LLMs RAG & Graph Knowledge

Mohaddeseh Tabrizian

Engineering with GenAI
09.12.2025

Agenda

- Introduction and Overview RAG
- Information Retrieval and Search Foundations
- Information Retrieval with Vector Databases
- LLMs and Text Generation
- Knowledge Graphs

Introduction and Overview RAG

Understanding RAG

LLMs are powerful

- Summarize text
- Generate code
- Rewrite content

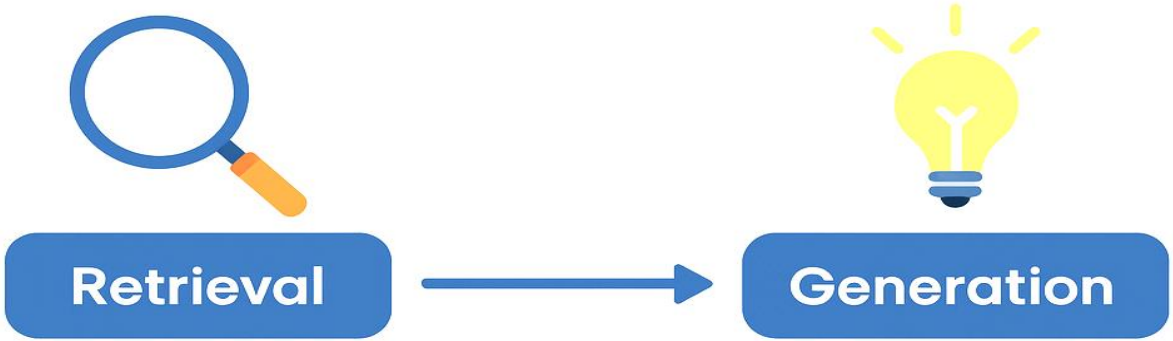
but RAG further improves them

LLM + New information

Two Steps for Answering Questions

Why Kaisers-lautern is so quiet on weekends?

Why Kaisers-lautern is so crowded today?



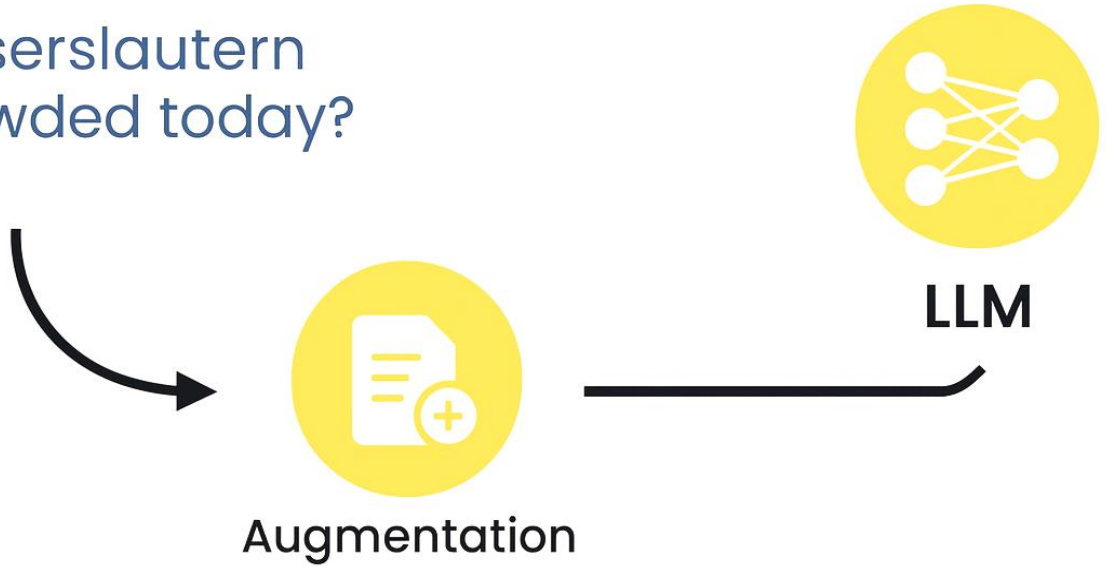
Just put it in the prompt

User prompt

Why Kaiserslautem is so crowded today?

User prompt

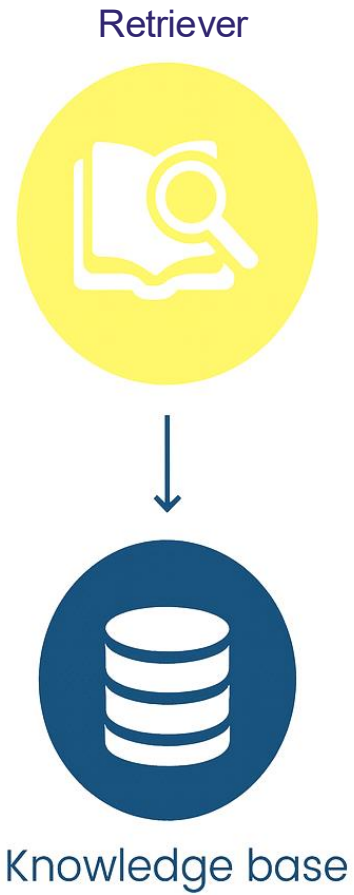
Why Kaiserslautern
is so crowded today?



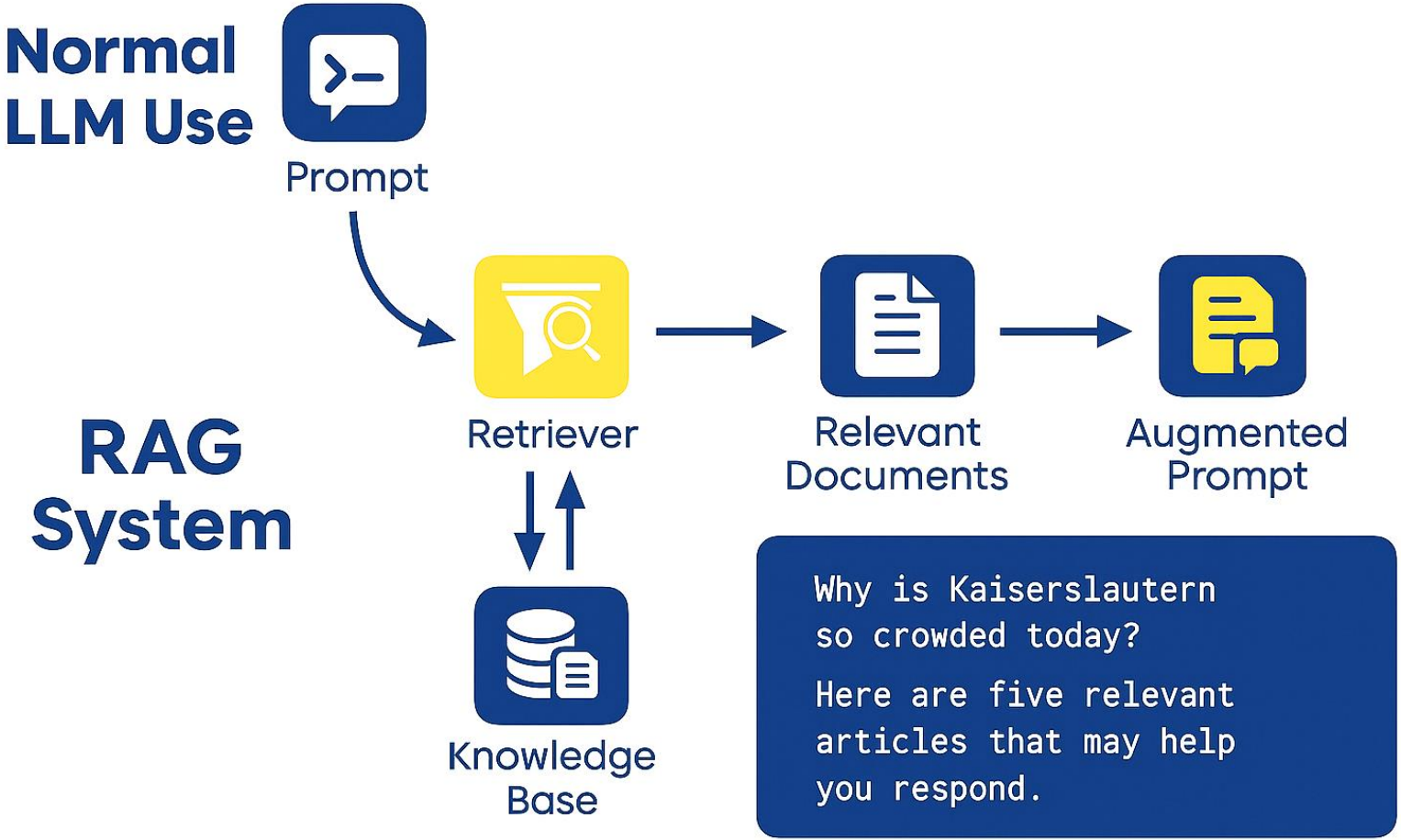
FCK has a game with
Dortmund today.

