

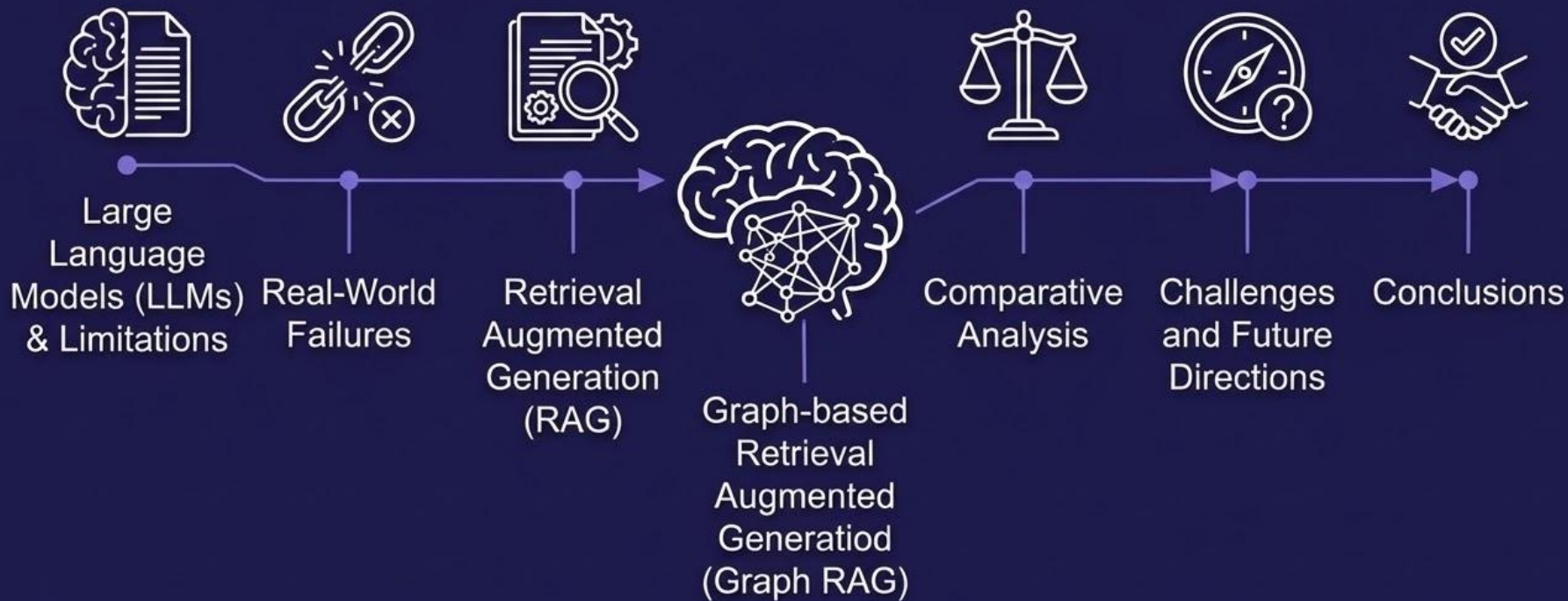
Retrieval Augmented Generation Techniques for Knowledge-Enhanced AI Systems

Nikolaos Fanourakis, PhD
Data Scientist @ SATALIA



CSD of UOC

Agenda



What is an LLM ?

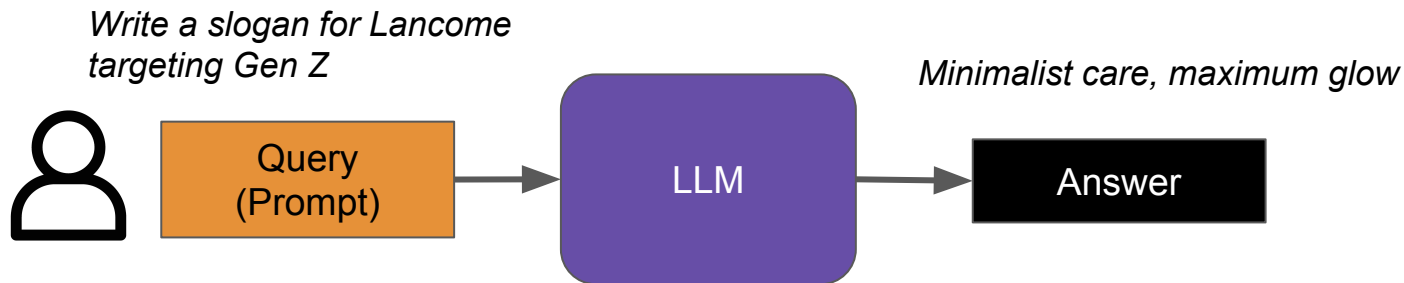
- LLMs: linguistic models trained on massive amounts of text
 - Recognize patterns
 - Predict next word in sequence
- Understand text
- Generate text, code, images etc



