Optimizing RAG for Academic Advising: A Hybrid Routing and Metadata Filtering Approach for Enhanced Accuracy and Efficiency

Anuraj Manish Sule*, Abdul-Rahman Mawlood-Yunis
Wilfrid Laurier University, Ontario, Canada

Abstract

Professional academic advisors are increasingly burdened by complex policy inquiries and high student volumes. While Retrieval-Augmented Generation (RAG) offers a solution, standard "brute-force" architectures suffer from high latency and "contextual noise" due to global database searches. This paper introduces a *Hybrid Routing Layer* that optimizes retrieval by pre-filtering queries through a multi-stage "waterfall" logic. The system utilizes a high-speed Regex Router for entity-specific identifiers (e.g., course codes) and a Semantic Router for intent-aware policy mapping. Validated on real-world queries from Wilfrid Laurier University, our approach reduces the search space by 97% and improves end-to-end latency by 7x (from 8.2s to 1.3s) while significantly increasing retrieval precision. This framework provides a scalable, low-cost solution for accurate AI-driven decision support in higher education.

Keywords: Academic Advising, Retrieval-Augmented Generation (RAG), AI Efficiency, Decision Support, Query Routing.

1. Introduction

Academic advising is essential for student success, yet professional advisors are often overwhelmed by a high volume of repetitive inquiries. While Large Language Models (LLMs) offer a scalable solution for automated support, standard Retrieval-Augmented Generation (RAG) frameworks suffer from two primary limitations: high latency and "contextual noise" that leads to hallucinations [1]. Traditional "brute-force" RAG architectures, which search an entire database for every query, scale poorly as the knowledge base grows. These systems often retrieve semantically similar but factually irrelevant information from unrelated departments, a risk that is unacceptable in an academic policy context.

To address these challenges, this paper introduces a *Hybrid Routing Layer* designed as an intelligent pre-retrieval filter. Our approach utilizes a "waterfall" logic that combines high-speed Regular Expressions (Regex) for entity detection with semantic similarity for intent-aware routing. By identifying specific identifiers like course codes or policy titles before retrieval begins, the system narrows the search space to a single relevant source, effectively eliminating the noise that causes errors.

This work contributes a novel hybrid architecture that reduces the computational search space by over 97% and a waterfall decision strategy that prioritizes speed without sacrificing accuracy. Through an empirical evaluation using real-world advising queries, we demonstrate that this framework achieves a 7x reduction in latency and significant improvements in retrieval precision compared to standard RAG baselines, providing a scalable and low-cost solution for academic decision support.

* yashsule40@gmail.com

2. Related Work

The integration of Large Language Models (LLMs) in academic advising addresses the limitations of rule-based chatbots by enabling natural dialogue [2]. To mitigate hallucinations, Retrieval-Augmented Generation (RAG) has become the standard for grounding LLM outputs in trusted policy databases [3, 4]. However, traditional "brute-force" RAG architectures, which search the entire index for every query, present significant latency and precision bottlenecks in production environments.

Recent advancements in "Adaptive RAG" attempt to solve this by dynamically adjusting retrieval strategies. Frameworks such as Self-RAG [5] and Adaptive-RAG [6] utilize "critic" models or complexity classifiers to determine retrieval necessity. While effective at improving accuracy, these methods often rely on auxiliary LLM calls for routing, which introduces unacceptable latency for real-time advising. Furthermore, benchmarks like BEIR [1] indicate that dense vector retrieval often struggles with exact identifier matching (e.g., distinguishing "CP104" from "CP164"). While hybrid search combining keywords and vectors is a known mitigation [7], our work uniquely utilizes this logic as a pre-retrieval metadata filter. By replacing heavy AI-based routers with a lightweight Regex and semantic waterfall, we achieve zero-noise retrieval and significantly lower the "time-to-first-token" compared to current adaptive methods.

3. Methodology

To address efficiency and precision bottlenecks in standard RAG, we introduce a query-aware framework centered on a *Hybrid Routing Layer*. This architecture pre-processes user queries to determine the optimal search strategy ranging from targeted metadata filtering to broad semantic search before retrieval begins [8].

3.1. Data Pipeline and Baseline System

The knowledge base was constructed by scraping Wilfrid Laurier University (WLU) web pages, resulting in over 10,000 PDF documents (e.g., syllabi and policy handbooks). These were segmented into text chunks and indexed in a ChromaDB vector store using `all-MiniLM-L6-v2` embeddings. We define our baseline as a standard "brute-force" RAG pipeline: a global search retrieves $k = 20$ chunks, which are re-ranked via a CrossEncoder model to select the top 5 for generation by a Mistral LLM. This baseline triggers an expensive global search for every query, regardless of complexity.

