality is investigated in a scenario of matching queries to knowledge-base articles: for a fixed set of queries with expert labels of relevant articles, different segmentation variants are compared, and retrieval quality is measured by the mean reciprocal rank of the first relevant fragment among the top-10 results (MRR@10), in line with standard practice in semantic search evaluation for open-domain question answering.

In the second group of experiments, two classes of retrieval modules are compared: sparse keyword-based search using classical ranking and semantic search over dense representations, applied to two distinct corpora, a mixed corpus and one consisting primarily of knowledge-base articles. Both module classes use the same queries and sample size, enabling direct comparison of metric values.

In the third experimental group, the system's ability to refuse correctly when no relevant information is present is evaluated. A test set is constructed in which some portion of queries is, by design, unsolvable using the available corpus. The system, configured to refuse when context is insufficient, is tested under two retrieval configurations. The analysis considers the overall refusal rate, the share of correct refusals among unanswerable queries, and, additionally, the proportion of confident but factually incorrect answers, consistent with contemporary approaches to evaluating selective refusal in generative systems.

Finally, to ensure the applicability of the results to industrial operation, temporal characteristics and system behavior under corpus updates are additionally measured. In a dedicated series of runs, latencies at key pipeline stages are recorded: query-embedding computation, vector-index search, answer generation, and full end-to-end time from query arrival to response, along with the median, ninetieth, and ninety-ninth percentiles, which are then compared with target service-level agreements. In parallel, the effectiveness of a hybrid strategy for maintaining corpus freshness is examined: critically important documents are indexed in near real time via an event stream, whereas the entire repository is reindexed nightly in batch windows, using content-hash versioning for atomic replacement of old versions. For these experiments, the time between document publication and its appearance in search results is measured, as well as the share of interactions in which the system cites outdated information, reconstructed from query logs and operator feedback signals.

RESULTS AND DISCUSSION

Results on Experiment 1

The results of the first experimental series are presented in Table 1. They demonstrate that the choice of document chunking strategy has a substantial impact on retrieval quality in the scenario of matching user queries to knowledge-base articles.

Table 1. Impact of Chunking Strategy on Retrieval Efficacy

Chunking Strategy	Parameters	MRR@10	Analysis
Fixed-Size	512 tokens, 0 overlap	0.72	Poor. Often failed when the answer spanned a chunk boundary.
Fixed-Size w/ Overlap	512 tokens, 100 overlap	0.81	Better. Overlap helped mitigate boundary splits, but it is still arbitrary.
Semantic (Paragraph)	N/A	0.90	Winner. Preserved logical context, ensuring retrieved chunks were self-contained.

With fixed-size segmentation without overlap, the average MRR@10 was 0.72, reflecting frequent cases in which the question and answer ended up in different fragments, with the relevant context falling outside the top positions. Introducing overlap between adjacent fragments improved the situation: MRR@10 rose to 0.81, although some issues with breaking logical units persisted. The highest quality was achieved with semantic segmentation along document structure, where MRR@10 reached 0.90; the relative gain over the naive variant was roughly 25%, aligning with recent studies demonstrating the advantages of semantic and adaptive chunking over strictly token-based schemes in retrieval-augmented support tasks [6]. Manual inspection of cases where fixed-size chunking failed showed that most involved the problem statement and its solution being located in adjacent paragraphs or under different subheadings: length-based splitting separated these parts into various fragments, whereas semantic segmentation kept them within a single block and enabled the retrieval module to rank the relevant fragment among the top results consistently.

Results on Experiment 2

The second group of experiments, devoted to comparing sparse and dense retrieval, confirmed that semantic search over continuous representations establishes the upper bound of quality for the corpora under consideration. The results are summarized in Table 2.