perplexity

Gemini

Normal Prompting Framework Under the Hood



Lancome: → is a luxury skincare brand

Luxury brand → often paired with “premium”, “minimalist”

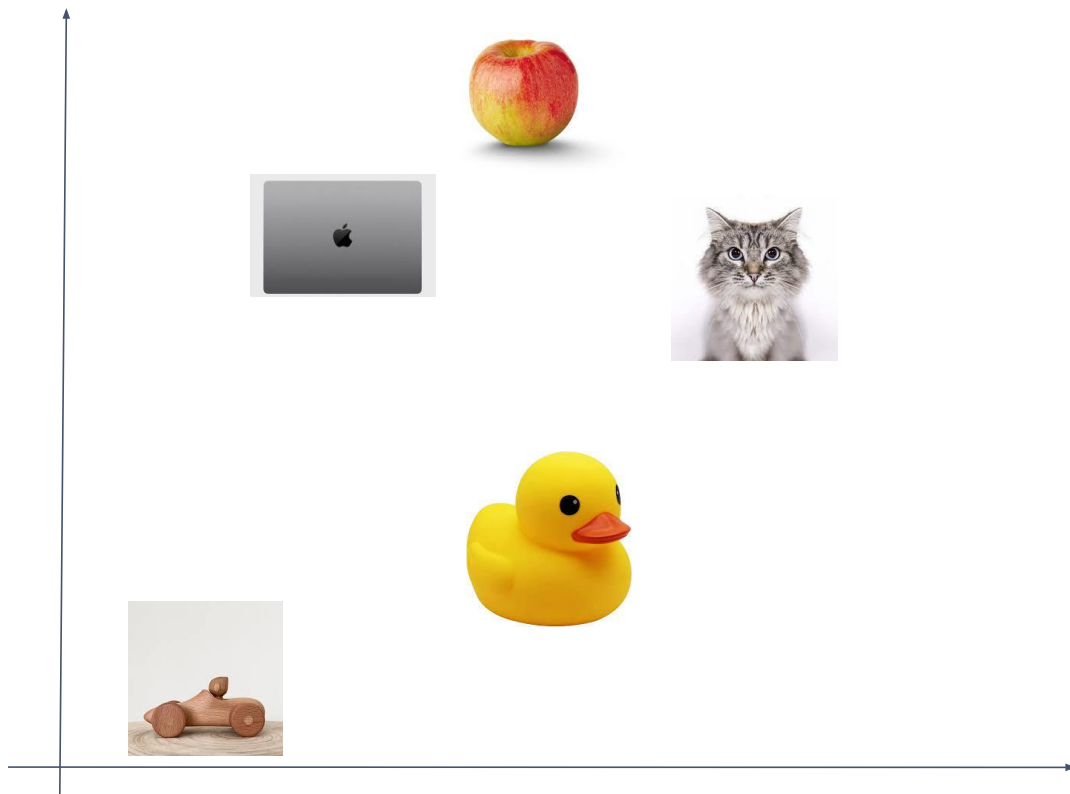
Skincare → often paired with “glow”, “healthy skin”

Gen Z → linked to “social media-friendly”, “self-care”

Let's organise the following



Suggestion



How about these ?

WPP today announces it has acquired a minority stake through a capital investment in OH-SO Digital, a new digital-first agency launching on 1 March.

WPP today announces that it has expanded its relationship with Telefónica across Spanish-speaking Latin America following a competitive pitch.

WPP today announced the merger of its two largest communications agencies, Hill & Knowlton and BCW, to form Burson, a powerhouse delivering modern communications leadership at scale to clients across the world. The merged company will become an industry-leading, full-service communications agency focused on building and protecting reputation.

