

Autonomous RAG-based Enterprise Search & Analytics: A Hybrid Retrieval Framework for Grounded Question Answering

Gilgene Foxy Landry Ndong Moundounga¹, Mohammad Sufyan², and Yojna Arora³

^{1,2,3}Department of Computer Science & Engineering, Sharda University, Greater Noida, INDIA

³Corresponding Author: yojana183@gmail.com



www.sjmars.com || Vol. 5 No. 5 (2026): October Issue

Date of Submission: 19-09-2026

Date of Acceptance: 28-09-2026

Date of Publication: 01-10-2026

ABSTRACT

Enterprise organizations increasingly hold their institutional knowledge in scattered, unstructured document repositories that traditional keyword search cannot interpret reliably. This paper presents an Autonomous RAG-based Enterprise Search & Analytics system, a prototype platform that couples hybrid information retrieval with large language model (LLM) generation and a live operational analytics layer. The retrieval stage runs a custom BM25 keyword ranker and a dense vector ranker in parallel, and a configurable fusion layer — Reciprocal Rank Fusion (RRF) or weighted score fusion, switchable at run time — merges the two ranked lists before the top-scoring passages are handed, with citations, to Claude 3.5 Sonnet for grounded answer generation. A companion dashboard exposes real-time key performance indicators so that operators can observe and tune retrieval behaviour without redeploying the system. We report a preliminary pilot evaluation of the prototype over a corpus of ten enterprise policy documents and roughly 30 to 40 test queries: a mean end-to-end latency of 2.0 seconds against a 3-second target, top-1 retrieval scores in the 75 to 95% range, approximately 70% of queries classified as high-confidence, and zero crashes across the test session. The pilot is explicitly small in scale and its numbers are treated as indicative rather than generalizable; the paper is correspondingly transparent about this limitation and about the prototype's current use of placeholder embeddings and an in-memory vector store, both slated for replacement in future iterations.

Keywords- retrieval-augmented generation, enterprise search, hybrid retrieval, BM25, reciprocal rank fusion, dense vector search, autonomous agents, business analytics, large language models, grounded generation.

I. INTRODUCTION

Most of what a modern organization “knows” is not stored in a database; it is scattered across policy PDFs, wikis, ticket histories, spreadsheets, and internal memos. Knowledge workers can spend a significant share of their working time simply looking for the right document, and even when a keyword search tool returns results, it frequently fails to understand the intent behind a query, returning matches that mention the right words but miss the right meaning. Retrieval-Augmented Generation (RAG) has emerged over the past two years as a practical remedy: instead of asking a large language model to answer purely from its parametric memory — a habit that can produce confident but factually unsupported answers — RAG retrieves relevant passages from an external, authoritative corpus at query time and asks the model to answer while grounded in that evidence [2].

Enterprise adoption of this pattern is now widespread, but most deployments still resemble either a plain chatbot bolted onto a single vector index, or a keyword-search box with no generative layer at all. Few small-scale systems combine three things at once: hybrid retrieval that plays sparse, keyword-based matching against dense, semantic matching rather than relying on one alone; a fusion strategy that an operator can inspect and tune rather than a black box; and an analytics layer that turns every query into an observable operational signal rather than a one-off chat transcript. This paper describes the Autonomous RAG-based Enterprise Search & Analytics system, a prototype built to explore exactly this combination.

The word “autonomous” is used here in a specific and deliberately modest sense: the pipeline resolves every query end to end — retrieving, fusing, generating, and scoring its own confidence — without a human choosing a retrieval strategy per query. It is not, at this stage, a multi-agent system capable of independent planning or tool selection in the sense of more elaborate, reflective RAG pipelines; Section IV returns to this distinction explicitly.

II. METHODOLOGY

The system processes each query through five coordinated stages: document ingestion and indexing, dual-path retrieval, configurable fusion, grounded generation, and analytics capture.

2.1 Document Ingestion and Indexing

The prototype corpus consists of ten enterprise policy documents, ingested and indexed twice in parallel: once into a BM25 term index and once into a vector index built from 128-dimensional placeholder embeddings. These placeholder embeddings allow the retrieval, fusion, and analytics layers to be built and exercised end to end before a production embedding provider is integrated. Document upload is designed to be extensible beyond the current static set, though this extensibility has not yet been evaluated under load.

2.2 Hybrid Retrieval Layer

Every incoming query is dispatched to both retrievers simultaneously. The BM25 component is a custom implementation of the probabilistic relevance scoring described by Robertson and Zaragoza [4], returning a ranked list of candidate passages by lexical relevance. The vector component performs a nearest-neighbour search over the placeholder embedding index and returns a separately ranked list by semantic proximity.

