A pragmatic guide to enterprise search that works

BEN LORICA
SEP 09, 2025

Newsletter

Subscribe • Previous Issues

The Enterprise Search Reality Check

Before the AI hype cycle exploded with ChatGPT in late 2022, I was focused on a less glamorous, but equally important shift: the resurgence of enterprise search. Neural retrieval and vector embeddings finally looked practical. After the release of ChatGPT, an assumption among some AI teams was that these powerful new models would solve the long-standing “enterprise search” problem. AI teams dove into fine-tuning, Retrieval-Augmented Generation (RAG), and agentic frameworks, expecting to conquer the corporate knowledge base. But despite the incredible advances in foundation models, enterprise search remains stubbornly difficult. It's a baffling disconnect: the same model that can eloquently explain quantum mechanics is often unable to give a straight answer to a seemingly simple question like, "What are our current quarterly goals?" After interviewing some founders and engineers working on this problem, I've discovered why. The real obstacles aren't what you'd expect.

1. The Foundational Rot: It's a Data Quality Problem, Not a Model Problem

The core issue in enterprise search is the nature of the data itself. Unlike the public web, where pages have clear owners and URLs serve as stable identifiers, enterprise information lacks clear ownership, governance, and structure. For example, a system might contain three different versions of a "Q3 Sales Strategy" document — a draft in a shared drive, an outdated wiki page, and a final PDF in an email. This inherent ambiguity is compounded by "shadow documents" created by employees when they can't find the original, further polluting the knowledge base. Staleness and duplication create a low-signal environment where even strong retrievers struggle to find ground truth. This isn’t a failing of an algorithm; it’s a reflection of the input. *Garbage in, garbage out.*

This reality forces a shift in focus from the AI model to the data foundation. One approach is organizational: appointing dedicated "Knowledge Managers" to curate critical information, establishing clear governance, and building a culture of data hygiene. The other is architectural: implementing systems like knowledge graphs that programmatically create structure by identifying entities (people, projects, documents) and mapping their explicit relationships. Graphs generate the reliable signals — like "Engineer A owns Jira Ticket B" — that are missing from unstructured text, creating a trustworthy foundation before a language model is ever involved. Without this foundational work, any search initiative is built on sand.

A heartfelt thank you to our paid subscribers. Join them to support our work and expand your knowledge.

2. The Signal Problem: Why Enterprise Ranking Fails

Web search thrives on a rich set of signals: PageRank, click-through rates, backlinks, and user behavior at a massive scale. Enterprise environments have none of this. Relevance is deeply contextual and ambiguous. Is a new document from a CEO more important than a five-year-old, battle-tested engineering policy? The answer depends entirely on who is asking and why. A sales executive searching for "quarterly goals" needs a completely different result than a software engineer using the same query. This lack of clear, universal authority signals means that simple retrieval methods, whether keyword-based or basic vector similarity, often fail, returning results that are semantically related but contextually useless.

To overcome this, teams use multi-faceted, hybrid retrieval systems. They might use **BM25** for exact phrase matching (crucial for finding specific contract clauses), **dense embeddings** for conceptual similarity (helpful when users don't know the exact terminology), and (**knowledge**) **graph traversal** for authority-based discovery (finding documents through trusted authors or recent approvals).