Embeddings

- **Embeddings** are **numerical representations** of data, transforming words, phrases, images etc into vectors in a **high dimensional vector space**.
- **Semantically similar** entity representations are located in **close proximity** to each other in the space.
- Used to help **identify semantic similarities**
- Can be used by search algorithms, retrieval applications etc
- They can model vectors in such a way so that
london - england + france = paris

Limitations of Standard LLMs

- Lack of true understanding
 - Recognise patterns, not meaning
- Lack of real-time knowledge
 - Knowledge is based on past training data
- Limited personalization
 - Doesn't truly know the user unless data is provided
- Not always accurate answers
 - Can generate confident but incorrect information aka “hallucinations”

Real-world Example LLM Failure

ChatGPT's data is outdated

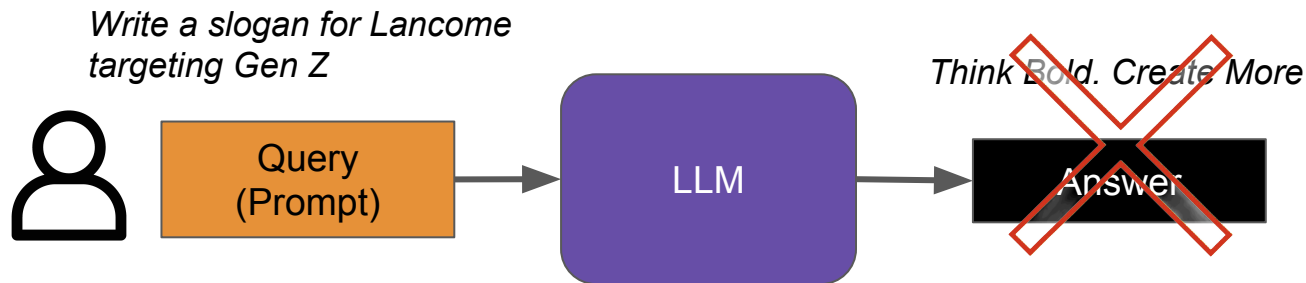
The AI only knows things from **before 2021**



Limitations

Limited knowledge of world and
events after 2021

Real-world Example LLM Failure



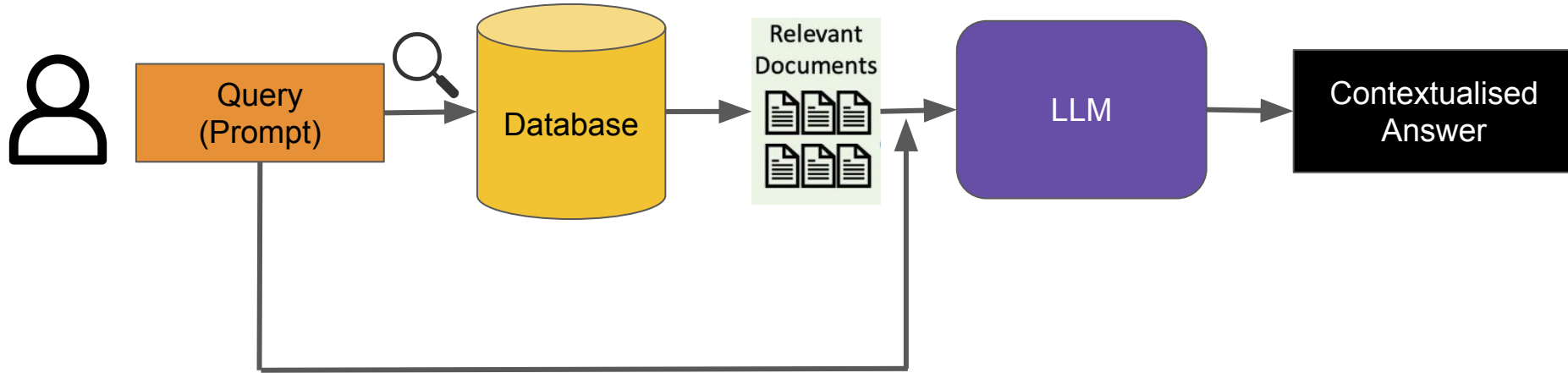
Lancome: → trained on data **before** Lancome became **widely recognized**

Luxury brand → often paired with “premium”, “minimalist”

Skincare → often paired with “glow”, “healthy skin”

Gen Z → linked to “social media-friendly”, “self-care”