3.2. Hybrid Routing Layer

The Hybrid Routing Layer analyzes user intent to decide between a filtered or global search strategy via two primary components.

1) Contextual Query Handling: To support conversational continuity, the system identifies follow-up queries using keyword detection (e.g., "what about"). If detected, the system generates a contextualized query by prepending the previous interaction to the current input, ensuring the router possesses the necessary context for classification.

2) Hybrid Waterfall Router: The router employs a sequential "waterfall" logic to direct queries toward the most efficient path (see Figure 1). First, a *Complex Query Check* scans for keywords (e.g., "compare," "difference") or multiple course codes; if found, the system defaults to a global search. Second, a high-speed *Regex Router* scans for specific identifiers (e.g., "CP104"). If matched, a metadata filter is applied, reducing the search space by over 97%. Third, for non-specific inquiries, a *Semantic Router* calculates cosine similarity between the query and pre-embedded document titles. Using a threshold of $\tau = 0.75$ within

an HNSW index, the system attempts to lock retrieval to a specific document. If no matches occur, a *Global Fallback* ensures a standard search is performed.

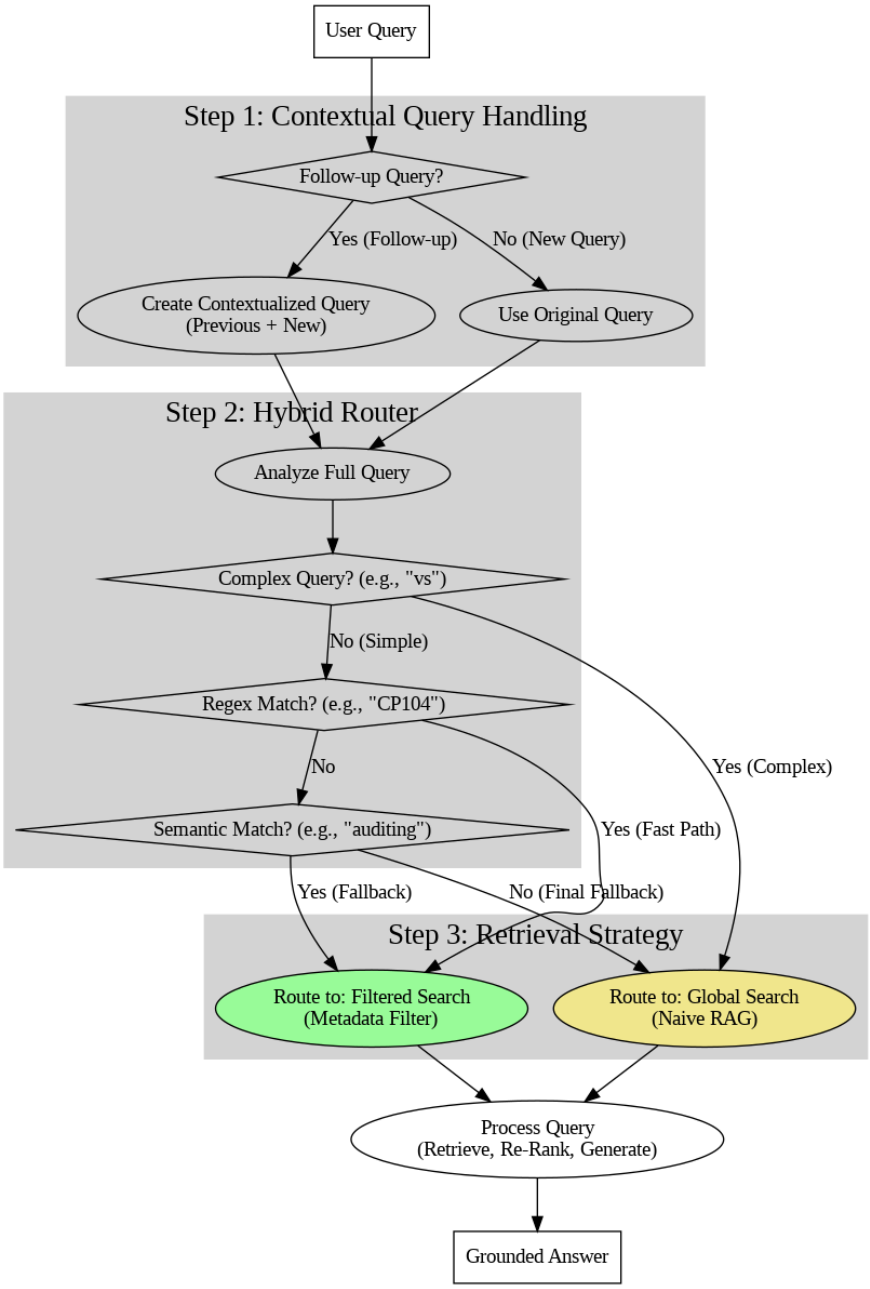

Fig. 1. Hybrid Routing Logic

Figure 1. The Hybrid Routing Logic: transitioning from initial query to filtered or global search paths.

