The R in RAG

Retrieval, Context Engineering, and How Intelligent Systems Really Work

Simon Gelinas, November 2025

01

Opening & Framing

Intelligent Systems Start With Selecting the Right Information

- Selecting information before+while reasoning = better AI
- Retrieval underpins search, recsys, RAG, agents
- Needed wherever data grows
- Today’s talk: technical + product + real-world tradeoffs

My Background

- Leading the AI search and recommendation platform at Upwork
  (Human + AI agent collaborative work)
- Head of engineering at a Objective
  (AI search startup, acquired by Upwork)
- Head of data science at Flourish, also founding engineer there
  (FinTech startup, acquired MassMutual)

Why Retrieval Is Central To Modern AI

- Contextualization / personalization at scale and fast
- Agents need tool routing
- Permissioned datasets
- Models grow slower than data (hardware limits)
- Cheap reasoning requires pruning
- Hallucinations

02

Retrieval as a Universal Pattern

Retrieval Is Everywhere

- Search
- Recommendations
- RAG
- Agent tool selection
- Memory & planning systems

Most Intelligent Systems Follow the Same Retrieval Funnel

- Candidate Generation & Filtering
- Reranking
- Expensive LLM Reasoning
- Shared data pipeline, different surfaces
- Tools for optimization and experimentation

Bigger Models Don’t Remove the Need for Retrieval

- Context limited
- Compute expensive
- Data grows infinitely
- Permissions & security
- Selectivity > brute force

03

Retrieval Funnel (Technical Core)

The funnel spans multiple layers

- Layers, from First → Last:
  - Speed: Fast → Slow
  - Data: Lots → Little
  - Compute per result: Light → Heavy
- Optimization: types of layers, parameters, structure (graph)

Candidate Generation (The Wide Net)

- Fast, coarse, high recall
- Millions → hundreds
- Continuum of inputs:

  keyword → sentence → paragraph → page
- Needed even in agent selection

Retrieval Signals: Lexical, Semantic, Structured

- Lexical (BM25) = rare terms, debuggable
- Dense embeddings = semantic similarity (multimodal)
- Structured signals (numbers, metadata)
- Numerical features often need English encoding for LLMs
- Hybrid search = combining them

Indexing and Filtering Data

- Indexing brings data in your retrieval system
- Data is used for relevance and filtering
- Filtering usually within candidate generation
- Faceting counts elements within a filter

Hybrid Retrieval Is the Industry Standard for Reliability

- Combine lexical + dense + numerical
- Improves recall & precision
- Robust to long-tail queries
- Industry-standard design

Reranking

- Gradually increasing model complexity
- XGBoost is a good baseline
- Parallelizable/faster models for larger candidate pools
- Late stage interactions

Heavy Reasoning Layer(s)

- Cross encoders
- LLM rerankers
- Deduplication & conflict resolution
- Summaries & compression
- Final synthesis

Retrieval in Agentic Flows

- MCP context bloat
- Skills = pre-built “retrieval-first” programs
- Constraints agent behavior
- Retrieval becomes decision routing

Why the Funnel Matters

- Latency & cost
- Human perception of relevance
- Data engineering alignment
- Funnel determines what LLM can “think about”

04

Context Engineering

More Context Doesn’t Equal Better Answers

- Order matters
- Redundancy hurts
- Selective inclusion > brute force
- Large context ≠ better reasoning

Good Systems Differentiate Data for Retrieval From Context for LLMs

- Chunking = retrievable units
- Context assembly = LLM-readable units
- Document processing / parsing (at indexing)
- Document → summary + chunks + multimodal

Context Engineering Is About Prioritizing What Truly Matters

- Summaries
- Routing models
- Optimized for llm consumption
- Multi-step retrieval

05

Retrieval Evaluation + Tradeoffs

Evaluating Retrieval (Reality-Based)

- Recall@k
- NDCG
- Diversity & coverage
- Structured + semantic relevance

Real Challenges in Evaluation

- Historical data bias
- Sometimes <1% of items ever shown
- Cold start across docs/users/queries
- Domain drift

LLM-as-a-Judge Helps Cold Start But Isn’t Ground Truth

- Great for cold start
- Cheap and scalable
- Biased and inconsistent
- Can do absolute and relative judgments

When to Retrieve vs When to Fine-Tune

- Stable knowledge → fine-tune
- Dynamic/permissioned/private → retrieve
- Heavy domain drift → retrieve
- Long-tail queries → hybrid

06

Real-world experiences

Where LLM-facing retrieval diverges from Human-facing one

- Latency
- Perceived relevance
- Information volume supported I
- Prompt injection risks

Thinking through Tradeoffs

- Relevance vs permissions
- Speed vs accuracy
- Retrieval only vs hydration
- Stakeholder disagreement

Issues and Challenges

- Client ≠ end-users
- Conflicting definitions of “relevant”
- The right answer must also “look” right
- Bottlenecks, data quality, data freshness, dependencies
- Liability & filtering issues

Stories of User Seeing Results

- Align system to human expectations
- Data engineering prevents “quiet failures”
- Retrieval shapes perceived intelligence, the UI/UX too
- Relevance is subjective

07

Closing

Key Takeaway: Retrieval Shapes Intelligence in Many Systems

- Retrieval is universal, it’s a way to prioritize relevance
- Funnel = scalable intelligence
- Data for retrieval ≠ data in context window
- Retrieval vs fine-tuning depends on constraints
- The right system is very contextual, hard to generalize

Q&A / Discussion

- Design choices
- Tradeoffs
- Product implications
- Career & system-design questions