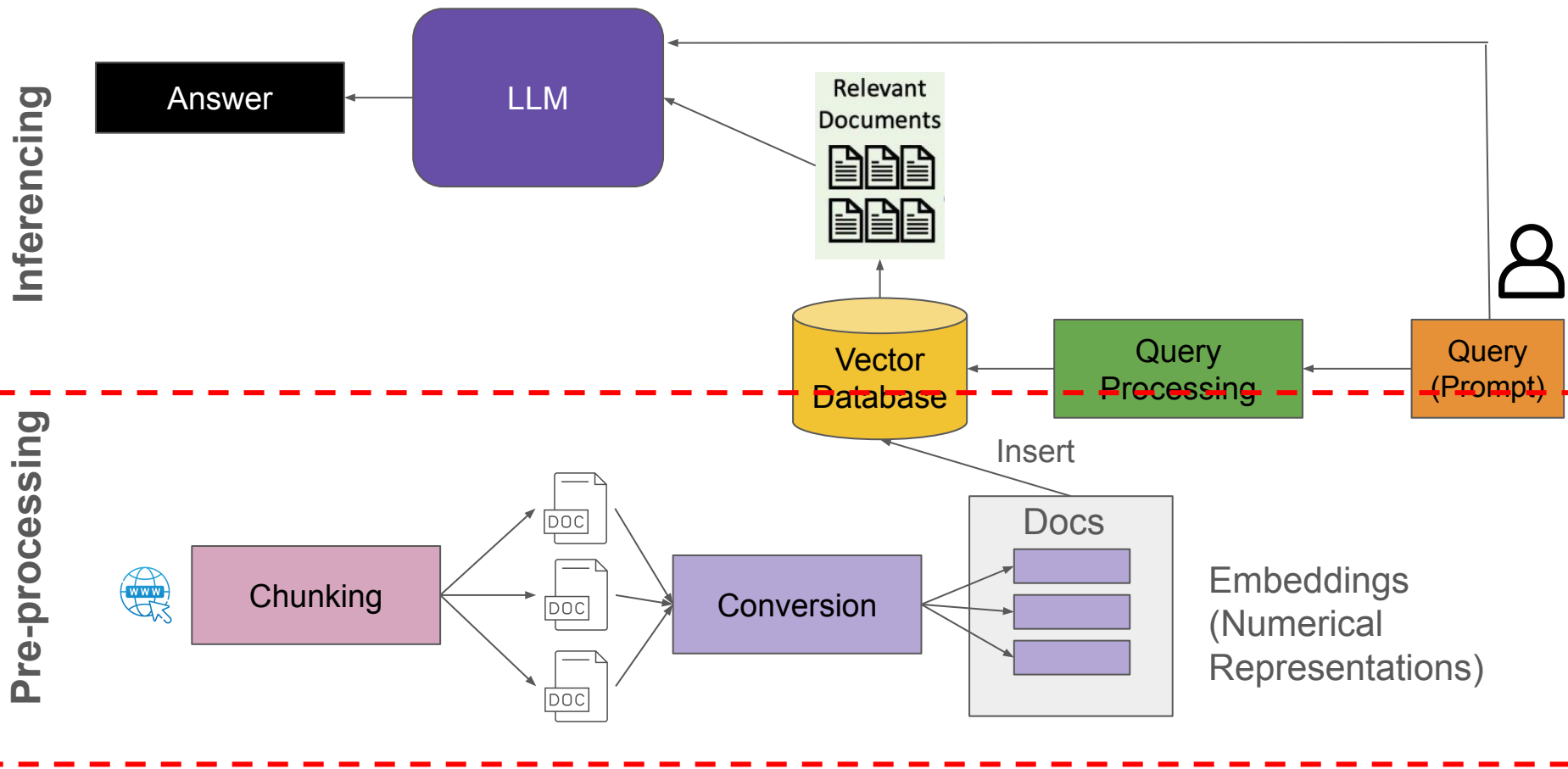
RAG: Retrieval Augmented Generation



RAG Applications

- **Healthcare**
 - Diagnose patients using retrieved peer-reviewed medical literature
- **Education**
 - Generate personalized study materials from academic sources.
- **Customer Service**
 - Enable chatbots to resolve queries using up-to-date product information.

