RAG

Retrieval-Augmented Generation

RAG in Action



Service Bots

Support grounded in real-time internal docs.



GitHub Copilot

Context-aware code completion for large repos.



Amazon Rufus

Analyzing vast review data for product advice.



Video Search

Deep semantic search within transcripts.

Zero-Training AI

0

Fine-tuning Required

Retrieve, Don't Train

RAG avoids the heavy cost of model training. It retrieves relevant **document chunks** at query time and injects them into the context.

This creates a system that is 100% grounded in your current data, essentially eliminating the risk of hallucinations.

NotebookLM



- **Zero Hallucination:** By restricting information to your specific sources, the model remains highly accurate.
- The model is **not trained** on your documents; it treats them as a real-time knowledge source.

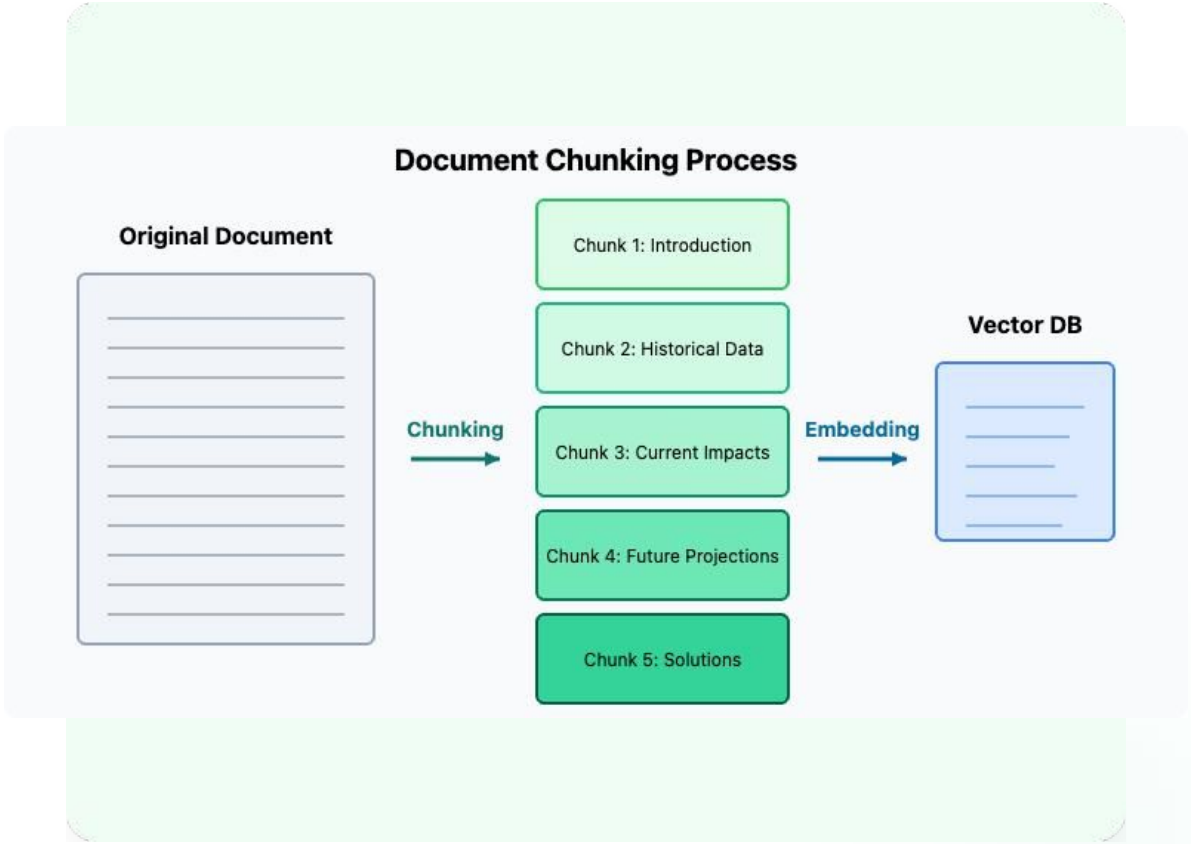
  It retrieves relevant **chunks at query time** and injects them directly into the context window.
- The LLM generates an answer **grounded** in those specific chunks, ensuring every claim is backed by the text.