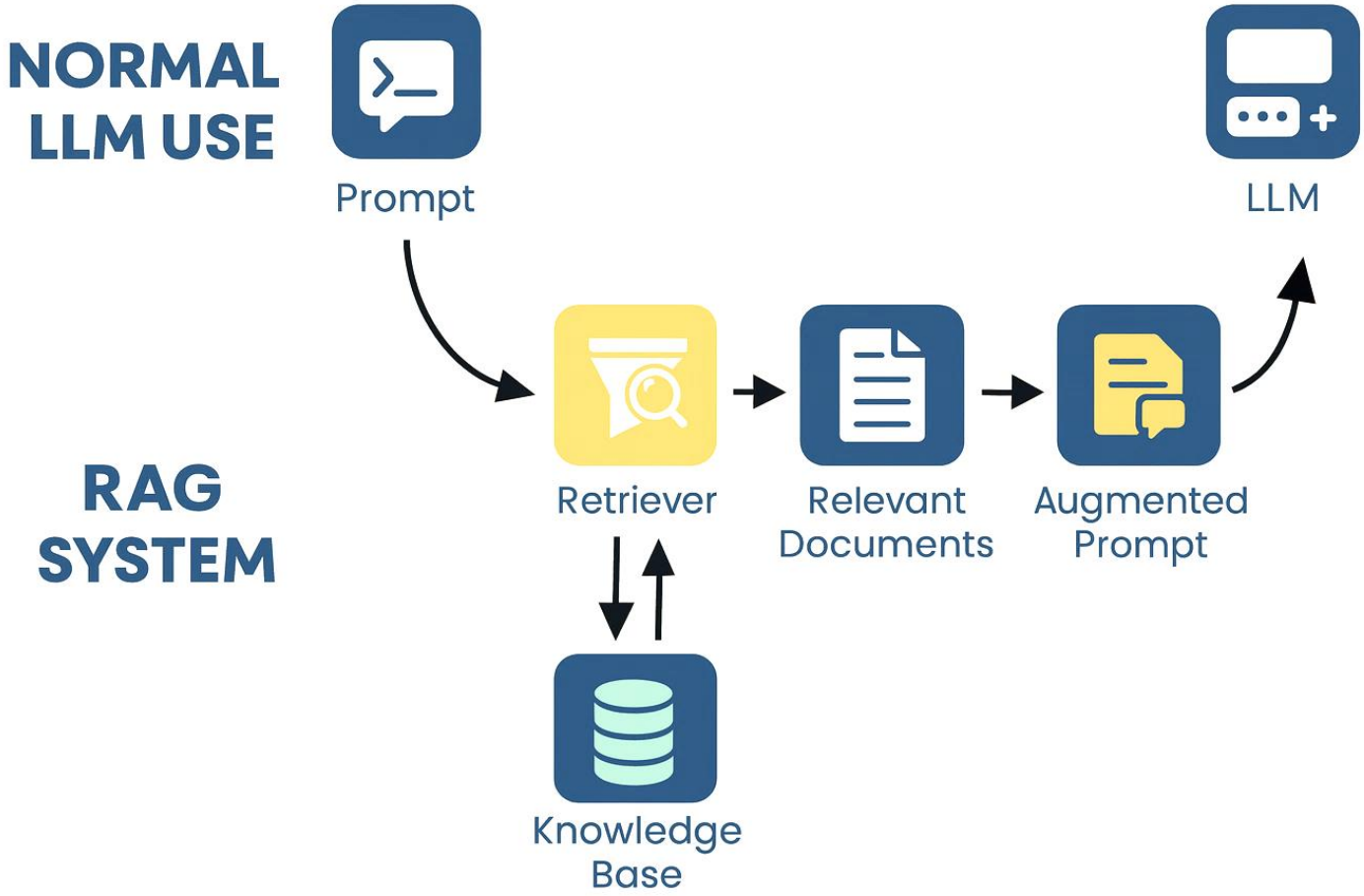
Retriever

- Manages knowledge base of trusted information
- Finds the most relevant information and shares information with the LLM
- Improves generation



Retreival Augmented Generation


Normal
LLM Use
Prompt
RAG
System
Retriever
Relevant
Documents
Augmented
Prompt
Knowledge
Base
Why is Kaiserslautern so crowded today?
Here are five relevant articles that may help you respond.


NORMAL
LLM USE
Prompt
LLM
RAG
SYSTEM
Retriever
Relevant
Documents
Augmented
Prompt
Knowledge
Base

Why LLMs Hallucinate

LLMs generate probable word sequences

LLMs produce word sequences based on training patterns

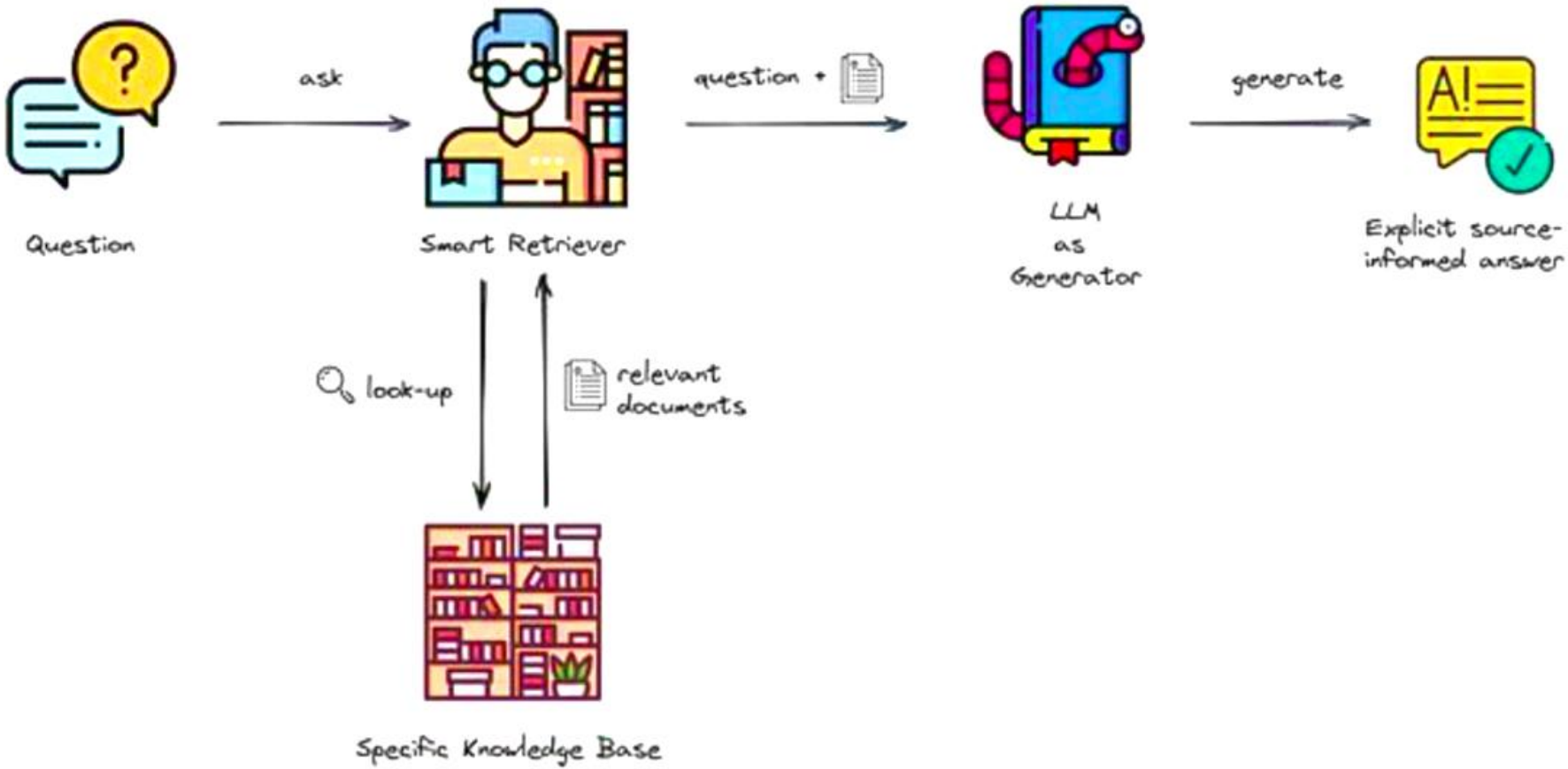
Knowledge gaps cause inaccurate responses

Responses might seem correct but are not

Truthful ≠ probable

LLMs are not designed to generate ‘‘truthful” text

How RAG solves the problem?