Architecture



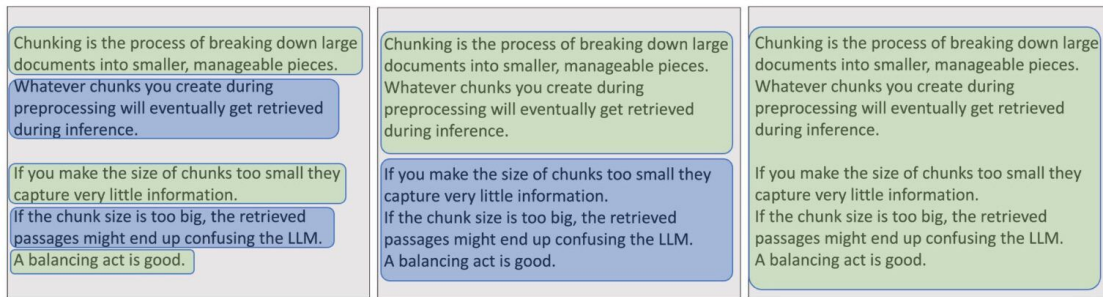
Pre-processing - Identify Data Source

- Choose a relevant data source based on the application
- Assess data quality—completeness, accuracy, consistency, and noise



Pre-processing - Chunking Data

- **Break down** the docs into **smaller documents or chunks**
- The **quality** of chunking impacts **retrieval effectiveness** and **RAG performance**
- **Chunks** created during preprocessing are the **exact units returned** during inference



Sentence-Level Chunking

Paragraph level Chunking

Entire document

retrieval less accurate (lack of context)

overload or confuse the LLM

Pre-processing - Structural Chunking

- Some datasets come with a **clear structure** that can guide chunking, **reducing** the need for **arbitrary size-based splits**
- Leverage existing structure **preserving context** and **improves retrieval quality**

Chunking (computing)

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

Not to be confused with [Thinking](#).

In computer programming, **chunking** has multiple meanings.

In memory management

Typical modern [software](#) systems allocate [memory](#) dynamically from structures known as [heaps](#). Calls are made to heap-management routines to allocate and free memory. Heap management involves some computation time and can be a performance issue. **Chunking** refers to strategies for improving performance by using special knowledge of a situation to aggregate related memory-allocation requests. For example, if it is known that a certain kind of object will typically be required in groups of eight, instead of allocating and freeing each object individually, making sixteen calls to the heap manager, one could allocate and free an array of eight of the objects, reducing the number of calls to two.

In HTTP message transmission

Main article: [Chunked transfer encoding](#)

Chunking is a specific feature of the [HTTP 1.1](#) protocol.^[1] Here, the meaning is the opposite of that used in memory management. It refers to a facility that allows inconveniently large messages to be broken into conveniently-sized smaller "chunks".

In data deduplication, data synchronization and remote data compression

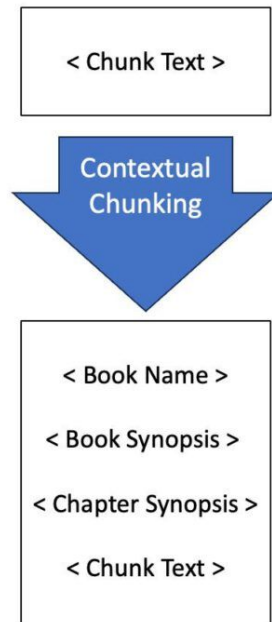
In [data deduplication](#), data synchronization and remote data compression, **Chunking** is a process to split a file into smaller pieces called chunks by the chunking algorithm. It can help to eliminate duplicate copies of repeating data on storage, or reduces the amount of data sent over the network by only selecting changed chunks. The Content-Defined Chunking (CDC) algorithm like [Rolling hash](#) and its variants have been the most popular data deduplication algorithms for the last 15 years.^[2]