Table 2. Retrieval Efficacy by Strategy (Dense vs. Sparse)

Corpus	Retrieval Strategy	MRR@10	Key Retrieval Insight
Corpus 1 (images, text)	Keyword Search (Sparse)	0.164	Low performance; fails to capture conceptual intent.
	Semantic Search (Dense)	0.90	Excellent; relevant context consistently ranked 1-2.
Corpus 2 (knowledge articles)	Keyword Search (Sparse)	0.20	Low performance; struggles with conversational, high-variability queries.
	Semantic Search (Dense)	0.95	Outstanding; nearly perfect ranking of the critical document.

For the first mixed corpus, which combines textual descriptions, multimedia elements, and free-form user phrasing, sparse keyword search yielded an MRR@10 of about 0.164, suggesting that the first relevant fragment appeared around the sixth position on average and often fell outside the top 10 entirely. Switching to dense search over vector representations increased this figure to 0.90, meaning that relevant context almost always appeared in the first or second position. A similar pattern emerged for the second corpus, based on knowledge-base articles: sparse search achieved an MRR@10 of roughly 0.20, whereas dense retrieval reached 0.95.

These differences are especially pronounced for conversational, incomplete, or abstract queries, where users rarely reproduce the terminology of the documentation, and broadly align with findings that dense representations better capture semantic proximity. In contrast, sparse models tend to fail on paraphrases and rare phrasings [7]. At the same time, for narrow, terminology-heavy queries, sparse search remained competitive, but its contribution to overall quality in the serviced scenarios was limited.

Results on Experiment 3

The third experimental series enabled tracing how retrieval quality influences system behavior in the presence of genuine knowledge gaps. The results are presented in Table 3.

Table 3. Knowledge Gap Detection and Refusal Rates when driven by the two distinct retrieval architectures

Retriever Type	Unanswerable Questions	Total Refusals	Refusal Rate	Exact Match (Correct Refusal)
Sparse (Keyword)	100	185	37%	35%
Dense (Semantic)	100	80	16%	88%

On a test set of five hundred queries, one hundred of which were deliberately unsolvable on the given corpus, the system with sparse retrieval exhibited a high overall refusal rate (37% of all answers), but only 35% of the hundred unanswerable queries resulted in a correct refusal; the remainder were split between erroneous answers and unnecessary refusals on queries that would have been solvable under ideal retrieval.

With dense retrieval, the overall refusal rate dropped to 16%. In comparison, the share of correct refusals on unanswerable queries rose to 88%, and the number of unwarranted refusals on answerable queries decreased markedly. These findings are consistent with recent work on selective refusal and safe behavior in language models, which emphasizes that the ability to distinguish between questions for which context is genuinely insufficient and those for which the model is expected to answer is a distinct capability sensitive to the quality of the retrieved neighborhood [8]. In this case, improved retrieval reduced the uncertainty the model had to decide on, thereby enhancing both refusal accuracy and robustness against dangerous hallucinations.

Results on Experiment 4

Analysis of pipeline timing characteristics showed that, for interactive scenarios, the limiting factor is not retrieval but rather answer generation by the large language model. The results of this experiment are given in Table 4.

Table 4. End-to-End Real-Time Pipeline Latency

Pipeline Stage	p50 (ms)	p90 (ms)	p99 (ms)	Analysis
Query Embedding (GPU)	50	75	110	Highly optimized and stable.
Vector Retrieval (k=5)	120	180	250	Fast. Meets all production requirements.
LLM Generation (w/ context)	400	750	1200	The primary bottleneck. Drives all tail latency.
End-to-End Total	570	1005	1560	

Queries showed embedding latency reaching 50 ms. The 90th percentile reached 75 ms. The 99th percentile reached 110 ms. Vector-index search reached 120 ms for the top 1 nearest fragment. Vector-index search reached 180 ms for the top 5 nearest fragments. Vector-index search reached 250 ms for the top 10 nearest fragments. The median answer latency reached 400 ms. The ninetieth percentile reached 750 ms. The ninety-ninth percentile reached 1200 ms. The median total latency was at 570 ms, and the p90 and p99 latencies were at 1005 ms and 1560 ms respectively. Hence, the p90 latency target of 1s was moderately exceeded by a bit upward.