Information Retrieval and Search Foundations

Two Search Approaches

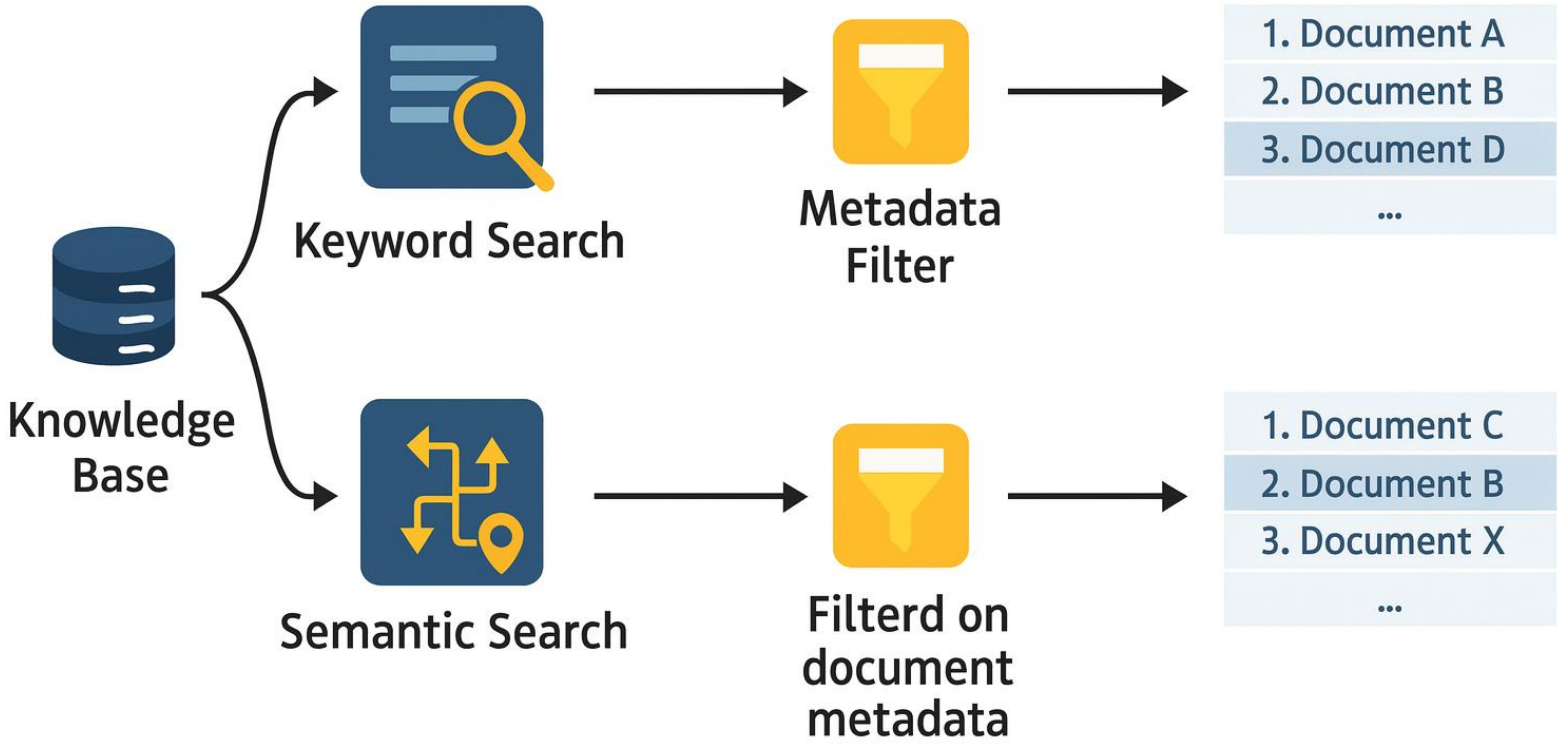
Keyword Search

Looks for documents containing the **exact words** found in the prompt.

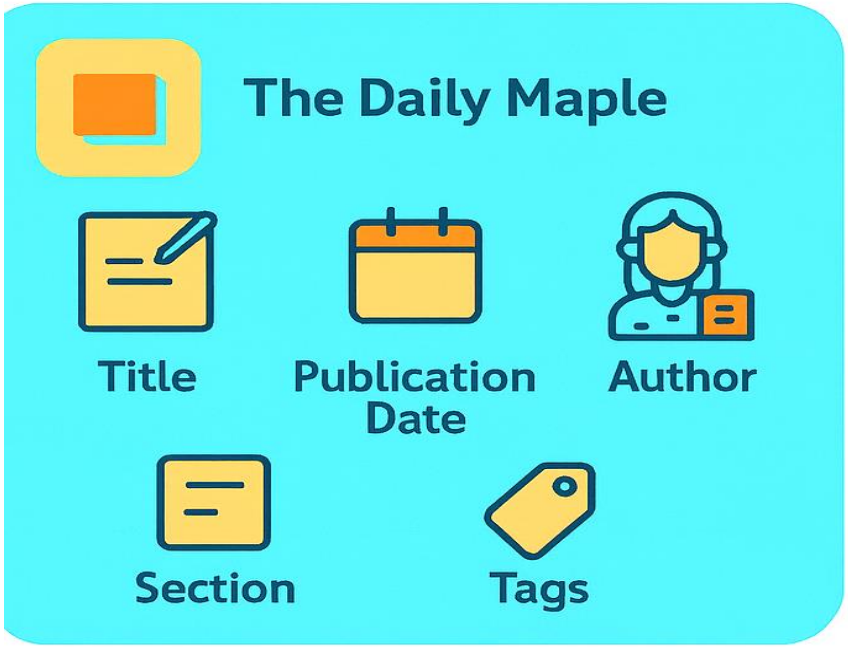
Semantic Search

Looks for documents with **similar meaning** to the prompt.

Search Techniques



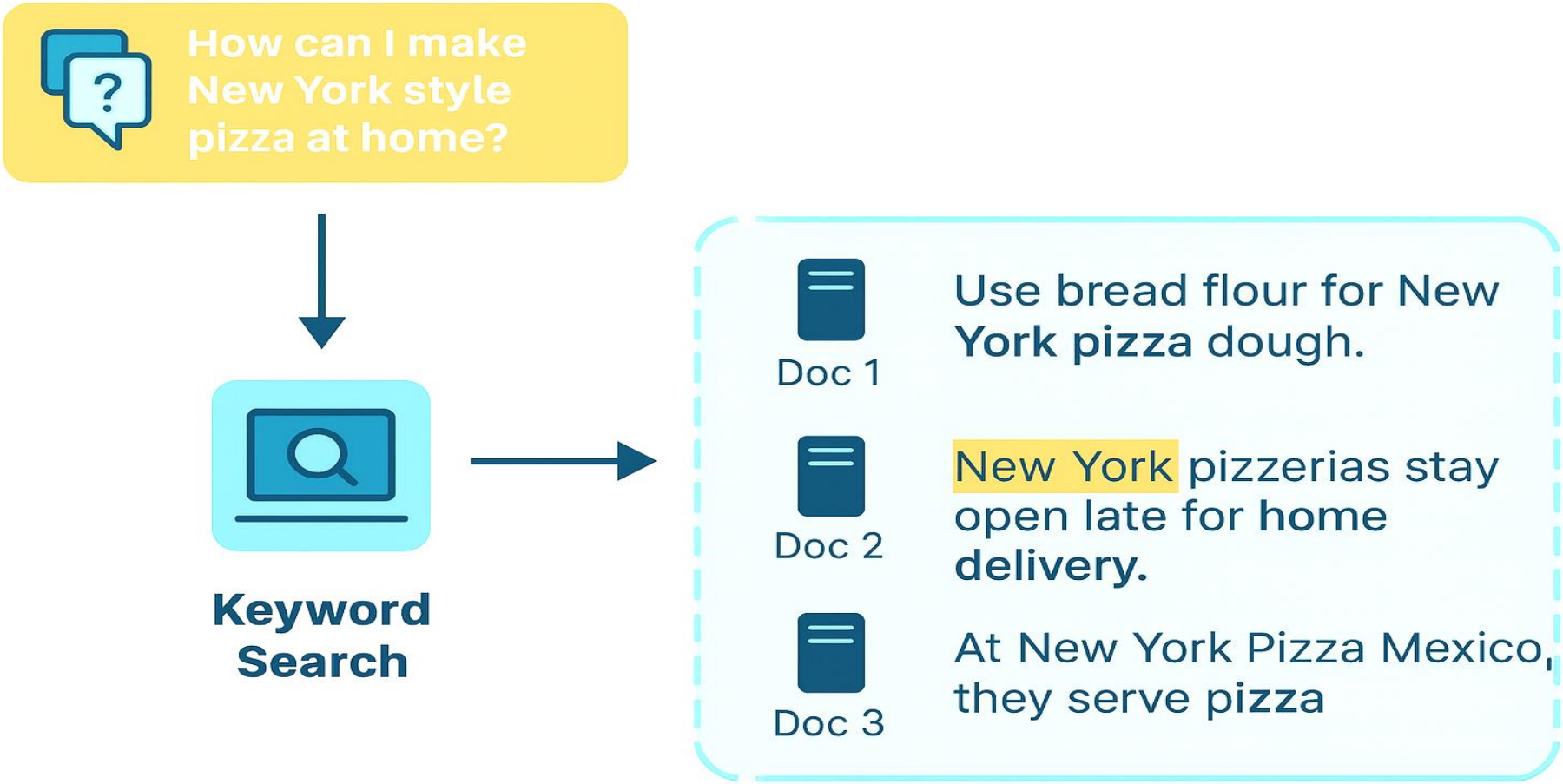
Metadata Filtering




```
SELECT * FROM articles
WHERE publication_date = '2023-10-01;'
```