Structured Chunking

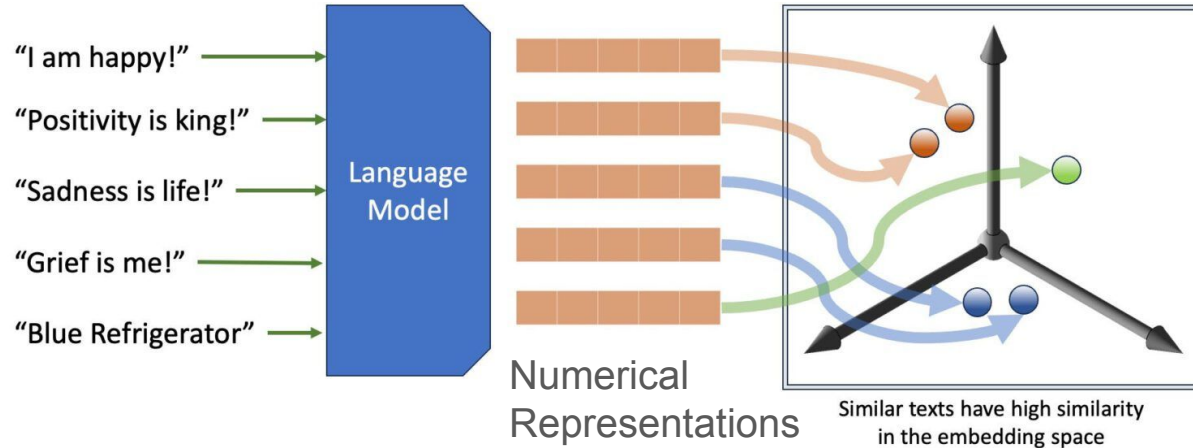
If you are chunking HTML,
use H2 (or H1) tags.

Pre-processing - Contextual Chunking

- Traditional chunking methods **fail** when documents **lose** their **broader context**
- Example:
 - If you chunk 10K of Sherlock Holmes paragraphs individually, you lose which book or chapter each paragraph came from
 - What was the first crime in “A Study in Scarlet” ?
 - Any chunk containing “crime” regardless of the book is retrieved