This latency profile aligns with observations from other evaluations of retrieval-augmented systems, which note that optimization should primarily target the generative component, via model downsizing, quantization, or multi-tier request routing, while the retrieval subsystem can achieve sub-centisecond latencies with a well-designed index and infrastructure [9].

The results related to corpus freshness confirm the effectiveness of the hybrid indexing strategy. For critical documents such as new knowledge-base articles and high-priority incident notifications, the average time from publication in the source system to appearance in search results remained within several minutes due to streaming delta-indexing; for the rest of the corpus, nightly batch re-indexing was used to eliminate accumulated inconsistencies and outdated versions. Content-hash-based versioning ensured atomic replacement of old vector representations with new ones, preventing scenarios in which different parts of the pipeline observe other versions of the same document. According to ticket logs and operator feedback, there was a noticeable reduction in complaints about outdated answers. The frequency of references to clearly stale documents decreased steadily after the introduction of the streaming update layer, in line with broader observations on the role of timely corpus updates in reducing confabulations and strengthening trust in retrieval-augmented systems [10].

Interpretation of Key Results

The interpretation of results begins with the influence of the document segmentation strategy, which emerged as a surprisingly strong factor in this study. The substantial quality gain when moving from naive fixed-length chunking to structurally meaningful segmentation can be explained by the fact that the system no longer disrupts natural semantic units: questions, prerequisites, and answers are more likely to fall within a single fragment rather than being scattered across multiple pieces. In effect, the number of cases in which the retrieval module must infer relevance from fragments lacking a complete thought is reduced. This is consistent

with recent work showing that adaptive segmentation into self-contained passages improves both search quality and the behavior of retrieval-augmented pipelines across question-answering tasks and domains [11]. In practical terms, this implies that corpus preparation should not be treated as a secondary detail but as a high-leverage optimization target: when designing the ingestion pipeline, semantic segmentation aligned with document structure needs to be built in from the outset rather than relying solely on mechanical length-based splitting.

The choice of retrieval module type proved equally illustrative. The experiments demonstrate that dense semantic search over vector representations effectively sets a new quality baseline relative to classical sparse term-based search, particularly in the presence of conversational, incomplete, and metaphorical formulations. Sparse ranking models remain useful where queries and documents share a stable vocabulary. Still, they degrade predictably when users describe problems in their own words and rarely repeat knowledge-base terminology, a pattern already noted in research on conversational search [12]. Dense representations, by contrast, better capture semantic similarity, as corroborated by both the present evaluation and comparative studies showing that such methods systematically outperform classical schemes on open collections and, in some setups, even dominate in quality–cost trade-offs when a carefully engineered index is used [7]. Against this background, the classical search component is best viewed as an auxiliary tool in enterprise architectures, for example, for exact lookup by identifiers or within hybrid schemes that combine sparse and dense retrieval via multi-stage ranking.

In terms of system trustworthiness, it is essential to note that retrieval-augmented architectures address several intertwined issues that are difficult to disentangle in purely parametric models. First, reliance on an up-to-date corporate corpus reduces the frequency with which the model is forced to fill in facts, which correlates with a lower rate of fabricated details, as documented in surveys on confabulations in retrieval-augmented generative systems [10]. Second, explicit inclusion of references to the source fragments and documents renders every statement verifiable: the operator can quickly navigate to the primary source and compare formulations, fostering a habit of critically assessing the proposed answer and perceiving the system as a tool for search and synthesis rather than as an unquestionable authority. Third, the study shows that improved retrieval enhances the model's ability to refuse correctly when the corpus lacks necessary information, thereby reinforcing the principle better an explicit admission of ignorance than a confident error.