Preprocessing & Chunking

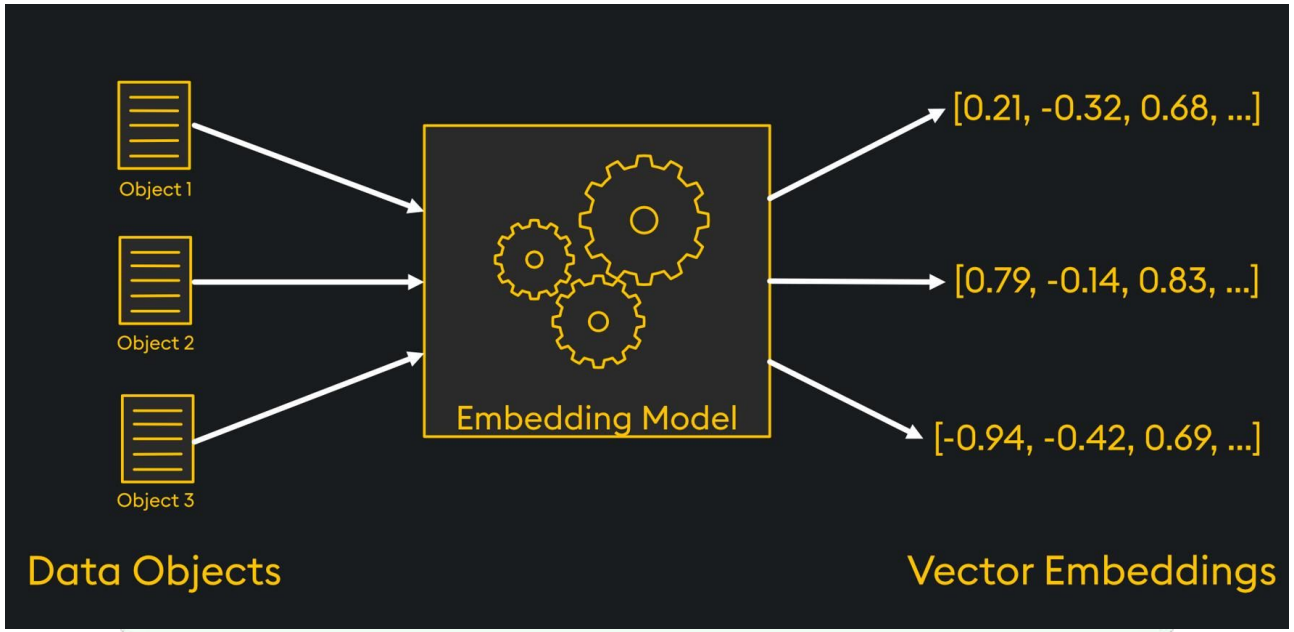
Breaking Down Complexity

Large documents cannot be sent to an LLM whole. We must split them into logical **chunks**

- LangChain for creating chunks
- Context Window Optimization
- Semantic Continuity



