RAG Signal™ Adaptive RAG Architecture: Multi-Source, Temporally Aware Vector Knowledge Base for Enhanced Context Quality

Bora Kurum

July 26, 2026

Abstract

This document presents the RAG Signal™ Adaptive RAG architecture, a production-grade Retrieval Augmented Generation (RAG) system designed to improve context quality in Large Language Model (LLM) based content generation systems. Addressing core limitations of classical RAG implementations — static indices, single-source dependency, task-agnostic retrieval, and temporal blindness — this architecture resolves these issues through multi-source differential weighting, an advanced temporal freshness function with 180-day hard cutoff, task-oriented query enrichment, normalized similarity bounds via ReLU clamping, hybrid ranking mechanisms with semantic-keyword fusion, and an autonomous KB Curator Agent. The architecture has been validated across ten production deployments spanning seven industry sectors, demonstrating a 77.1% average LLM attribution rate across 63–105 prompts per deployment.

License: Apache License 2.0. This paper serves as a Defensive Publication to establish prior art.

GitHub: `https://github.com/borakurum-source/ragsignalseo`

1 Introduction

The rapid proliferation of generative AI systems has fundamentally reshaped the digital information landscape. Users increasingly rely on synthesized answers from these models rather than ordered lists of links. This paradigm shift makes brand representation in LLM-generated content a critical competitive concern. Traditional Search Engine Optimization (SEO) is inadequate for this new reality, leading to the emergence of Generative Engine Optimization (GEO) [1, 2, 8, 11].

Most production RAG deployments exhibit four critical limitations that the RAG Signal architecture directly addresses:

1. **Static indices:** No automatic updates when content changes — resolved via djb2 content-hash-based incremental indexing.
2. **Single-source dependency:** Only site content is indexed — resolved via multi-source ingestion with differential weighting.
3. **Task-agnostic retrieval:** The sole ranking criterion is cosine similarity; task context is ignored — resolved via task-oriented query enrichment.
4. **Temporal blindness:** Old and new data are weighted equally — resolved via rational decay freshness scoring with a 180-day hard cutoff.

2 Related Work

Retrieval Augmented Generation (RAG) systems have become a significant research area for addressing the issues of information currency and accuracy in Large Language Models (LLMs). While traditional RAG approaches often rely on static knowledge bases and simple similarity metrics, recent work has focused on overcoming these limitations [3, 4, 7].

Adaptive RAG approaches aim to dynamically adjust retrieval strategies based on query context or user feedback. SPARC-RAG [5] offers adaptive sequential-parallel scaling with context management to address inference-time RAG scaling challenges. The RAG Signal™ architecture extends this paradigm by incorporating multi-source differential weighting and task-aware query augmentation.

Temporal RAG optimizes retrieval processes by considering the recency of information and its evolution over time. STAR-RAG [6], proposed by Zulun Zhu et al., integrates temporal constraints into the retrieval process by constructing time-aligned rule graphs. Chronofy [12] has deepened this field by using temporal-logical decay for information validity.

3 System Architecture

The system consists of five primary components: Data Ingestion and Indexing, Vector Representation, Advanced Retrieval with Composite Scoring, KB Curation, and Generation with Closed-Loop Feedback.

3.1 Robust Hash-Based Incremental Updates

A djb2 content hash of the raw page content serves as a fingerprint, enabling incremental re-indexing:

$$H(content) = \text{djb2}(raw\_content)$$

The djb2 algorithm ($hash = 5381$; for each byte c: $hash = (hash \times 33) \oplus c$) is selected for speed, simplicity, and collision resistance at production scale. When a page is re-crawled and its hash matches the stored hash with an existing embedding, the page is skipped entirely.

3.2 Vector Representation

The primary embedding model produces 1,024-dimensional vectors via Jina AI's `jina-embeddings-v3`, selected for multilingual capabilities and strong factual retrieval performance. Vectors are stored in PostgreSQL with the pgvector extension and HNSW indexing.

3.3 Normalized Similarity with ReLU Clamping

Similarity between query embedding q and chunk embedding c is measured by cosine distance. To prevent negative cosine similarities from producing meaningless scores, semantic similarity is clamped:

$$s'(q, c) = \max(0,\ \text{sim}(q, c))$$

This normalization ensures all downstream operations operate on well-behaved, non-negative values.

3.4 Differential Source Weighting

Source trustworthiness weights dynamically adjust based on source type:

Source	Weight	Rationale
`feedback`	1.5	Strongest signal — user endorsement
`gsc`	1.3	Real user query behavior data
`prompt`	1.1	Curated, task-specific context
`firecrawl`	1.0	Neutral baseline — web crawl
`manual`	1.0	Manually entered content
`audit`	0.8	May contain speculative claims

3.5 Temporal Freshness Scoring with Hard Cutoff

Data recency is quantified via a rational decay function with a 180-day hard cutoff:

$$f(\Delta t) = \begin{cases} \dfrac{1}{1 + \Delta t/30} & \text{if } \Delta t < 180 \\ 0 & \text{if } \Delta t \geq 180 \end{cases}$$

Age	Score
Today	1.00
30 days	0.50
60 days	0.33
90 days	0.25
180+ days	0.00

The 180-day hard cutoff prevents heavily outdated data from remaining in the system at diminishing but non-zero scores, ensuring genuinely current retrieval results.