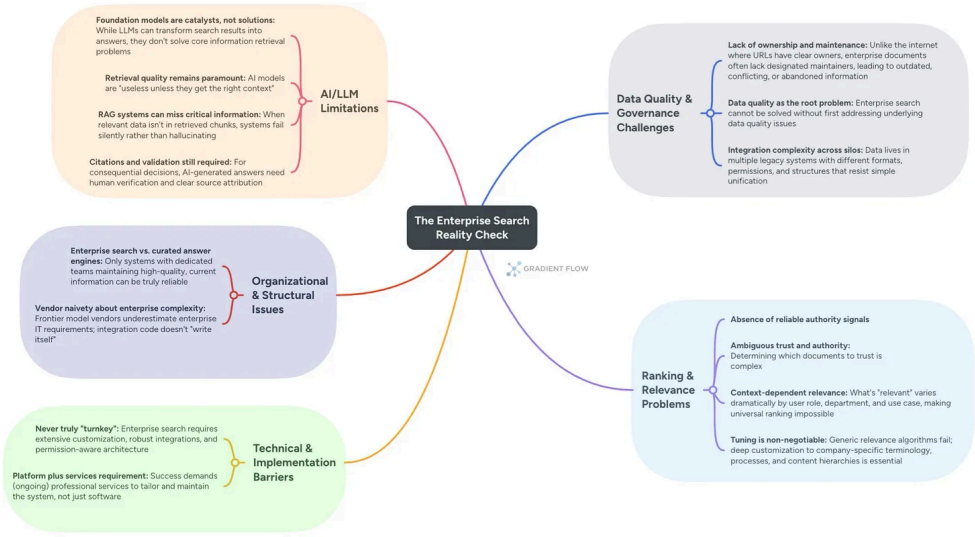
Advanced systems add an **"instructable reranker"** layer that can be explicitly programmed with business logic. A pharmaceutical company might configure their reranker to always prioritize FDA-approved documents over internal research notes, while a law firm might boost documents based on the seniority of the authoring partner. This transforms ranking from an opaque algorithm into a configurable business tool. The most sophisticated systems improve the signals feeding that reranker with **hard-negative mining** (teach the system to tell near-neighbors apart) and **enterprise-tuned embeddings** that understand your acronyms and ontology.

3. The RAG Paradox: A Powerful Tool That Magnifies the Core Problem

RAG and its many variants have become the default architecture for grounding LLMs in private data. However, its effectiveness is entirely dependent on the quality of the initial retrieval step. If the right documents aren't surfaced in the first pass, the system fails. Anant Bhardwaj describes this failure mode as "worse than hallucination" because the model provides a confident, well-written answer based on incomplete or

incorrect information. An employee asking about parental leave policy might receive a perfectly articulated summary of an outdated draft, a dangerously misleading outcome. The system doesn't know what it doesn't know, and the polished output masks the critical omission.

This highlights that RAG is not a magic bullet but a component in a larger system that needs to be engineered for robustness. Some teams respond with long-context models and "just dump more text," which helps recall but inflates cost and latency and could still miss tables, images, or cross-doc dependencies. The more reliable pattern is "RAG 2.0": start with **document intelligence** (layout-aware parsing, section hierarchy, provenance); retrieve with a **mixture of retrievers** to maximize recall; apply a **strong reranker** to enforce your rules; then generate with a **grounded model** that cites sources and is trained to say "I don't know" on insufficient evidence. For recurring questions, seed a curated FAQ/answer bank so common queries don't depend on brittle retrieval at all. For sensitive topics, gate any external lookups with confidentiality filters.



4. The Architectural Shift: From Search Boxes to Curated "Answer Engines"

The goal of enterprise search is evolving beyond just returning a list of links. Employees, now accustomed to tools like ChatGPT, expect direct answers. However, a general-purpose search tool layered over a messy data lake cannot reliably provide them. The liability of providing an incorrect answer often outweighs the efficiency gains of having any answer at all: providing an incorrect or incomplete answer is too high for business-critical functions like HR, finance, or legal compliance.

This is driving a strategic split. Instead of one monolithic "enterprise search," the more practical approach is to build multiple, curated "answer engines" for specific, high-value domains. For example, an HR team might maintain an engine built exclusively on a vetted, up-to-date corpus of official policy documents. This approach treats the problem as one of building a trustworthy, predictable system, where a well-

understood scope and predictable failure modes are more valuable than a high but brittle accuracy score on a broad, uncontrolled dataset.

Employees prefer using a reliable, narrow system over a broad but unpredictable one. For coverage gaps, blend three sources: internal documents; **pre-written expert answers** for anticipated questions; and (when allowed) on-demand external enrichment — kept on a short leash and never used for sensitive queries. This narrows scope, raises trust, and keeps supportable SLAs.

5. The Implementation Reality: Enterprise Search is a Service, Not a Product

Many vendors, particularly those new to the enterprise space, underestimate the sheer complexity of real-world IT environments. Enterprise data is fragmented across dozens of siloed SaaS applications and legacy systems, each with its own APIs, permissions, and quirks. Deploying an effective search system requires deep integration, robust security plumbing that respects fine-grained access controls, and significant customization to align with a company's unique vocabulary and workflows.