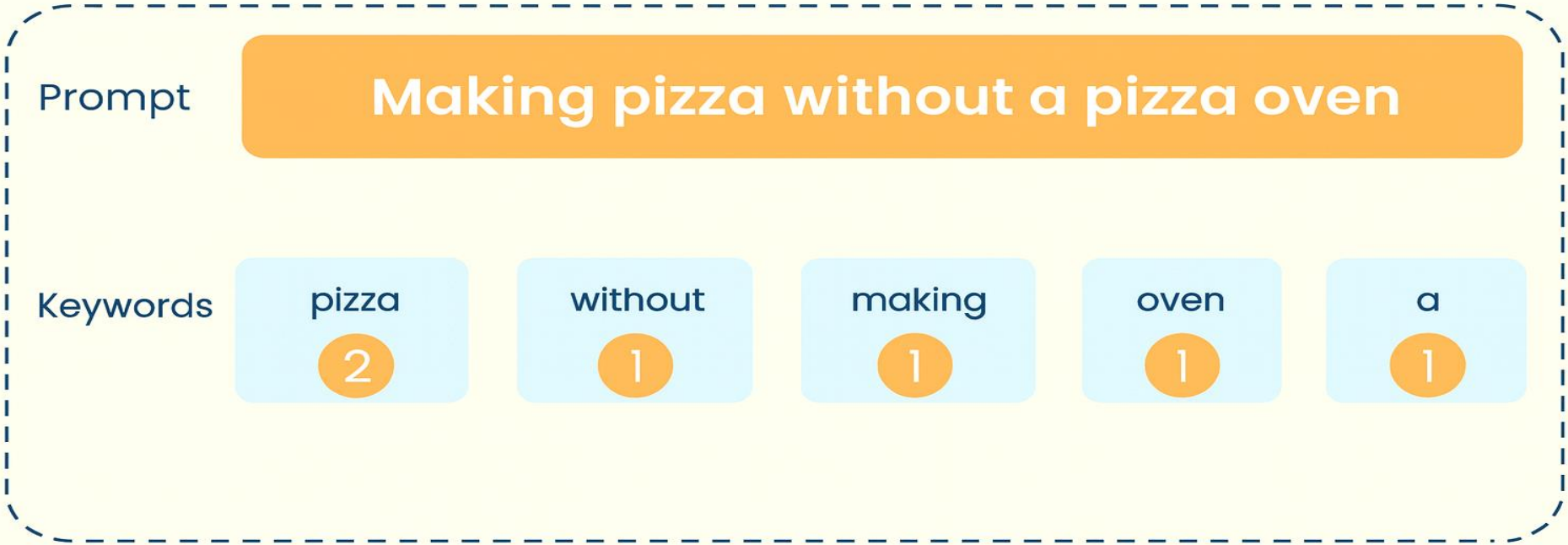
Return all articles published on specific

```
section = "Opinion"
author = "Michael Chen"
date = June to July 2024
```


How can I make New York style pizza at home?
Keyword Search
Doc 1
Use bread flour for New York pizza dough.
Doc 2
New York pizzerias stay open late for home delivery.
Doc 3
At New York Pizza Mexico, they serve pizza

Bag of Words

Word order is ignored, only word presence and frequency matter



Bag of words

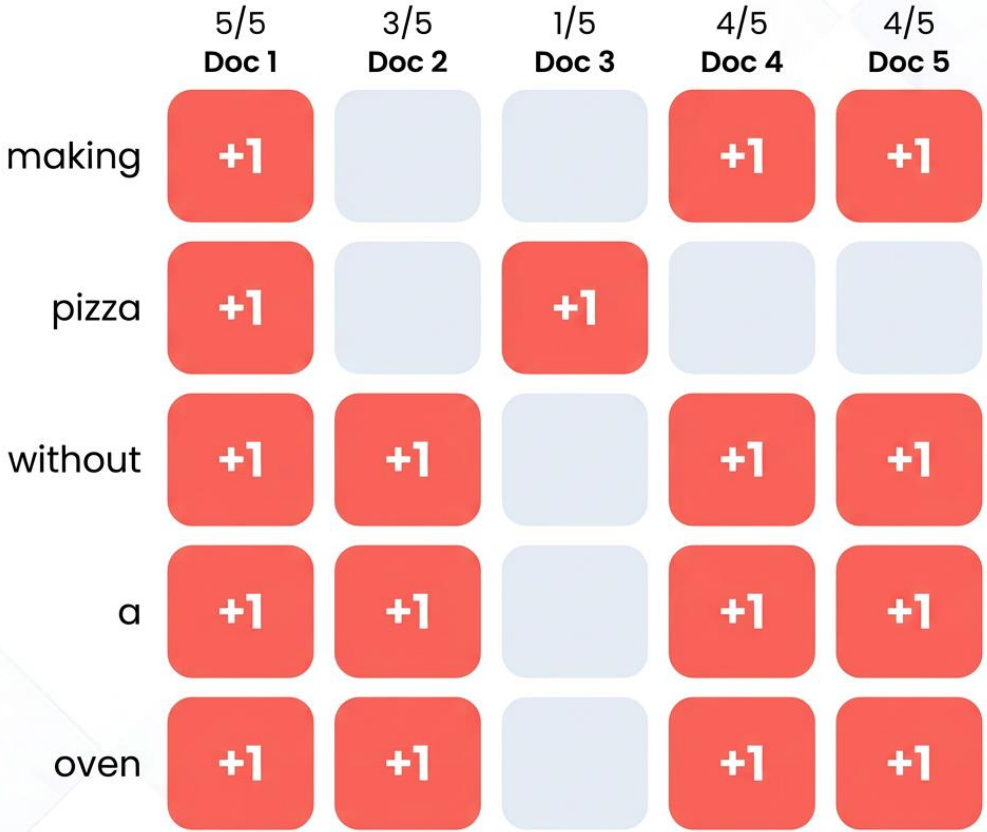
Sparse Vectors

Making pizza without a pizza oven

making	pizza		without	drink		
1	1		1	0		
pan	**burger**	**taco**	**with**	**salad**	**tea**	
0	0	0	0	0	0	

Most words aren't used. The bag of words is sparse, with few non-zero entries.

MAKING PIZZA WITHOUT A PIZZA OVEN: DOCUMENT ANALYSIS



https://www.coursera.org/learn/retrieval-augmented-generation-rag/home/module/1

Frequency Based Scoring

Document 1

Homemade pizza in oven is better than frozen pizza

Contains: Pizza (2x) Oven (1x)

Simple Scoring = 2 points

TF Scoring = 4 points

Document 2

Wood-fired oven is a better oven than a stone oven for pizza

Contains: Pizza (1x) Oven (3x)

Simple Scoring = 2 points

TF Scoring = 4 points

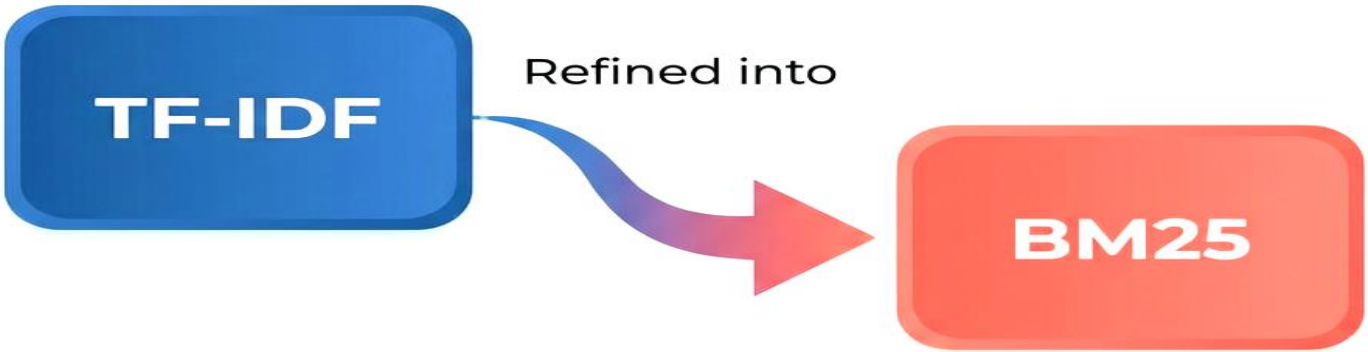
TF–IDF

PROBLEM	SOLUTION	FORMULA
Basic TF scoring treats all words equally, whether they're common filler words or rare, meaningful terms. 	Weight terms using 'inverse document frequency' (IDF). 	**Score** = TF(word, doc) × log(Total docs / Docs containing word) 