2.3 Configurable Fusion Layer

The two ranked lists are merged by one of two interchangeable strategies, selectable at run time from the analytics dashboard without redeploying the service. Reciprocal Rank Fusion, following Cormack et al. [5], scores each candidate passage d as $\text{RRF_score}(d) = \sum_r 1 / (k + \text{rank}_r(d))$, summed over the two retriever rankings, with the constant k fixed at 60 as in the original formulation. Weighted fusion instead linearly combines each retriever's normalized relevance score with a configurable weight, useful when an operator has reason to trust one retriever more than the other for a given corpus.

2.4 Grounded Generation and Confidence Scoring

The fused, top-ranked passages are assembled into a grounded prompt and passed to Claude 3.5 Sonnet, which is instructed to answer strictly from the supplied evidence and to attach passage-level citations to its claims. A lightweight confidence heuristic is computed from the fused retrieval scores and the degree of agreement between the two retrievers: queries where both retrievers strongly agree on the same top passages are classified as high-confidence, while queries with weak or divergent retrieval signal are flagged for the operator's attention rather than answered with unwarranted certainty.

III. PRIOR APPROACH

Keyword-based retrieval built on the Probabilistic Relevance Framework, and in particular the BM25 ranking function, remains a strong and well-understood baseline for text search [4]. BM25 scores a document by the frequency of query terms within it, normalized against document length and corpus-wide term rarity, and it requires no training data or embedding model — a practical advantage when a corpus is small or domain vocabulary is unusual.

Dense, embedding-based retrieval represents queries and passages as vectors in a learned semantic space, allowing a system to match meaning rather than surface wording. Dense Passage Retrieval demonstrated that a dual-encoder trained on question–passage pairs could outperform a strong Lucene-BM25 baseline on open-domain question answering benchmarks [3], establishing dense retrieval as a serious complement to sparse methods.

RAG was formalized by Lewis et al. as a general-purpose recipe that couples a parametric sequence-to-sequence model with a non-parametric retrieval index, allowing the generator to condition on retrieved passages rather than on memorized facts alone [2]. This grounding strategy directly targets the hallucination problem widely documented in the large language model literature. The Transformer architecture underlying essentially all contemporary LLMs was introduced by Vaswani et al. and remains the substrate on which both retrievers and generators are built [1].

Because sparse and dense retrievers make different kinds of mistakes, combining their outputs — hybrid search — tends to outperform either retriever alone. Reciprocal Rank Fusion, introduced by Cormack, Clarke, and Büttcher, merges ranked lists from multiple retrieval systems using only rank position rather than raw, differently-scaled relevance scores, and was shown to consistently outperform both individual rankers and the previous standard, Condorcet fusion [5].

Once a corpus grows beyond what fits comfortably in a brute-force scan, approximate nearest-neighbour libraries such as FAISS become necessary for practical query latency [6]. A growing body of work extends RAG beyond a single retrieve-then-generate pass toward pipelines in which an agent reflects on retrieved evidence and plans multi-step retrieval. Enterprise adoption of these patterns is still largely experimental, with comparatively few deployments reporting rigorous, production-grade validation of generative output quality.

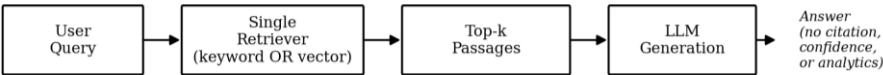


Figure 1: Existing (Single-Path) RAG Mechanism

As Figure 1 illustrates, most existing RAG deployments follow a single, fixed retrieval path: one retriever — either keyword-based or embedding-based, rarely both — feeds its top-k passages directly to the generator. There is no fusion step to reconcile lexical and semantic relevance, no run-time mechanism to reweight the retrievers, and typically no operator-facing signal of how confident the system is in a given answer. Section IV describes how the proposed architecture departs from this pattern.

IV. OUR APPROACH

Within the landscape surveyed in Section III, the specific combination this paper targets — a small, operator-tunable hybrid retrieval pipeline paired with an always-on analytics layer — remains comparatively under-explored, which is the gap this prototype was built to probe. Figure 2 shows the proposed architecture, contrasted directly with the single-path mechanism of Figure 1: two retrievers run in parallel, a configurable fusion layer reconciles their rankings, and an analytics dashboard observes every stage of the pipeline.

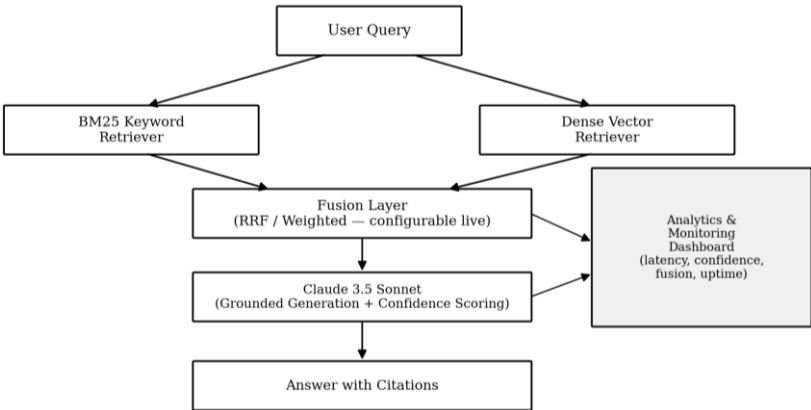


Figure 2: Proposed Hybrid RAG Architecture

Table 1 summarizes the maturity of each system component at the time of evaluation.