This reality has led to two clear trends. First, enterprises are increasingly choosing to *buy* specialized third-party applications rather than *build* their own search solutions from scratch, acknowledging that it is a full-time engineering challenge. Second, the successful model for deployment is a "platform plus services" approach. This combines a strong, flexible software platform with professional services to handle the extensive integration, tuning, and customization required. For AI teams, this means budgeting not just for software licenses, but for the significant engineering effort needed to make it work. As Jakub Zavrel of Zeta Alpha notes, “turnkey” enterprise search solutions rarely survive contact with reality.

6. The Measurement Mandate: Proving Reliability in Your Own Context

The first question a practitioner often asks is, "How good is this model?" The immediate temptation is to check public leaderboards. This is a mistake. A model that aces a public trivia QA benchmark is useless if it can't distinguish between your company's internal 'Project Titan' and the dozen other 'Project Titans' it learned about from the public web. Enterprise success is not measured by open-domain accuracy but by reliability within a specific, messy, and private context. Standard benchmarks fail to test for the things that actually break enterprise systems: procedural multi-step queries, the ability to synthesize answers from multiple documents, and, most importantly, knowing when to say nothing at all because the information is missing or ambiguous.

Enterprise search is never going to be turnkey out of the box. It requires deep customization.

— Jakub Zavrel of Zeta Alpha

The only way to solve this is to stop looking at external leaderboards and start building your own internal evaluation suite. This starts by creating a gold-standard test set from a versioned snapshot of your own knowledge base. This internal benchmark must be designed to probe for common failure modes, including questions that are intentionally unanswerable. To measure relevance, many are moving away from noisy 1-10 scores and toward pairwise comparisons — using either human judges or an LLM to decide which of two results is better for a given query. This creates a clearer signal for what "good" means to your users. Ultimately, trust comes from explainability. The system must be able to provide clear citations and trace the lineage of its answers, proving not just what it knows, but how it knows it.

7. The Next Frontier: From Retrieval to Agentic Workflows

The paradigm for enterprise information access is undergoing its third major shift. The **first** was the search box: "find me a document." The **second**, driven by RAG, was the chatbot: "answer my question." The emerging **third** paradigm is the agent: "do this task for me." This evolution is driven by the need to automate complex business processes that require more than a single query-response loop. Answering a question like, "Summarize the key risks and decisions from the Q3 product planning cycle," is not a single search. It requires finding meeting notes, cross-referencing Slack channels, checking related project tickets, and synthesizing a coherent narrative from these disparate sources.

This leap from simple retrieval to multi-hop reasoning requires a new architecture. Instead of a monolithic RAG pipeline, teams are building agentic systems that treat retrieval as one of many "tools" an orchestrator can use. In this model, an agent can plan and execute a sequence of steps: query a database, look up a file, parse its contents, and then feed the synthesized context to a language model. These workflows are often encoded as repeatable graphs (DAGs) to ensure reliability and support human-in-the-loop checkpoints. This is the true endgame for enterprise AI: not just to make information findable, but to put that information to work, automating the complex knowledge-based tasks that impact business metrics.

What AI teams should internalize

Enterprise search is fundamentally a systems engineering and data governance challenge that happens to use AI, not an AI problem that happens to involve data. Foundation models have transformed what is possible — turning search results into conversational answers — but they have not eliminated the hard parts. If anything, they have raised the stakes. An incorrect answer from a chatbot is a nuisance; an automated action from an agent based on flawed data is a liability. This is why the most mature teams are shifting their focus from chasing leaderboard scores to building rigorous, internal evaluation frameworks that prize reliability over occasional brilliance.