TF-IDF: Document Frequency Analysis

COUNT DOCUMENTS (DF)	FLIP TO REWARD RARE WORDS (IDF)	APPLY LOG (LOG IDF)
	 High Score	 Still higher
Pizza: Appears in 5 out of 100 documents.	**IDF = 1 / 0.05**	**log(20)**
DF = 5/100 **0.05**	**20**	**1.30**
T	T	T
The: Appears in all 100 documents.	Low Score	Too common, no weight
DF = 100/100	**IDF = 1 / 1.0**	**log(1)**
1.0	**1.0**	**0**

Evolution of Information Retrieval Models

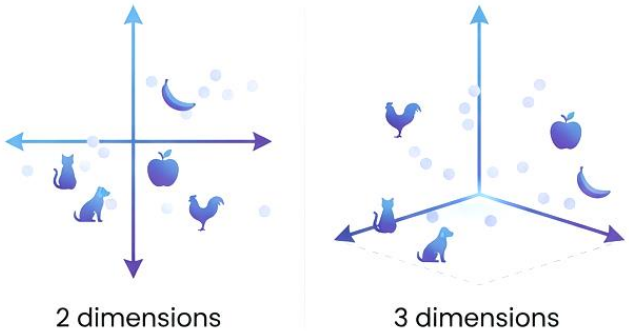


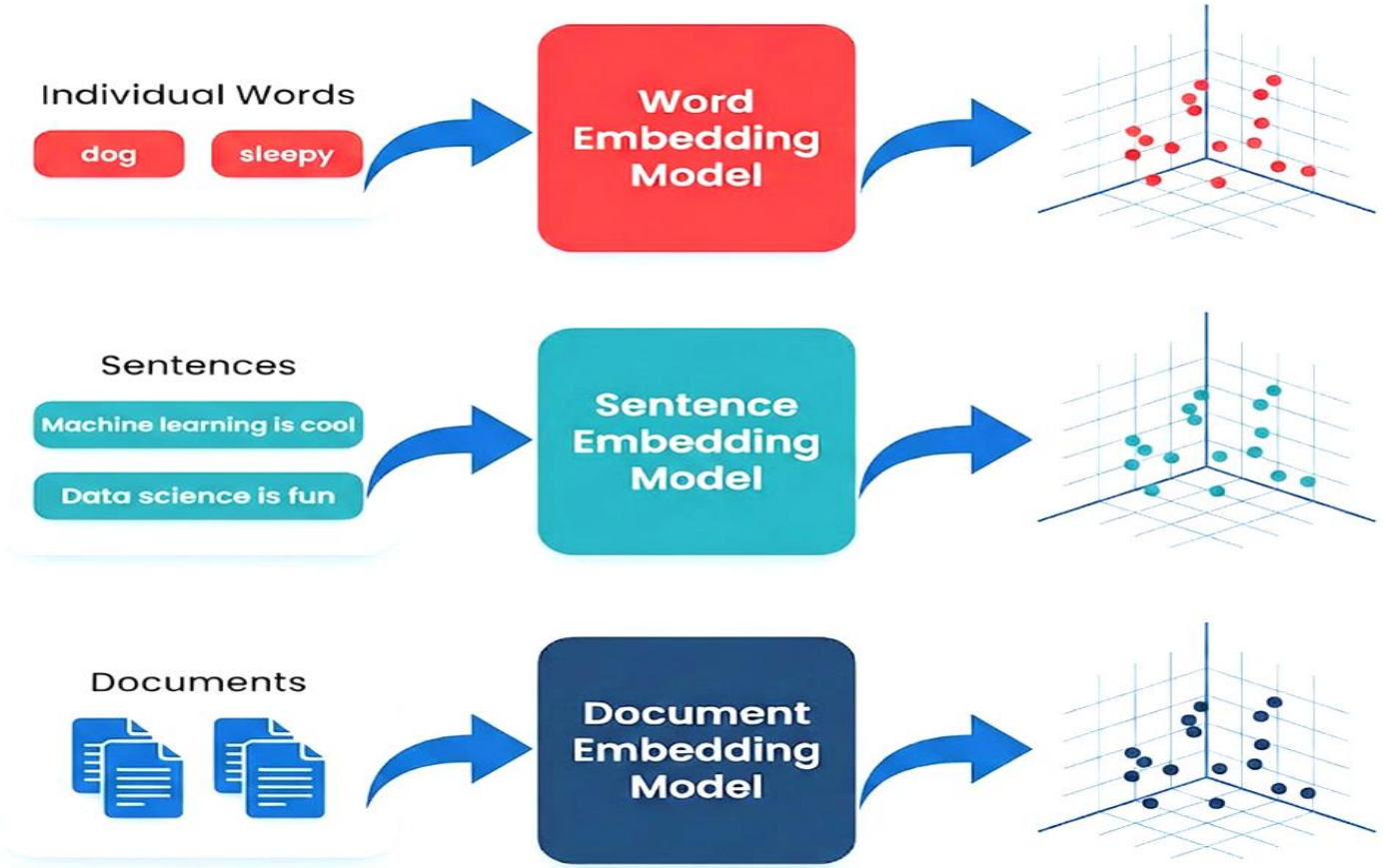
Modern systems use a slightly refined version called BM25

Semantic Search vs. Keyword Search

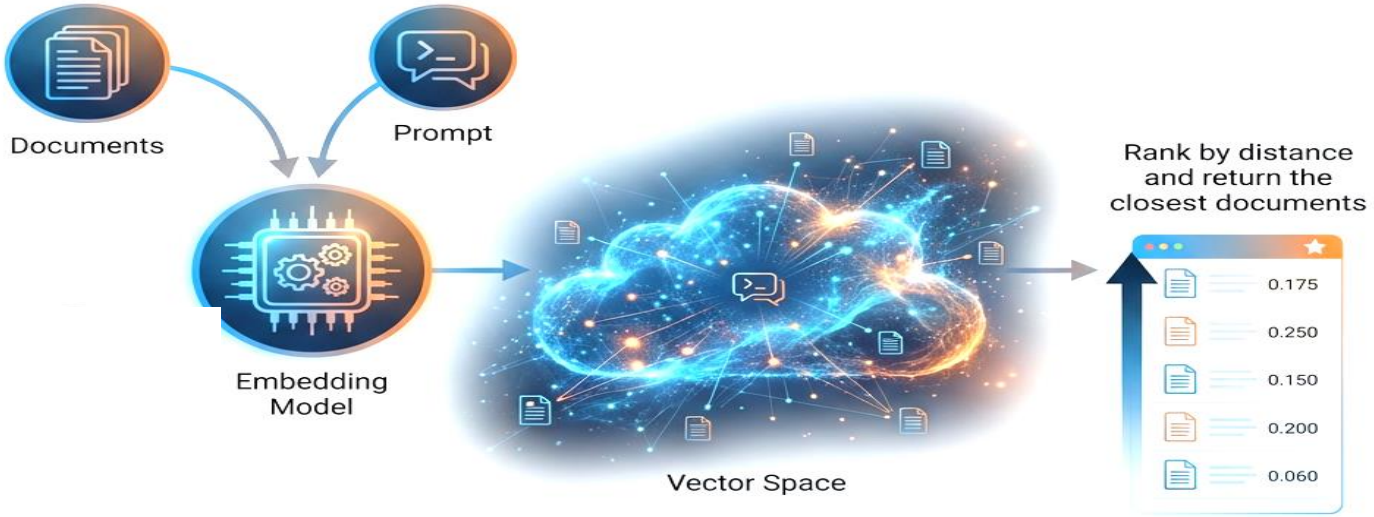
- Prompt and documents each get a vector
- Vectors compared to generate scores
- The main difference is how vectors are assigned
  - **Keyword Search:** count words
  - **Semantic Search:** use embedding model