3.6 Task-Oriented Query Enrichment

Raw user queries are augmented with task-specific keyword suffixes before embedding:

Task	Augmentation Suffix
`meta`	page title description content keywords
`discovery`	services audience USP keywords brand positioning
`audit`	technical issues recommendations content quality
`hallucination`	facts services team location founding year
`score`	content structure readability entities schema

3.7 Composite Ranking Score

The mathematical pre-ranking score is the product of three independent factors:

$$\text{score}(c) = s'(q, c) \times w(source(c)) \times f(updated\_at(c))$$

The top $2k$ candidates (where k is the desired result count) proceed to LLM re-ranking.

3.8 Hybrid Ranking

Retrieval uses a 70/30 semantic-to-keyword weighted hybrid approach via PostgreSQL's `match_brand_knowledg` function, ensuring both conceptually relevant and term-exact content is surfaced.

3.9 LLM Re-Ranking

After mathematical pre-ranking produces the top $2k$ candidates, an LLM-based re-ranker (DeepSeek Reasoner) performs contextual relevance assessment. On re-ranker failure, the system gracefully degrades to mathematical ranking.

3.10 KB Curator Agent

An autonomous curation pipeline classifies each ingested chunk as **keep**, **improve**, or **delete**. Navigation menus, cookie notices, boilerplate, and duplicate content are removed. Improved chunks are re-embedded with fresh vectors. The curator is idempotent — only uncurated content ($curated = null$) is processed.

4 Implementation

The RAG Signal™ Adaptive RAG architecture is implemented as Supabase Edge Functions (TypeScript/Deno), open source under Apache 2.0 at `https://github.com/borakurum-source/ragsignalseo`.

Component	File
Core RAG engine	`functions/_shared/rag.ts`
KB Curator Agent	`functions/kb-curator/index.ts`
Unified KB API	`functions/kb-api/index.ts`
Brand indexing (djb2)	`functions/index-brand-knowledge/index.ts`
Hybrid search (SQL)	`migrations/*hybrid_search*.sql`
Embedding schema	`migrations/*embedding_1024*.sql`
KB noise detection	`functions/_shared/kb-noise.ts`

5 Experimental Results

The RAG Signal architecture has been validated across ten production client deployments representing seven diverse industry sectors.

Controlled A/B Tests: Four deployments (Filmfolk, Bora Kurum, ABS Void Formwork, ABS Kör Kalıp) showed significantly higher attribution rates compared to baseline LLM output without RAG. Attribution was measured via Mistral Large with web search across 63–105 branded prompts per deployment.

Production Monitoring: Six additional deployments in live environments consistently recorded high attribution rates across weekly measurement cycles.

Method	Deployments	Avg. Attribution
A/B Test (Controlled)	4	81.3%
Production Monitoring	6	74.3%
Overall	**10**	**77.1%**

6 Conclusion

The RAG Signal™ Adaptive RAG architecture successfully addresses the four core limitations of traditional RAG systems through djb2 hash-based incremental updates, multi-source differential weighting, ReLU-clamped normalized similarity, task-oriented query enrichment, temporal

freshness with 180-day hard cutoff, hybrid semantic-keyword fusion, and autonomous KB curation. With a 77.1% average attribution rate across ten production deployments, the architecture provides a robust, validated framework for the next generation of Generative Engine Optimization.

References

[1] O. Martinez, "Optimizing Visibility in Generative Engines: A Critical Survey of GEO (2023–2026)," arXiv:2607.14035, 2026.

[2] Z. Tian, Y. Chen, Y. Tang, J. Liu, and R. Jia, "Diagnosing and Repairing Citation Failures in GEO," arXiv:2603.09296, 2025.

[3] "What is RAG? Latest Advances in Retrieval-Augmented Generation," Squirro Blog, 2026.

[4] "How Retrieval-Augmented Generation Works for Enterprise AI," Techment Blog, 2026.

[5] "SPARC-RAG: Adaptive Sequential-Parallel Scaling with Context Management for RAG," arXiv:2602.00083, 2026.

[6] Z. Zhu, H. Liu, M. He, and S. Luo, "Right Answer at the Right Time — Temporal RAG via Graph Summarization," arXiv:2510.16715, 2025.

[7] "Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG," arXiv:2501.09136v4, 2026.

[8] "GEO Guide 2026: Generative Engine Optimization Explained," Digital Applied Blog, 2026.

[9] "Generative Engine Optimization (GEO): The 2026 Guide," LLMRefs, 2026.

[10] "Generative Engine Optimization Statistics (2026)," Omnibound AI Blog, 2026.

[11] "Mastering generative engine optimization in 2026: Full guide," Search Engine Land, 2026.

[12] M. Syed et al., "Chronofy: A Temporal-Logical Decay Architecture for Information Validity in Time-Aware RAG," arXiv:2607.20560, 2026.