Table 1: Component Status Summary

Aspect	Technology	Status
LLM	Claude 3.5 Sonnet	Production-ready
Embeddings	Placeholder (128-dim)	Prototype
Vector store	In-memory	Prototype
Keyword search	Custom BM25	Complete
RAG framework	Custom, modular	Complete
Document sources	10 static policy documents	Extensible
Analytics dashboard	Real-time KPI dashboard	Complete
Fusion	RRF and weighted fusion	Configurable live

4.1 Analytics and Monitoring Dashboard

A real-time dashboard aggregates system behaviour across queries: per-query and rolling-average latency, the distribution of confidence classifications, which fusion strategy is active and how it is performing, query volume over time, and service uptime. This turns the RAG pipeline from a single-shot chat interface into an observable service, a feature more often associated with mature commercial platforms than with early-stage academic prototypes.

4.2 Preliminary Pilot Evaluation

We emphasize that the results below come from a single-developer pilot test conducted by two contributors, not a controlled or multi-participant study, and should be read as an early signal of feasibility rather than a statistically

generalizable benchmark. Testing consisted of roughly 30 to 40 queries against the ten-document corpus over approximately two hours of active use. Table 2 summarizes the measured operational metrics.

Table 2: Pilot Evaluation — Operational Metrics

Metric	Observed Value	Target / Note
Mean end-to-end latency	2.0 s	Target: < 3 s
Top-1 retrieval score	75% – 95%	Range across test queries
High-confidence rate	~70% of queries	Per confidence heuristic
System uptime	100%	Zero crashes observed
Queries tested	~30–40	Two testers, ~2 hours
Corpus size	10 documents	Enterprise policy documents

Citations attached to generated answers were consistently accurate against the underlying passages, and grounded generation kept unsupported claims rare within the tested query set. Two limitations were noted directly: retrieval recall was constrained by the small, ten-document corpus, and retrieval precision was affected by the use of placeholder rather than production-grade embeddings. The small sample size, absence of a baseline comparison, and reliance on placeholder embeddings are treated as the immediate priorities for the next evaluation cycle rather than as flaws that undermine the architecture itself.

4.3 Discussion

The pilot results are best read as a feasibility signal rather than a performance claim. That the two weakest-flagged aspects of the system — retrieval precision and recall — trace directly to the two components explicitly marked prototype in Table 1 suggests the architecture itself is not the limiting factor, and that the planned migration to production embeddings and a FAISS-backed index is a targeted rather than open-ended fix. On the question of autonomy, the system autonomously executes retrieval, fusion, generation, and confidence scoring for every query without per-query human routing, but it does not yet reflect on its own retrieved evidence or coordinate multiple specialized agents — capabilities that define the more ambitious agentic RAG pattern illustrated in Figure 2 above.

V. CONCLUSION

This paper presented the Autonomous RAG-based Enterprise Search & Analytics system, a prototype hybrid-retrieval pipeline that fuses BM25 keyword search with dense vector search through a configurable, run-time-switchable fusion layer, grounds Claude 3.5 Sonnet's generation in the fused retrieval results with passage-level citations, and exposes the whole pipeline's behaviour through a real-time analytics dashboard. A preliminary pilot evaluation over a ten-document enterprise corpus and roughly 30–40 queries showed a mean latency of 2.0 seconds, top-1 retrieval scores of 75–95%, a high-confidence classification rate of approximately 70%, and zero system failures — results presented as an early, honest feasibility signal rather than a mature benchmark.

Future work will prioritize replacing the placeholder embeddings with a production embedding model and migrating from the in-memory store to FAISS [6], expanding the document corpus and running a multi-participant user study, evaluating the hybrid pipeline against keyword-only and vector-only baselines, and moving the system from single-pass autonomy toward genuine agentic behaviour through reflection and multi-step query planning. The broader aim of this project is a lightweight, transparent, and operator-tunable alternative to opaque enterprise RAG platforms.

ACKNOWLEDGEMENTS

This work was completed as part of the third-year Project-Based Learning (PBL-3) course at Sharda University during the 2026–2027 academic session. The authors thank the Department of Computer Science & Engineering, SSET, Sharda University, for the resources and support provided throughout the development and evaluation of this project.

REFERENCES

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.
- [2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474.

-
- [3] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, (2020). Dense passage retrieval for open-domain question answering. *Proc. 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 6769–6781.
 - [4] S. Robertson and H. Zaragoza, (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4), 333–389.
 - [5] G. V. Cormack, C. L. A. Clarke, and S. Büttcher, (2009). Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. *Proc. 32nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 758–759.
 - [6] J. Johnson, M. Douze, and H. Jégou, (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), 535–547.