Understanding Embedding Models



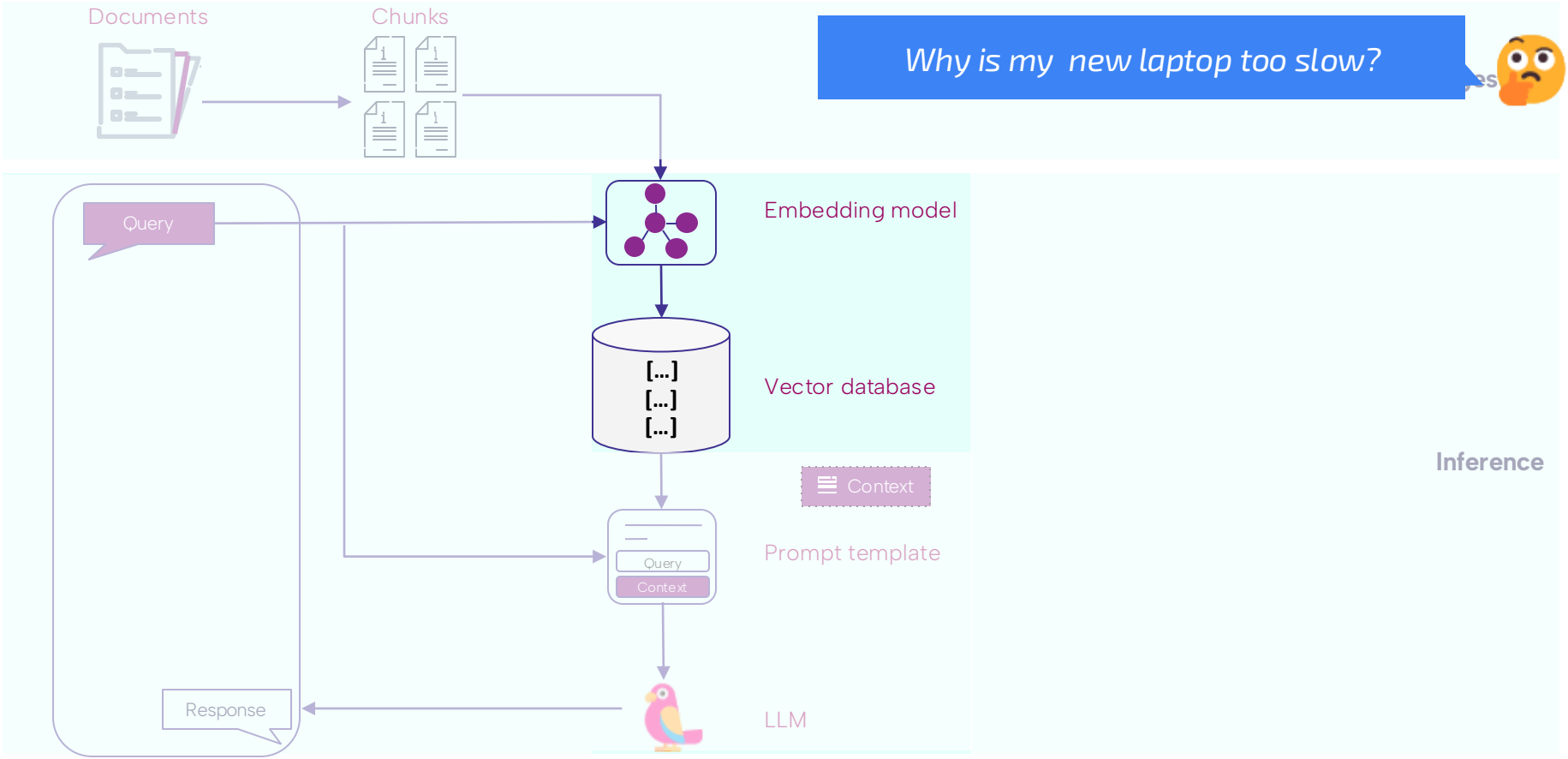

Individual Words
dog
sleepy
Word Embedding Model
Sentences
Machine learning is cool
Data science is fun
Sentence Embedding Model
Documents
Document Embedding Model

Semantic Search



Information Retrieval with Vector Databases

How does RAG work? | VectorDBs



https://weaviate.io/blog/introduction-to-rag

Vector Databases

The Vector DB: An Overkill Solution

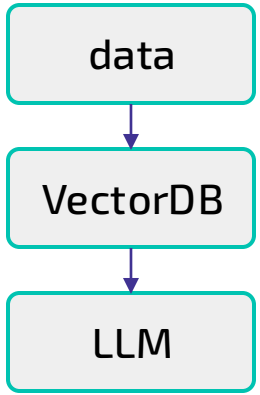
Auto-GPT relies on vector databases for faster k-nearest neighbor (kNN) searches. These databases retrieve earlier chains of thoughts, incorporating them into the current query context for GPT to provide a memory effect. However, given Auto-GPT's constraints and limitations, this approach has been criticized as excessive and unnecessarily resource-intensive.

The main argument against using vector databases stems from the cost constraints associated with Auto-GPT's chain of thoughts. A 50-step chain of thoughts would cost 14.4 dollars, while a 1000-step chain would cost significantly more. As a result, the memory size or length of the chain of thought rarely exceeds four digits. In such cases, an exhaustive search (i.e., the dot product between a 256-dim vector and a 10,000 x 256 matrix) for the nearest neighbor proves to be sufficiently efficient, taking less than one second to complete. In contrast, each GPT-4 call takes approximately 10 seconds to process, making the system GPT-bound rather than database-bound.

https://jina.ai/news/auto-gpt-unmasked-hype-hard-truths-production-pitfalls/

Vector Databases

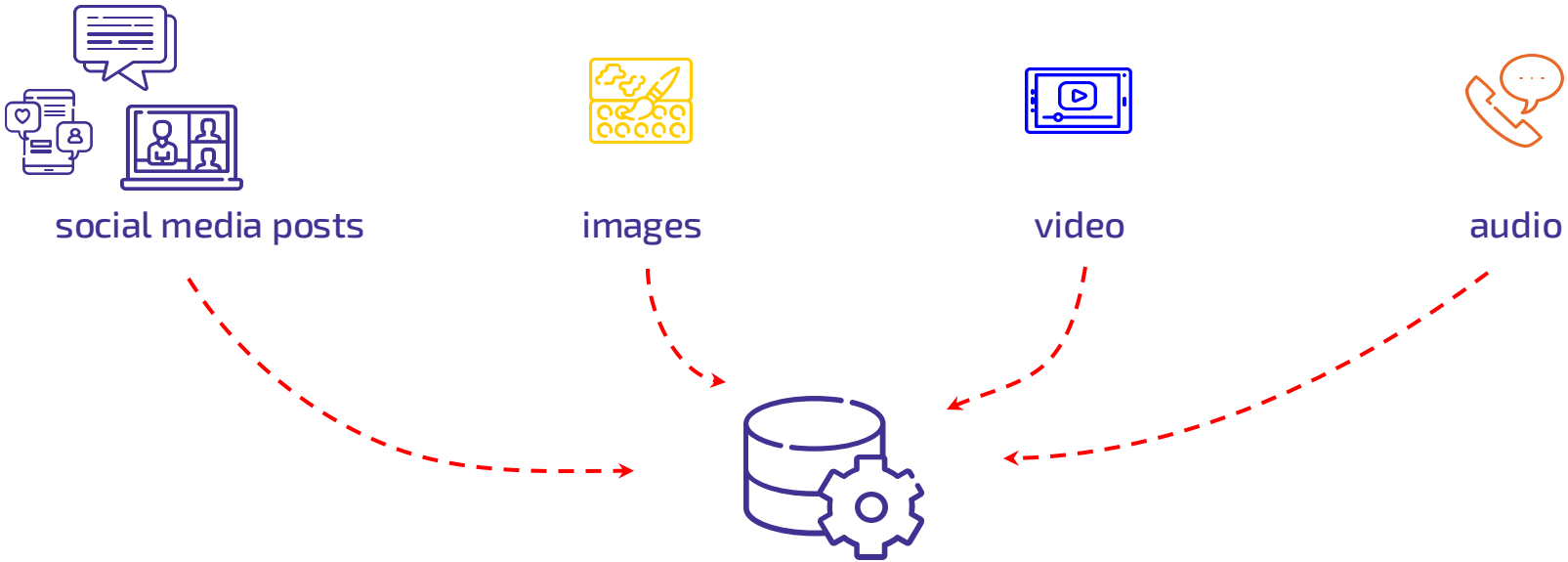
- What is a vector DB?
- How does it work?
- Use cases



Vector Databases | Why

unstructured data

> 80%



LLMs and Text generation

Controlling LLM Output:

Key Techniques

Greedy Decoding (Deterministic)
Always picks highest-probability token. Good for code/debugging, but can be bland or repetitive.

Temperature (Adjusts Sharpness)
Lower → Predictable, focused.
Higher → Varied, creative.
(Temp 0 = Greedy).

Sampling Methods (Refines Choices)
Top-k: Restricts to "k" most likely tokens.
Top-p (Nucleus): Includes tokens with cumulative probability exceeding a threshold (e.g., 90%). Adapts to certainty.

Takeaway: Tuning temperature, top-k, top-p, and penalties shapes predictability vs. creativity. Experiment to find the right fit for your application.

Choosing the Right LLM for Your RAG Application: Key Factors

Quantifiable Metrics (Performance & Cost)

Model Size (Parameters): Billions (e.g., 1–10B vs. 100–500B+).
Trade-off: Capability vs. Cost.

Cost: Price per million tokens (Input/Output). Newer/larger models cost more.

Context Window: Max tokens (Prompt + Completion). More flexibility, pay per token.

Speed & Latency: Time to first token, Tokens/sec. Critical for real-time.

Training Cutoff: Knowledge cutoff date. Later is preferable for recent events.

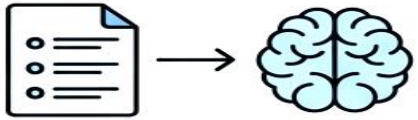
Quality & Benchmarking (Evaluation)

Quality is hard to quantify (reasoning, readability). **Benchmarks** help compare.

Takeaway: Balance these quantifiable metrics and benchmark results against your project's specific speed, quality, and budget requirements.

Advanced Prompt Engineering for RAG: Techniques & Application

In-Context Learning (Grounding Responses)



Concept: Provide examples to guide LLM structure, tone, and output format.

