

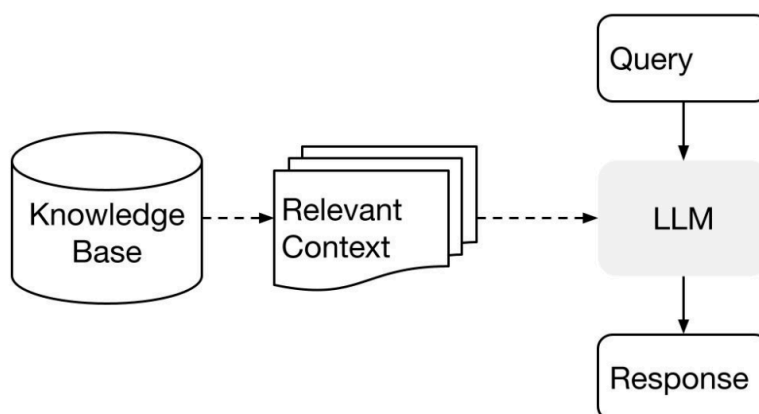
RAG FUNDAMENTALS 01

Who am I

- CEO of <https://hypedigitaly.ai>
- almost 3 years of experience building RAG systems
- over 30 clients in enterprise and governmental niche
- RAG accuracy over 95% in live deployments

What is RAG (Retrieval Augmented Generation)

- Asking AI about information it's not trained on
 - LLMs have knowledge cut-off
 - The AI doesn't know your data unless you give it access
- **Why it matters:**
 - current / up-to-date information
 - no hallucinations
 - context window limitations of LLMs
 - GPT-5.1: 400k tokens
 - Claude 4.5 Sonnet/Opus: 1M tokens
 - Enterprise data: 1 QUINTILLION tokens!
- **Components:**
 - Your knowledge base
 - Search/retrieval
 - LLM generation



- **INGESTION PIPELINE** → Load → Chunk → Store in vector DB / KB
- **RETRIEVAL PIPELINE** → Query → Search & Retrieve → Generate response with LLM

Main problems with RAG & Gen AI projects (95% of RAG projects fail in production)

Problem #1 - Data structure

- **How to upload company data** (dozens of thousands of unstructured URLs and documents into a knowledge base in structured format) ?
 - **Problem A:** data dumped into the KB without any preprocessing / proper structure
 - **Problem B:** KB cannot return valid information in order to answer the user query with precision and without hallucinations
 - **Problem C:** Data needs to have high quality and should consist of unique information (removing unnecessary boilerplate)

Problem #2 - Chunking

- In ideal world each single and whole document, or URL's content - would each sit in its own fit chunk with cleaned and well structured information
- **Vector stores have token limits:**
 - allowed max chunk size from 500 up to 4096 tokens for most vector stores
- **Most platforms use naive chunking strategies**
 - Divide content by fixed token counts
 - Allow around 800-1500 tokens for single chunk
 - Text cut in the middle of sentences
 - Context lost, retrieval fails
 - Fixed token overlap
- Ideally you could choose between different ways of chunking
 - Each coherent section of the URL/document that belong together logically, would sit in their own chunks
 - Metadata -> big picture idea (page number, url link, summary how the chunk before ended, current chunk summary, source page summary, etc.)

Problem #3 - Metadata

- Each chunk should have rich and detailed metadata
 - Origin (document filename / page number, url link)
 - Fields for filtering
 - Current chunk summary
 - Chunk overlap summary

Problem #4 - Automated updates

- KB needs to be updated in regular intervals
- Built once, never updated
 - AI gives old information, or hallucinates

RAG FUNDAMENTALS 02

Bad / poor chunk

None

*"Sign up Sign in Share The AI industry faces a significant challenge with 95% project failing, as an MIT study revealed. This indicates a shift in the market where concrete results are now paramount, moving beyond the previous "build-it-and-they-will-come" mentality, a point also discussed on the All-In Podcast. As David Sacks famously said on the All-In Podcast, "AI is a feature, not a product." Press enter or click to view image in full size ![Image 3](https://miro.medium.com/v2/resize:fit:700/0*Hzam2oEP-BaosX0t) This reality check is validated by a study from MIT's Project NANDA initiative, titled "The GenAI Divide: State of AI in Business 2025." The methodology used in this research involved surveying over 500 enterprises across various industries and conducting in-depth interviews with IT leaders and business executives and found a the following reality: An astonishing 95% of enterprise generative AI projects fail to deliver meaningful business impact or revenue acceleration. The report found that despite between \$30 billion and \$40 billion in enterprise invest"*