3.3. Optimized Retrieval Pipeline

The routing layer modifies the retrieval scope dynamically. In a *Filtered Search*, the retriever is constrained to chunks matching the identified metadata tag (e.g., `source: cp104.pdf`). This eliminates "contextual noise" and reduces latency by minimizing the number of vectors processed during similarity search and re-ranking.

4. Performance Evaluation

We evaluated the Hybrid Routing Layer against a standard "brute-force" RAG baseline [8] across three metrics: accuracy, latency, and computational efficiency.

4.1. Experimental Setup

A dataset of $N = 100$ student queries was synthesized through interviews with professional academic advisors to ensure real-world relevance. Queries were categorized into Course-Specific (30), Policy-Specific (30), and Complex/Vague (40). Retrieval precision was measured using an "LLM-as-a-Judge" methodology with GPT-4, which graded retrieved chunks on a binary relevance scale. To validate the judge, we manually reviewed a random sample of 25% of the queries, achieving a 95% agreement rate between human advisors and the automated score.

4.2. Results and Discussion

The experimental results confirm that pre-retrieval routing transforms system performance across all categories.

1) Accuracy and Precision: As shown in Figure 2, Hybrid RAG achieved 98% precision on course-specific queries, whereas the baseline dropped to 65% due to identifier confusion (e.g., CP104 vs. CP164). On complex queries, Hybrid RAG maintained 85% relevance compared to 70% for the baseline. This trend reversal where the baseline performs better on vague queries than specific ones occurs because global search lacks the granularity to distinguish between similar alphanumeric identifiers, while the Hybrid Router locks onto the correct document context immediately. Table 1 highlights these successes, though failures occurred in 2% of cases where user typos in course codes bypassed the Regex router.

Query	**Baseline (Global)**	**Hybrid (Filtered)**
Prerequisites for CP312?	CP104, CP164 (CP212 file)	CP213, CP216 (Correct)
Late penalty CP104?	5% per day (Generic)	Not accepted (Course-specific)

Table 1. Qualitative comparison of retrieval accuracy.

2) Latency and Efficiency: Speed is measured by "Time-to-First-Token" (TTFT). Figure 3 shows the median latency dropped from 8.2s (baseline) to 1.3s (Hybrid), a 7x improvement. Furthermore, the Hybrid system searched 97.5% fewer vectors (average 85 vs. 3,500), as illustrated in Figure 4. This reduction demonstrates that the system is model-agnostic and scalable; adding documents increases the baseline's cost linearly while the Hybrid model's filtered search remains constant.

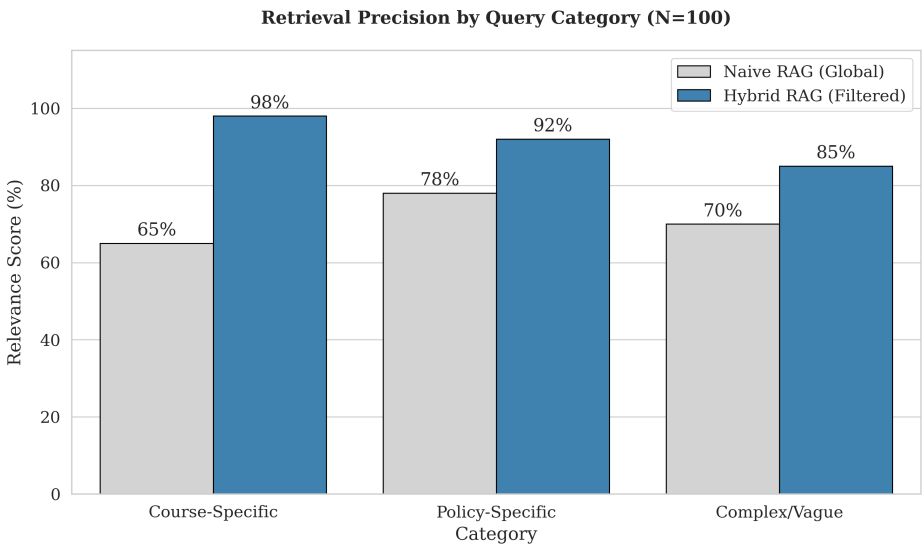

Figure 2. Retrieval Precision ($N = 100$). Hybrid RAG significantly outperforms the baseline in course-specific accuracy.