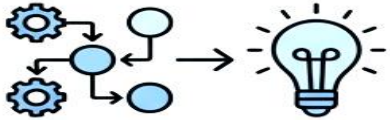
Types: One-shot (single example), Few-shot (multiple examples).

Implementation: Hard-code stable examples or use RAG to retrieve contextually relevant past examples.

Benefit: Grounds the LLM, improving response quality and consistency.

RAG for Examples: Retrieve successful past interactions to inject dynamic, high-quality examples.

Reasoning & Chain of Thought (Step-by-Step Logic)

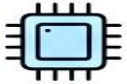


Concept: Encourage the LLM to plan and reason before generating a final answer.

Techniques: 'Think step-by-step', 'Scratchpad' tags (intermediate thinking space), Chain of Thought (generate steps, then follow them).

Benefit: Increases accuracy for complex tasks (math, coding, planning), makes errors traceable.

Evolution: Rise of 'Reasoning Models' with built-in planning & thought-process tokens.

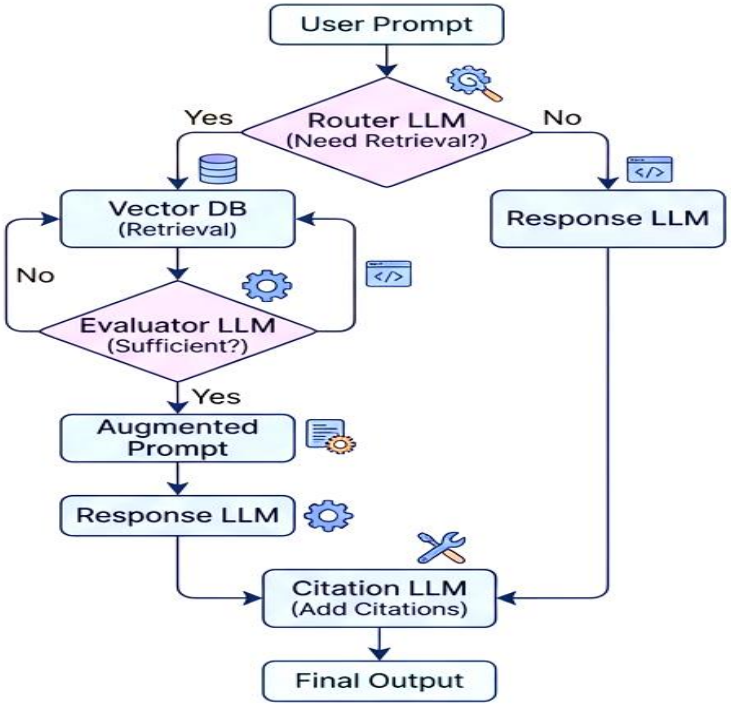


Reasoning Models: Designed specifically for multi-step complex workflows right out of the box.

Goal: Leverage these techniques to produce more accurate, structured, and high-quality RAG outputs.

Agentic Workflows in RAG: Modular & Specialized Systems

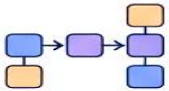
Example Agentic RAG System
(Multi-Step Process)



Key Concepts

- **Tasks as Series of Steps**
  (Flow Chart Design)
- **Access to Tools**
  (Code, Web, Vector DB)
- **Specialized LLMs per Step**
  (Mix & Match Sizes/Strengths)

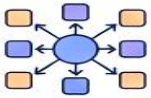
Common Patterns



Sequential
Linear flow of LLMs

Conditional
Router LLM decides path

Iterative
Feedback loops for refinement

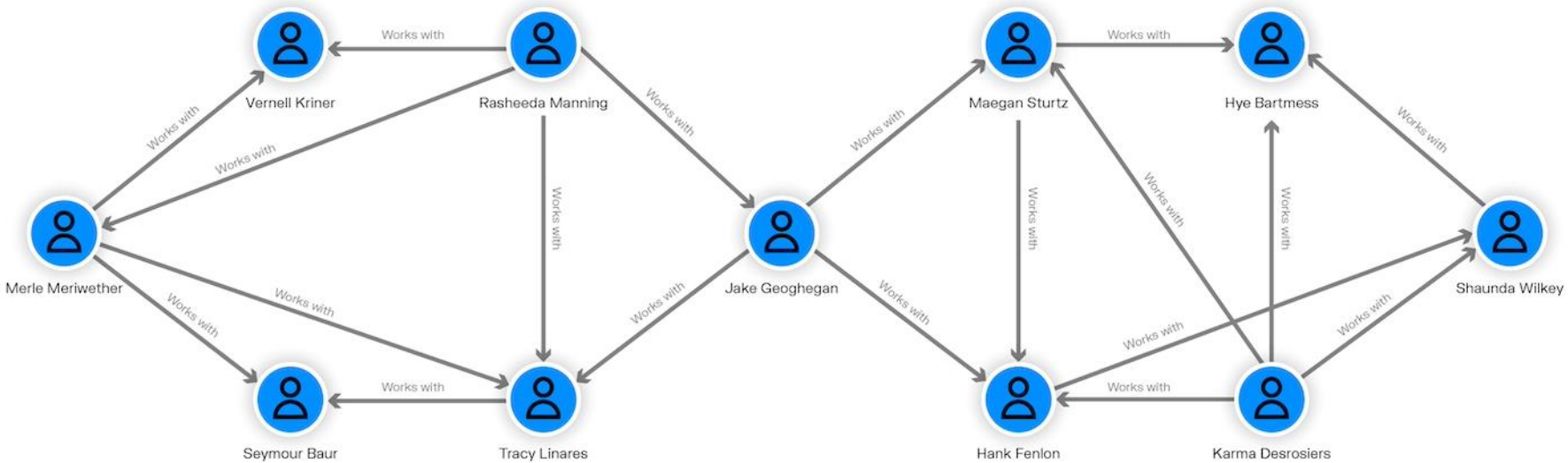


Parallel
Orchestrator breaks & synthesizes tasks

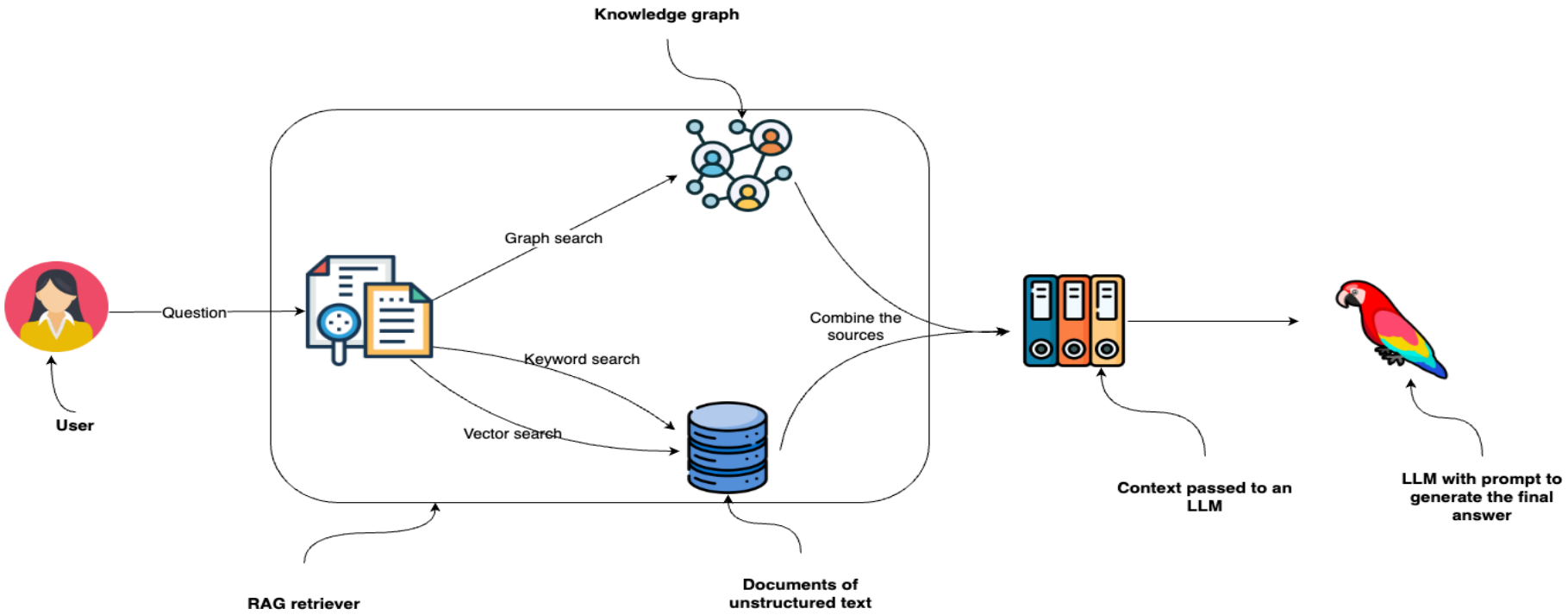
Mindset Shift: LLMs as Modular Pieces in a Larger Workflow for Enhanced RAG Capabilities.