From an engineering perspective, the results underscore fundamental trade-offs embedded in such architectures. On the one hand, larger models capable of nuanced interpretation of complex context tend to deliver higher accuracy and better refusal quality. Still, their use inevitably increases latency and resource consumption, critical factors for interactive support scenarios. On the other hand, simplifying the model or aggressively shrinking the context reduces latency but may increase both error rates and unwarranted refusals. Research on optimizing retrieval-augmented pipelines indicates that effective strategies include multi-tier model selection depending on query complexity, caching of embeddings and results, and decomposing responses into rapid draft and slower refinement stages [13]. A similar trade-off arises in maintaining corpus freshness: frequent updates increase answer currency but introduce risks of temporal desynchronization and additional load on the index, whereas purely batch re-indexing simplifies operations but raises the likelihood of serving outdated information. The effectiveness of the hybrid scheme observed here, combining streaming delta-indexing of hot documents with periodic full re-indexing, aligns with recommendations that emphasize matching update frequency to domain-specific requirements for answer fidelity [14].

Limitations and Practical Recommendations

Several limitations should be considered when interpreting the findings. The corpus is drawn from a specific corporate domain dominated by internal regulations, instructions, and service-oriented tickets; thus, direct transfer of the quantitative estimates to other industries and languages requires separate validation. In addition, the study deliberately focuses on technical metrics of retrieval quality, refusal behavior, and latency, while leaving outside the scope direct business indicators such as user satisfaction, average resolution time, or the share of deflected tickets, as well as detailed user studies of how operators and end clients perceive the system.

Overall, our work suggests a few practices to further improve deployment of retrieval-augmented architectures in enterprise support domains: investing in high-fidelity semantic document segmentation and dense retriever performance, careful evaluation of results including ranking metrics, explicit tests for refusal correctness, and latency profiling for each stage, and making sure the interface and user-facing features are tailored to the enterprise context. This includes showing clear source documents, adding easy ways for users to signal errors or obsolescence, and giving clear signals when the model is purposefully refusing an answer. Survey articles on RAG architecture evaluation confirm that it is precisely the combination of such measurable algorithmic properties and carefully designed user interaction that determines whether a system is perceived as technically capable but untrustworthy, or as a genuine partner in day-to-day support operations [15].

CONCLUSION

In light of the objective of constructing a trustworthy retrieval-augmented support architecture, the study shows that the

decisive factors are not abstract properties of the large language model, but rather concrete engineering decisions regarding the corpus, the retrieval module, and the pipeline's operational characteristics. Using a real high-load enterprise environment as the setting, the results demonstrate that semantic segmentation of documents into meaningful fragments and the transition from sparse keyword search to dense semantic retrieval over vector representations do not merely deliver marginal quality gains but radically reshape the system's behavioral profile, increasing the likelihood of locating relevant context and stabilizing ranking. This, in turn, reduces the number of situations in which the generative component must fill in answers and lays the foundation for more predictable, controllable support.

An equally important conclusion is that retrieval quality determines not only answer accuracy but also the system's ability to respond appropriately to data scarcity. The deliberately unanswerable query scenario introduced in the study shows that, under high-quality semantic retrieval, the model significantly less often produces confident yet factually incorrect answers and substantially more often opts for a deliberate refusal when the corpus genuinely lacks the necessary information. Coupled with the practice of explicitly citing the underlying fragments and documents, this turns the retrieval-augmented architecture into a tool that not only answers but also delineates the boundaries of its own knowledge, crucial for enterprise support settings where the cost of error extends beyond a single dialog.

From an engineering standpoint, the work demonstrates that, in interactive scenarios, the pipeline's bottleneck lies not in the retrieval subsystem but in the generative component: it dominates tail latency. It most strongly affects the system's ability to meet stringent service-level agreements. At the same time, the proposed hybrid data-freshness strategy, combining streaming delta-indexing of critical documents with nightly batch re-indexing of the entire corpus and content-level versioning, simultaneously reduces the share of interactions affected by outdated information while keeping operational complexity at an acceptable level. As a result, the architecture exhibits not only high retrieval and answer quality but also operational suitability in the face of a continuously evolving corpus.