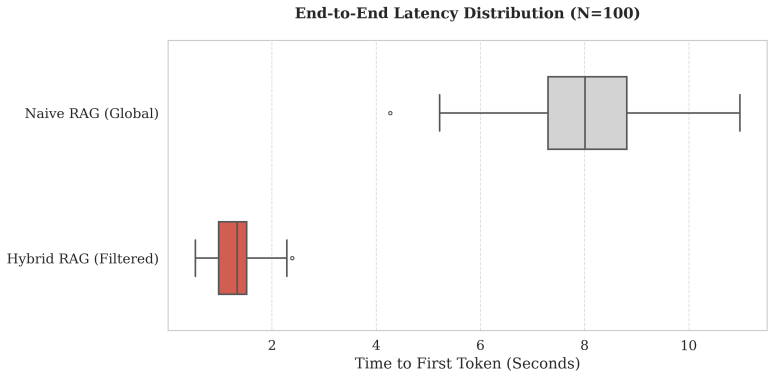

Figure 3. End-to-End Latency Distribution (s).

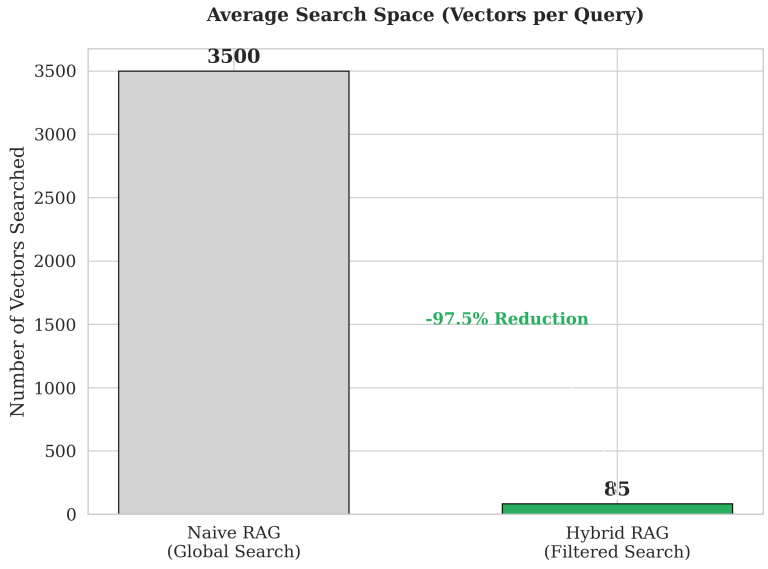

Figure 4. Vector Search Space Reduction.

5. Conclusion

This paper addressed the latency and precision bottlenecks in standard RAG architectures by introducing a *Hybrid Routing Layer*. While brute-force retrieval introduces contextual noise and high latency, our methodology utilized a high-speed Regex and semantic waterfall logic to narrow the search space to a single relevant document. Our evaluation demonstrated that this approach achieves a 7x reduction in query latency and significant improvements in precision by eliminating irrelevant context.

Beyond academic advising, this framework is generalizable to any domain requiring exact identifier matching and high-stakes accuracy, such as legal or technical documentation. This contribution transforms the RAG-powered advisor into a high-performance system practical for real-time decision support. Future work will focus on improving the semantic router's handling of vague queries and developing a student-facing web interface.

References

[1] Thakur, N., et al.: BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. NeurIPS (2021).
[2] Khare, R., et al.: Enhancing Academic Advising with AI. OLA (2025).
[3] Sule, A., et al.: Enhancing Academic Assistance with RAG. DIS (2025).
[4] Bilquise, G., et al.: Bilingual AI-Driven Chatbot for Academic Advising. IJACSA (2022).
[5] Lewis, P., et al.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS (2020).
[6] Asai, A., et al.: Self-RAG. ICLR (2024).
[7] Jeong, S., et al.: Adaptive-RAG. NAACL (2024).
[8] Omrani, P., et al.: Hybrid RAG Approach for LLMs. ICWR (2024).
[9] Touvron, H., et al.: LLaMA. arXiv:2302.13971 (2023).
[10] Jiang, A. Q., et al.: Mistral 7B. arXiv:2310.06825 (2023).
[11] OpenAI: GPT-4 Technical Report. arXiv:2303.08774 (2023).
[12] Chroma Authors: Chroma Database (2023).
[13] Gao, Y., et al.: RAG for LLMs: A Survey. arXiv:2312.10997 (2023).