Knowledge Graphs

What is a Knowledge Graph

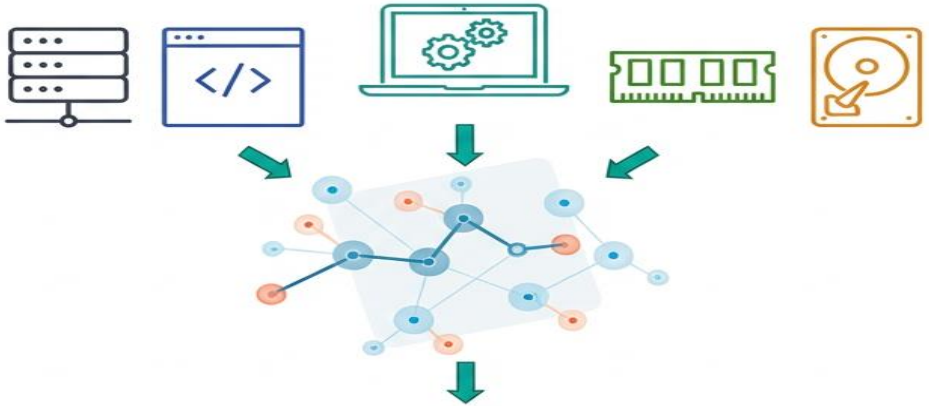


Knowledge Graphs + RAG



Graph-enabled RAG

Why is my new laptop running so slow? 🤔



How different technical details are *connected*.

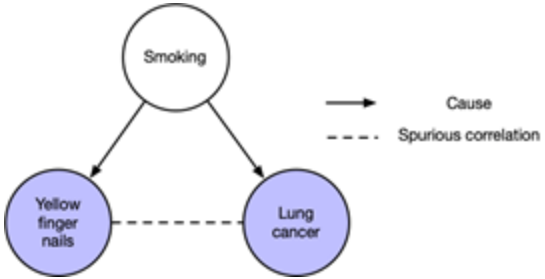
Your slow laptop performance, combined with high CPU usage from background apps and low available RAM, suggests insufficient memory for your workload.
Consider upgrading RAM or closing unnecessary programs.

LLM with GraphRAG

Why Graphs?

https://deepgraphlearning.github.io/project/wikidata5m

Knowledge Graph



An introduction to causal reasoning in health analytics (Zhang et al)

Causal Graph

https://towardsdatascience.com/visualizing-protein-networks-in-python-58a9b51be9d5

Interaction Graph

In many ways, graphs are the main modality of data we receive from nature.

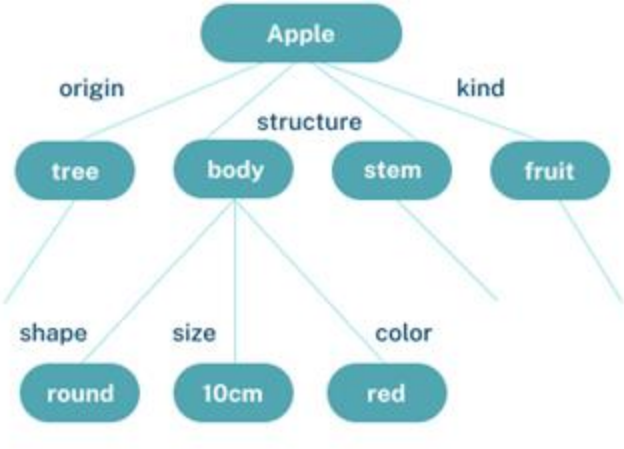
– Petar Veličković

Why Graphs?

Human View of an Apple

Vector View of an Apple

Knowledge Graph View of an Apple



In many ways, graphs are the main modality of data we receive from nature.

– Petar Veličković

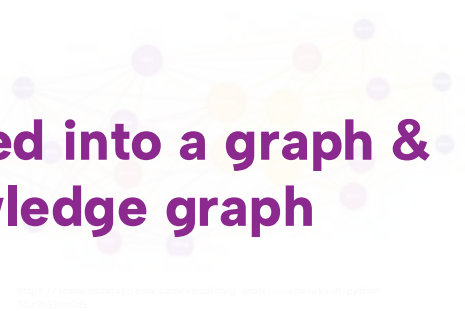
https://neo4j.com/blog/graphrag-manifesto/

Why Graphs?

(Practically) all structured data can be converted into a graph & (almost) every graph can be considered a knowledge graph

Knowledge Graph

Causal Graph



Interaction Graph

In many ways, graphs are the main modality of data we receive from nature.

– Petar Veličković

What is a Knowledge Graph?

A KG consists of entities (**nodes**) that are connected by relationships (**edges**). Typically represented as list of (**subject**, **predicate**, **object**) triplets

WikiData
> 100 million items

CSKG
> 2 million items

Yago
> 10 million items

DrugBank
20 thousand items

What is a Knowledge Graph?

A KG consists of entities (**nodes**) that are connected by relationships (**edges**). Typically represented as list of (**subject**, **predicate**, **object**) triplets

Create private KG

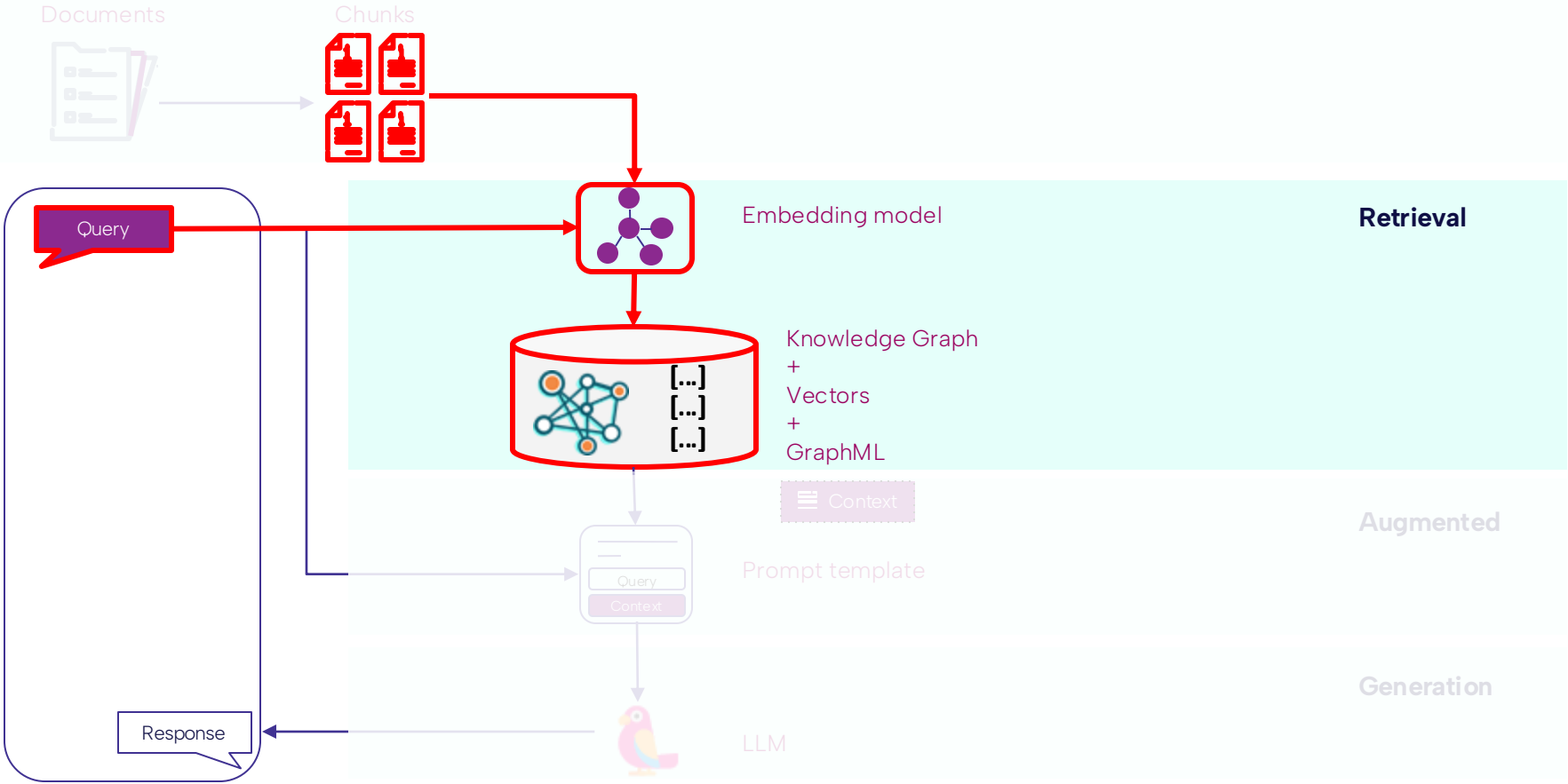
Financial Data

Medical Data

Vacation Planning

Reading List

GraphRAG



https://weaviate.io/blog/introduction-to-rag

dfki ai

DSA

RPTU Rheinland-Pfälzische Technische Universität Kaiserslautern Landau

Thank you!

Engineering with GenAI
09.12.2025