Taken together, these results indicate that a retrieval-augmented support architecture can serve as a convincing candidate for a foundational pattern in the design of trustworthy enterprise support systems, provided that its implementation reflects the engineering priorities identified in the study. Despite these caveats, and the fact that only one domain has been examined and that the study is limited to only quantitative measures, it is clear that a careful construction of the data-pipeline, choice of the retrieval method and the integration of refusal and traceability into the engine architecture, rather than enforcing them externally, will be key to building more reliable support-oriented AI in the future.

REFERENCES

1. Adam M, Wessel M, Benlian A. AI-based Chatbots in Customer Service and Their Effects on User Compliance. Electronic Markets. 2021 Mar 17;31:427–45.
2. Rawte V, Chakraborty S, Pathak A, Sarkar A, Islam T, Chadha A, et al. The Troubling Emergence of Hallucination in Large Language Models - An Extensive Definition, Quantification, and Prescriptive Remediations. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. 2023 Dec;2541–73.
3. Klesel M, Wittmann HF. Retrieval-Augmented Generation (RAG). Business & Information Systems Engineering. 2025 Jun 1;67:551–61.
4. Lavrinovics E, Biswas R, Bjerva J, Hose K. Knowledge Graphs, Large Language Models, and hallucinations: An NLP perspective. Journal of Web Semantics. 2024 Dec 1;85:100844.
5. Wang Z, Gao C, Xiao C, Huang Y, Si S, Luo K, et al. Document Segmentation Matters for Retrieval-Augmented Generation. Findings of the Association for Computational Linguistics: ACL 2022. 2025 Jan 1;8063–75.
6. Gomez-Cabello CA, Prabha S, Haider SA, Genovese A, Collaco BG, Wood NG, et al. Comparative Evaluation of Advanced Chunking for Retrieval-Augmented Generation in Large Language Models for Clinical Decision Support. Bioengineering. 2025 Nov 1;12(11):1194.
7. Luan Y, Eisenstein J, Toutanova K, Collins M. Sparse, Dense, and Attentional Representations for Text Retrieval. Transactions of the Association for Computational Linguistics. 2021;9:329–45.
8. Cao L. Learn to Refuse: Making Large Language Models More Controllable and Reliable through Knowledge Scope Limitation and Refusal Mechanism. Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. 2024 Nov;3628–46.
9. Es S, James J, Anke LE, Schockaert S. RAGAs: Automated Evaluation of Retrieval Augmented Generation. Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. 2024 Mar;150–8.
10. Zhang W, Zhang J. Hallucination Mitigation for Retrieval-Augmented Large Language Models: A Review. Mathematics. 2025 Mar 4;13(5):856.
11. Liu Z, Simon CE, Caspani F. Passage Segmentation of Documents for Extractive Question Answering. Lecture notes in computer science. 2025 Apr 1;15574:345–52.

12. Lin SC, Yang JH, Lin J. Contextualized Query Embeddings for Conversational Search. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021 Nov;1004–15.

13. Behera L, Poosapati V. Optimizing Latency And Accuracy Trade-Offs In Large-Scale Retrieval-Augmented Generation Pipelines. International Research Journal of Modernization in Engineering Technology and Science. 2025 Sep 9;7(7):4302–10.

14. Heredia Álvaro JA, Barreda JG. An advanced retrieval-augmented generation system for manufacturing quality control. Advanced Engineering Informatics. 2024 Dec 4;64:103007.

15. Yu H, Gan A, Zhang K, Tong S, Liu Q, Liu Z. Evaluation of Retrieval-Augmented Generation: A Survey. Communications in Computer and Information Science. 2025;2301:102–20.