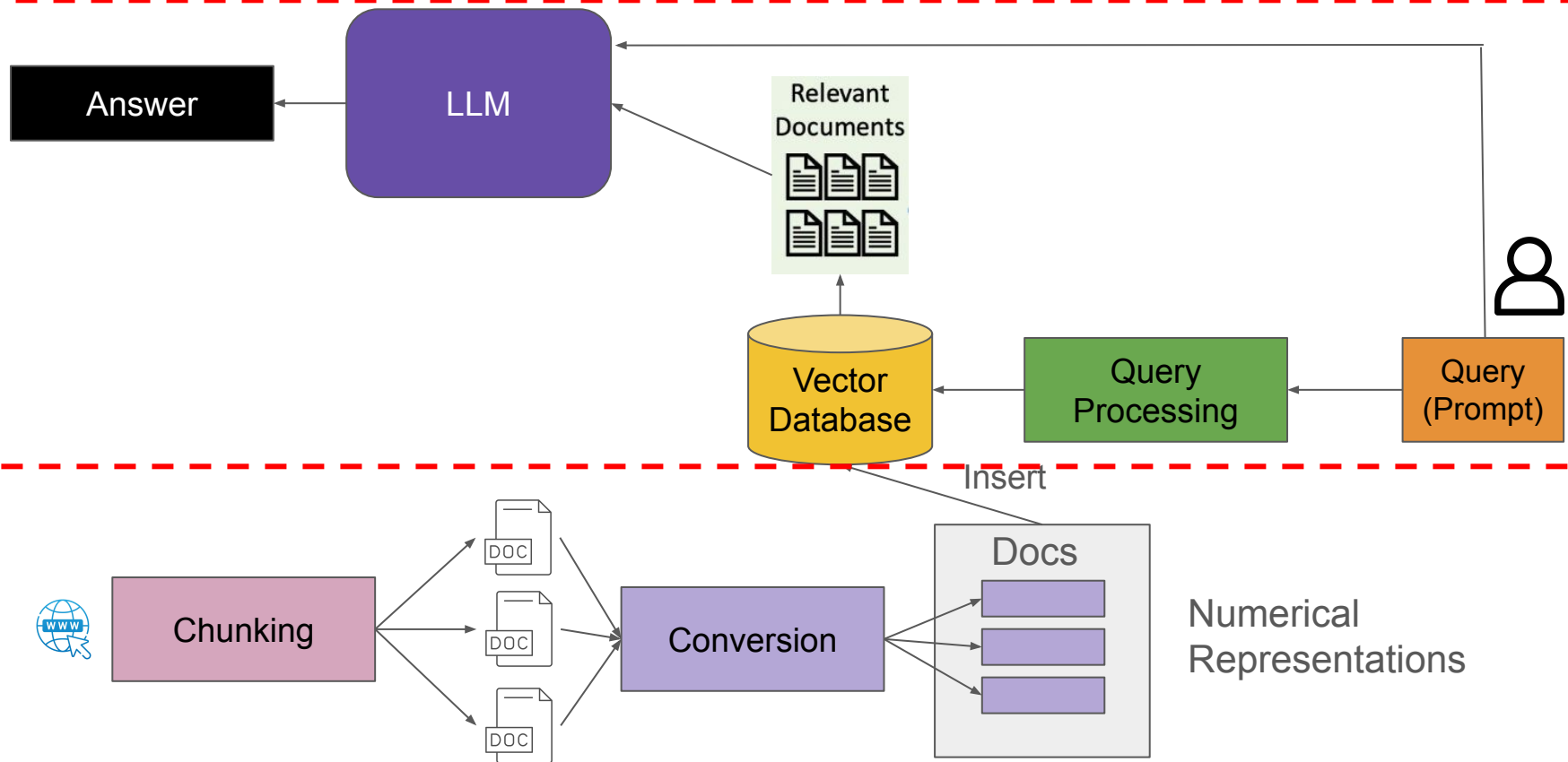
Pre-processing - Data Conversion



Architecture

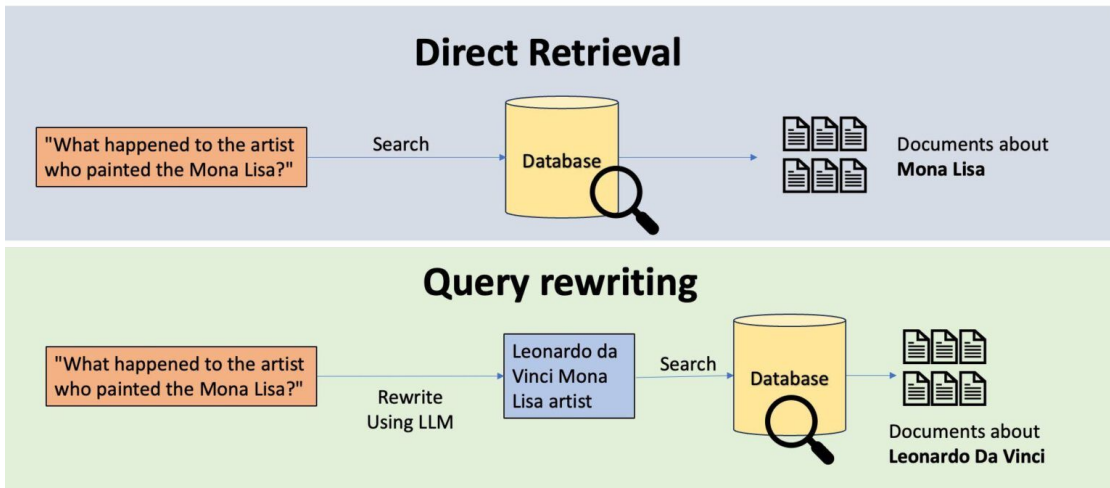
Inferencing

Pre-processing



Inferencing - Query Processing

- **Query** is **firstly** converted into an **embedding** and then find the **nearest/most similar docs** using **similarity metrics** e.g., cosine similarity
- **Retrieve** relevance docs