Ideal / rich chunk

```
{
  "id": "art_nati_fail_05",
  "search_payload": {
    "text_content": "# Pragmatic Approach: A Hybrid AI Approach with a Deterministic Substrate\n\nInstead of \u201cthrowing\u201d a large language model (LLM) at a business problem, a far more effective strategy is to use a **Hybrid AI** approach with a **\u201cDeterministic Substrate\u201d** pattern. This model flips the script, making the generative AI a powerful, yet controlled, co-pilot rather than the sole pilot.\n\nIn this model, the **deterministic layer** (your existing, reliable, and auditable automation framework) acts as the core foundation. It contains all the business logic, security policies, and known workflows. The **generative AI** is then layered on top, serving as a powerful assistant that provides insights or enriches data.\n\nIts outputs are always funneled back through the deterministic layer for validation and execution, ensuring every action is predictable, safe, and auditable.\n\nThis pattern is a direct countermeasure to the key problems identified by the MIT research, providing a clear path to production and a tangible ROI.",
    "chunk_header": "Title: 95% of AI Projects Fail | Section: Pragmatic Approach",
    "keywords": [
      "Hybrid AI",
      "Deterministic Substrate",
      "ROI",
      "Automation Framework",
      "MIT Research"
    ]
  },
  "filter_metadata": {
    "published_date_int": 20250825,
    "author_id": "nati_shalom",
    "source_domain": "medium.com",
    "doc_type": "article",
    "source_page_summary": "Analysis of MIT study showing 95% AI project failure rate. Arguments against top-down enterprise AI in favor of empowering 'shadow AI' power users. Proposes a technical solution: 'Hybrid AI' using a 'Deterministic Substrate' (automation layer) to validate GenAI outputs.",
    "chunk_overlap_summary": "Transition from 'Employee-Enterprise Disconnect' to 'Hybrid AI' solution. Connects the bottom-up adoption strategy to the technical implementation of deterministic substrates.",
    "chunk_index": 5,
    "prev_chunk_id": "art_nati_fail_04",
    "next_chunk_id": "art_nati_fail_06",
    "source_url": "https://medium.com/@natishalom/95-of-ai-projects-fail-heres-why-that-s-a-good-thing-8a5936ebdea8"
  }
}
```

Feature / Aspect	BAD CHUNK (The "Naive" Approach)	GOOD CHUNK (The "Production" Approach)
1. JSON Structure	Flat & Messy. simple <code>text</code> field mixed with minimal <code>metadata</code> . Hard to separate signal from noise.	Hierarchical. Strict separation between <code>search_payload</code> (what we embed) and <code>filter_metadata</code> (what we filter).
2. Data Hygiene	Polluted. Contains UI noise, "sign up" links, image alt tags, and navigation clutter.	Clean. Pure information signal only. All non-content elements are stripped out.
3. Semantic Integrity	Fragmented. Often cuts off mid-sentence or mid-thought due to arbitrary token limits.	Complete. Captures a self-contained logical concept or paragraph from start to finish.
4. Structure & Tone	Flat. A single block of plain text. The AI can't tell what's a header or what's emphasized.	Structured. Preserves Markdown (headers, bolding, lists) to guide the AI on what is important.
5. Filtering Power	Weak. Relies on fuzzy strings (e.g., <code>"date": "August 25"</code>). Slow and imprecise.	Strong. Uses precise scalars (e.g., <code>"date_int": 20250825</code>) for instant, O(1) database filtering.
6. Contextual "Glue"	Isolated. The chunk has no awareness of the broader article or what came before it.	Connected. Includes metadata with a "global summary" and "overlap text" to maintain flow.