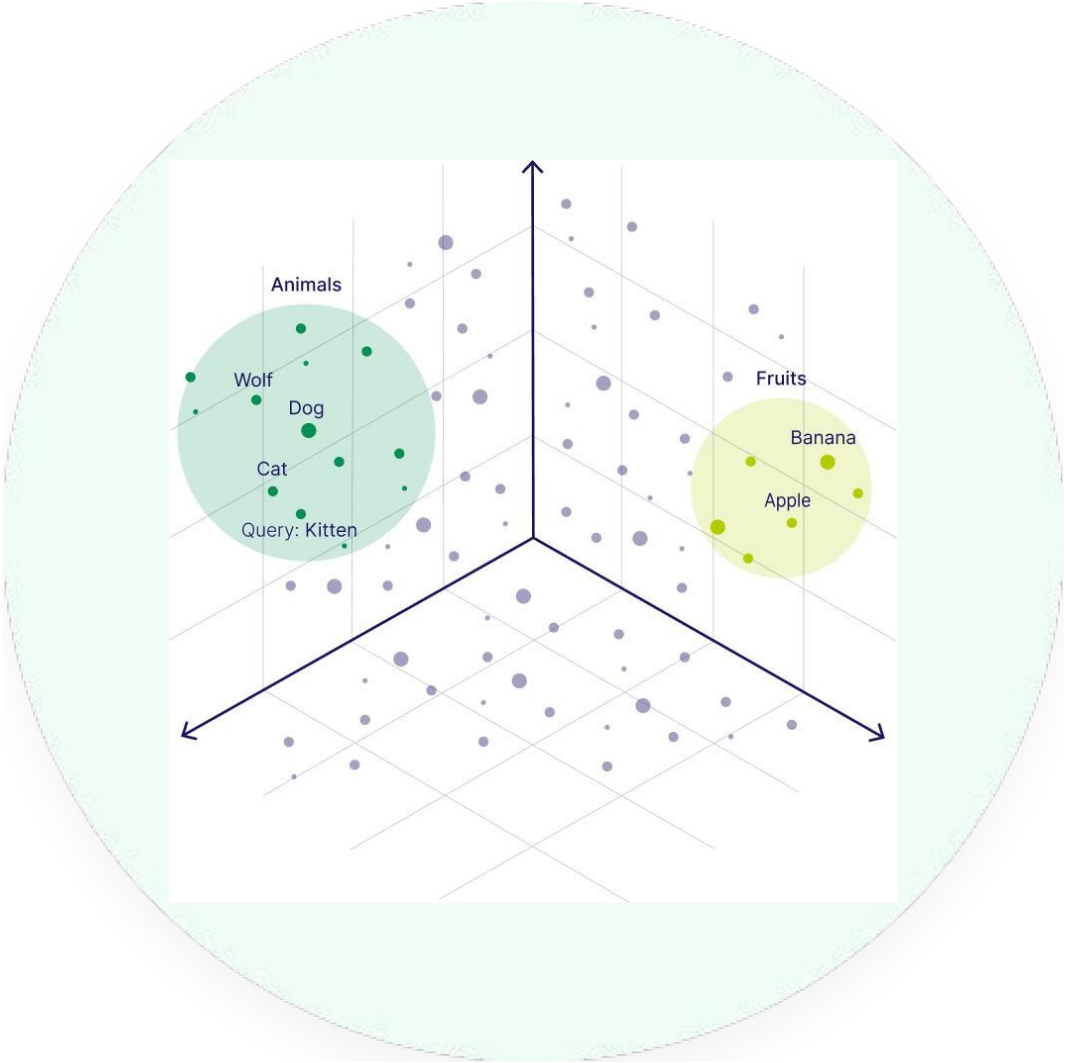
Vectorization



Semantic Math

Words with similar meanings (e.g. **"cat"** and **"kitten"**) sit closer together in mathematical vector space. All vectors then stored in a database.

R: Retrieval



Cosine Similarity

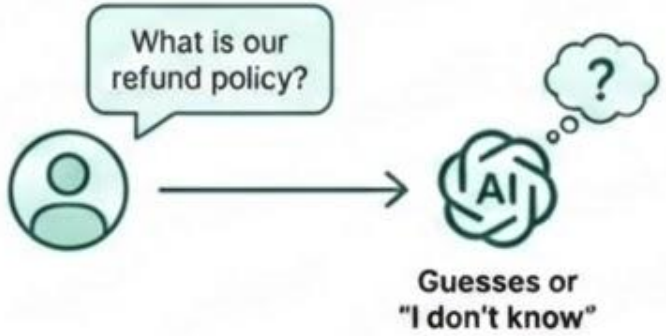
When a user asks a question, we vectorize it and perform a **K-Nearest Neighbor** search against the Vector DB.

We retrieve the most mathematically similar **chunks** to provide the LLM with the perfect "cheat sheet" to answer the user's question.

