What managers should know about RAG: A complement to AI models

Retrieval-Augmented Generation shifts the enterprise AI question from "which model should we use?" to "how do we connect AI models to what we already know?"

By Rahul De & Minarva Mallik

Updated - July 31, 2026 at 04:12 PM.

Imagine hiring the world's most intelligent analyst in your team — someone who can synthesise information instantly, reason across complex topics, and communicate with remarkable clarity. Now imagine that analyst has never seen a single internal document from your company. That is precisely the limitation facing enterprises that deploy AI today.

Large Language Models are extraordinarily capable — but they are generalists. They have no knowledge of your customer research, compliance documents, or last quarter's support tickets. When asked, they either confess ignorance or generate a plausible-sounding but fabricated answer — a phenomenon known as hallucination. Retrieval-Augmented Generation, or RAG, attempts to solve this problem. It is one of the most significant architectural developments in applied AI, and yet it remains poorly understood outside specialist circles.

RAG is a method that combines two capabilities: searching a private knowledge base, and generating intelligent responses grounded in what is found. Instead of relying on what a model memorised during training, RAG first retrieves the most relevant information from your own documents — then uses that to generate an accurate, sourced answer. Think of it as giving your AI five minutes to review the relevant files before responding — every time.

How RAG Works: A Two-Phase Process

Phase 1 — Building the Index (done once). Company documents, your documents, such as interview transcripts, policy manuals, legal briefs, and support tickets are broken into meaningful segments. Each is converted into a numerical representation called an embedding vector, capturing its English language meaning. Think of it as tearing a book into topic pages and filing each in a smart cabinet that understands context, not just keywords.

Phase 2 — Answering a Question (every time). When a user gives a prompt or asks a question to the AI system, the question is converted into the same numerical format and matched against the index by meaning similarity, not keywords (which is the usual norm in other search methods). The closest segments are retrieved and handed to the AI model, which reads the source material and generates a grounded answer, citing where the information came from.

“Find the most relevant information, show it to the AI, let the AI explain it in plain English.”

RAG in practice: A product management use case

Let us now consider how RAG works in practice. The example is of a product manager working on an expense management application, and who has accumulated documents over months of research, including interview transcripts, support tickets, and app reviews, user test documents, records of outputs, etc, amounting to thousands of pages. Finding specific insights means manual searches, which requires reading through all the material, missing connections, and hours of synthesis work.

With a RAG system built on the same documents, answers can be obtained in seconds. A query for “reimbursement complaints” retrieves passages where users described “getting paid back” with semantic or meaning matching, which keyword search would miss entirely. Each response from the AI system also cites the source document and includes a confidence rating, building the traceability that a product manager needs.

RAG applications across industries

The same RAG method described above can be applied wherever organisations hold large volumes of domain-specific documents that employees must search and synthesise. Applications of RAG are emerging from different industries: In banking and financial services, compliance officers query internal policy libraries to find applicable rules instantly, reducing manual effort and compliance risk.

In healthcare, clinical teams retrieve patient notes and treatment protocols without exposing sensitive data to external AI providers. In legal services, associates find relevant precedents from a firm’s entire case history by describing the legal situation in plain English, not by searching case names. In manufacturing, engineers query thousands of pages of equipment manuals to diagnose failures and retrieve repair procedures in real time.

Enterprise RAG adoption jumped from 31 per cent to 51 per cent in 2024 alone, as quoted by Menlo Ventures, with organisations now deploying it for 30-60 per cent of their AI use cases, particularly where accuracy and proprietary data are non-negotiable. The average RAG project delivers $3.70 for every dollar invested (as observed by Menlo Ventures), and the vector database market, where these are special software that hold the number vectors created in Phase 1 of the RAG process, grew 4 four times, and are projected to reach $9.86 billion by 2030.

A growing ecosystem of providers serves this demand. ChromaDB and Qdrant suit prototyping and smaller deployments. Pinecone offers a fully managed cloud-native service for enterprise scale. Weaviate combines vector and keyword search in a single query. Milvus handles GPU-accelerated deployments at hundreds of millions of documents. The choice of provider for RAG databases, with direct consequences for cost, latency, and accuracy, deserves as much attention as the choice of AI model.

The challenges

RAG's promise is real, but prototype success does not guarantee production reliability. Three challenges consistently trip up implementations:

Scale degrades accuracy. A system that performs well on hundreds of documents begins returning contextually wrong passages at tens of thousands — and the AI model, while grounding its answers in those passages, produces confident but incorrect responses. In other words, it begins to hallucinate, the very problem RAG was introduced to solve.

Chunking and embedding quality are decisive. How documents are segmented, and which model converts them to vectors, directly determines retrieval accuracy. Poor defaults at prototype stage become expensive problems when converted to full-scale systems used everyday.

Maintenance is ongoing. In any organisation, there is a continuous stream of documents being created and accessed, and as these documents change or are added, the index must be updated. Access controls must be enforced. Response quality must be evaluated systematically, not through occasional spot-checks.

"A technology manager at a major MNC deployed RAG over 30,000 internal documents — and had to pull the product when responses became slow and unreliable. The system worked in testing. It failed at scale."

These are solvable problems, but they require deliberate planning and investment, not simply installation. Organisations that treat RAG as an engineering discipline rather than a plug-and-play product are the ones collecting its returns.

Conclusion for enterprises

RAG shifts the enterprise AI question from "which model should we use?" to "how do we connect AI models to what we already know?" The organisations that gain the most will not necessarily have the most powerful models, they will be those that most effectively ground AI in their own institutional knowledge.

RAG is not a finished product. It is a framework — one that scales from a prototype built by a single product manager to a deployment serving thousands across the, possibly global, organisation. What remains constant is the underlying principle: the AI is only as good as what you feed it. RAG ensures you feed it the right thing.

Mallik is an IIM Bangalore Alumni; De' (Retired), IIM Bangalore and Memoric AI