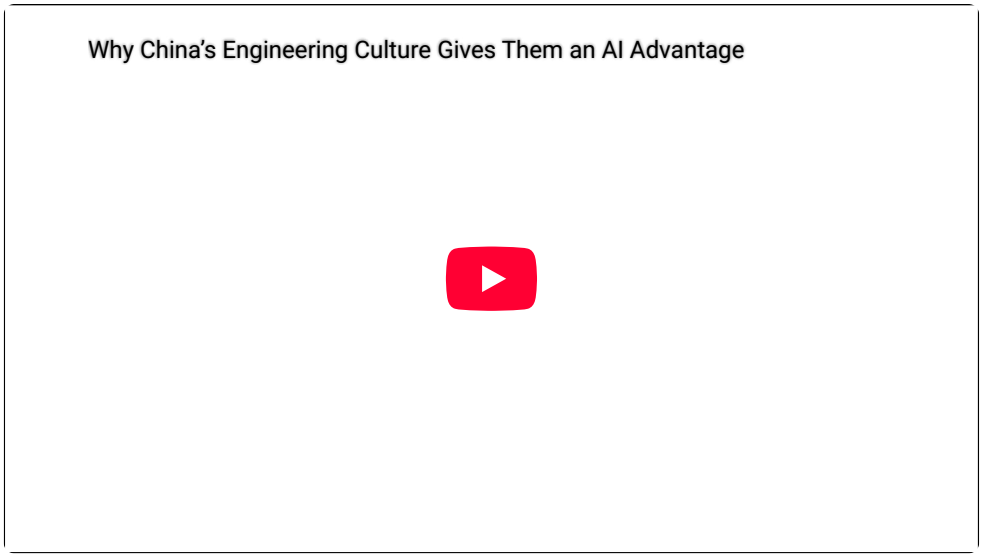
The enterprises succeeding with AI-powered search are not those with the biggest models, but those that have accepted the messy reality of their data and built systems designed for predictability and trust. They understand that the true endgame is not

just to find documents, but to build an auditable, trustworthy foundation upon which reliable automation can be built. They are engineering the information supply chain for an agentic future.

Pragmatic steps to take now

- **Start with a Data Census, Not a Model Evaluation.** Inventory your critical knowledge sources, identify owners (if they exist), and understand update cadences and access controls. The gaps you find will define the real scope of your challenge.
- **Ship Hybrid Retrieval with Reranking.** Your RAG system's intelligence is capped by what it retrieves. Combine keyword, dense, and graph approaches; add an instructable reranker and hard-negative mining. A brilliant language model working with the wrong documents is worse than useless.
- **Stand Up One Curated Answer Engine.** Build narrow and deep before going broad and shallow. Pick a high-value, well-bounded use case like HR or IT support; restrict its sources, require citations, and implement "I don't know" as a feature, not a limitation.
- **Evaluate Privately and Continuously.** Version your knowledge bases and build internal benchmarks that include unanswerable questions and multimodal data. Prioritize predictable failure over unpredictable brilliance; a system that is right 80% of the time with understood failure modes is more valuable than one that is right 90% of the time but fails randomly.
- **Think in Workflows, Not Just Answers.** Before building a complex agent, map the human process it is meant to replace. Start by augmenting that workflow with reliable, single-step tools before attempting full, end-to-end automation.
- **Budget for Integration and Stewardship**. Whether building or buying, expect platform-plus-services costs. Assume you will spend as much on integration, customization, and maintenance as on core technology. Treat any promise of a "turnkey" solution with healthy skepticism; it likely signals a misunderstanding of the problem's depth.

Quick Takes



Evangelos Simoudis and I cover these three topics:

1. The "AI Governance Industrial Complex": Who Should Regulate AI

2. Dan Wang's "Breakneck": Inside China's Engineering-Led AI Quest

3. The Recent MIT Survey: What to Do When AI Value Doesn't Match the Hype

Ben Lorica edits the Gradient Flow newsletter and hosts the Data Exchange podcast. He helps organize the AI Conference, the AI Agent Conference, the Applied AI Summit, while also serving as the Strategic Content Chair for AI at the Linux Foundation. You can follow him on Linkedin, X, Mastodon, Reddit, Bluesky, YouTube, or TikTok. This newsletter is produced by Gradient Flow.

3 Likes · 1 Restack

← Previous

Robotics Is Becoming AI's Ultimate Testing Ground

BEN LORICA
SEP 02, 2025

7 4 Share