A: Augmentation

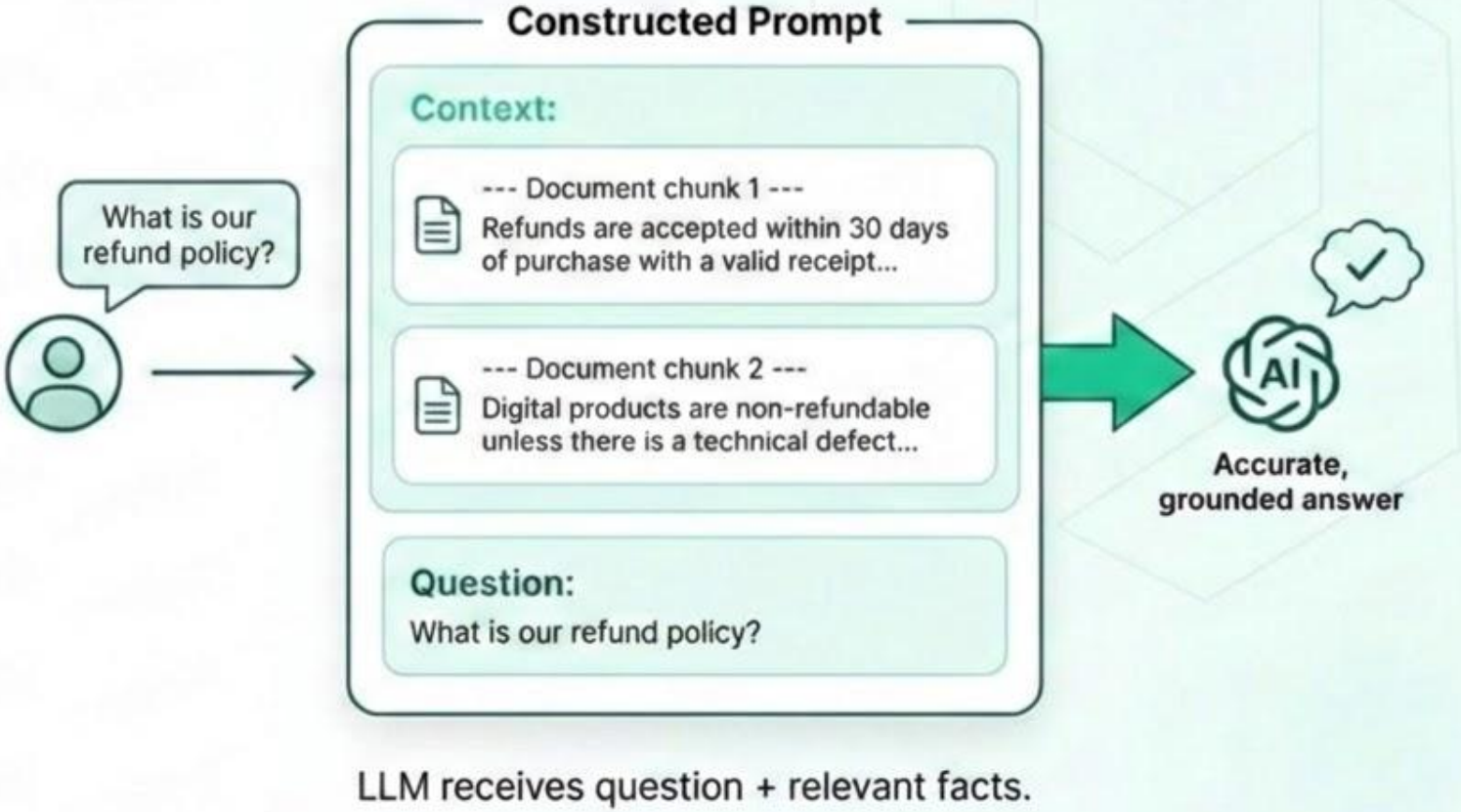
Injecting Context

Without Augmentation



LLM receives bare question.

With Augmentation



LLM receives question + relevant facts.

G: Generation

- **Synthesis:** The LLM drafts a response using only the provided facts
- **Verifiability:** Every answer can be tied back to a specific document page or chunk.
- **Security:** By grounding answers in internal data, we protect against prompt injection and errors.

Limitations



Data Quality

The system is only as good as the underlying document source.



Logic Gaps

Difficulties in relating distant chunks of information.



Cost

Initial vectorization can require large expensive databases.



Ambiguity

Can miss the mark on vague or poorly phrased user questions.