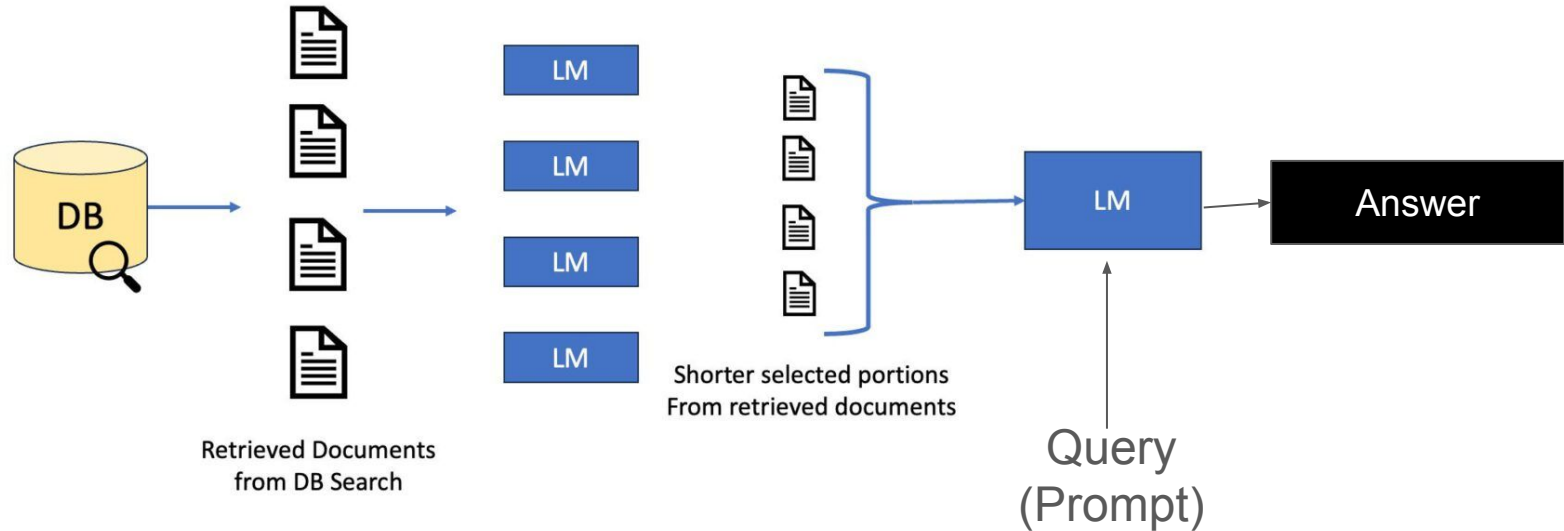


Pre-processing - Database

- **Vector databases** are the most common database type used in RAG systems
- They store documents by **indexing** them using **embeddings**
- Their strength is **fast similarity search** between query vectors and indexed vectors, making them ideal for RAG

Inferencing - Synthesis

Ask LLM to select important portions from the retrieved docs



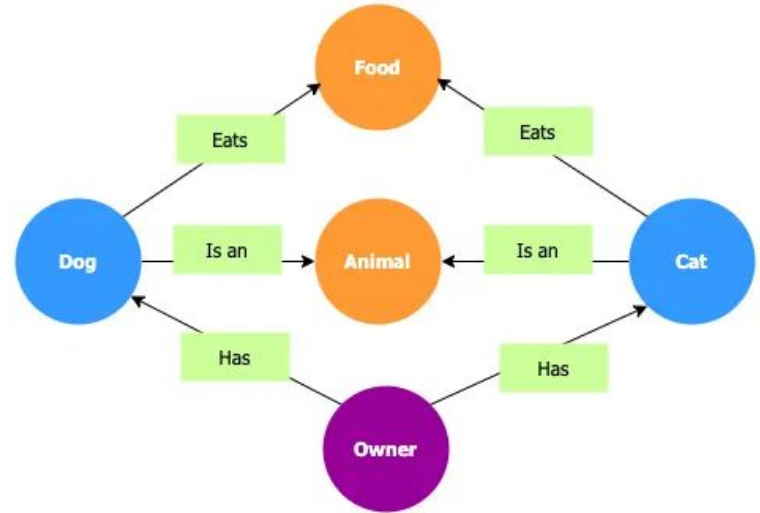
When RAG is not enough ?

- **Long unstructured docs:** Difficult to retrieve **relevant** chunks **efficiently**
- **Limited semantic understanding:** Standard retrieval algorithms may fail to fully capture the meaning of a query, reducing the relevance of fetched documents.
- **Missing relationships between concepts:** Unstructured text does not capture entity connections leading to poor reasoning

Need for structured knowledge: Structured data (graphs) are required

Knowledge Graph

- **KGs** are **structured representations** of real-world **entities** and their **relationships**
- They consist of two main components: **nodes (entities)** and **edges (relations)**
- **Triple:** Dog → Eats → Food



Graph RAG

- Graph RAG enhances traditional RAGs by using **knowledge graphs** to store and retrieve structured information
 - Stores data as **nodes (entities) and edges (relationships)**
 - Combines vector similarity with graph-based reasoning
 - **Vector similarity search** (to retrieve semantically similar documents)
 - **Graph traversal** (to find related entities and their relationships)

Example

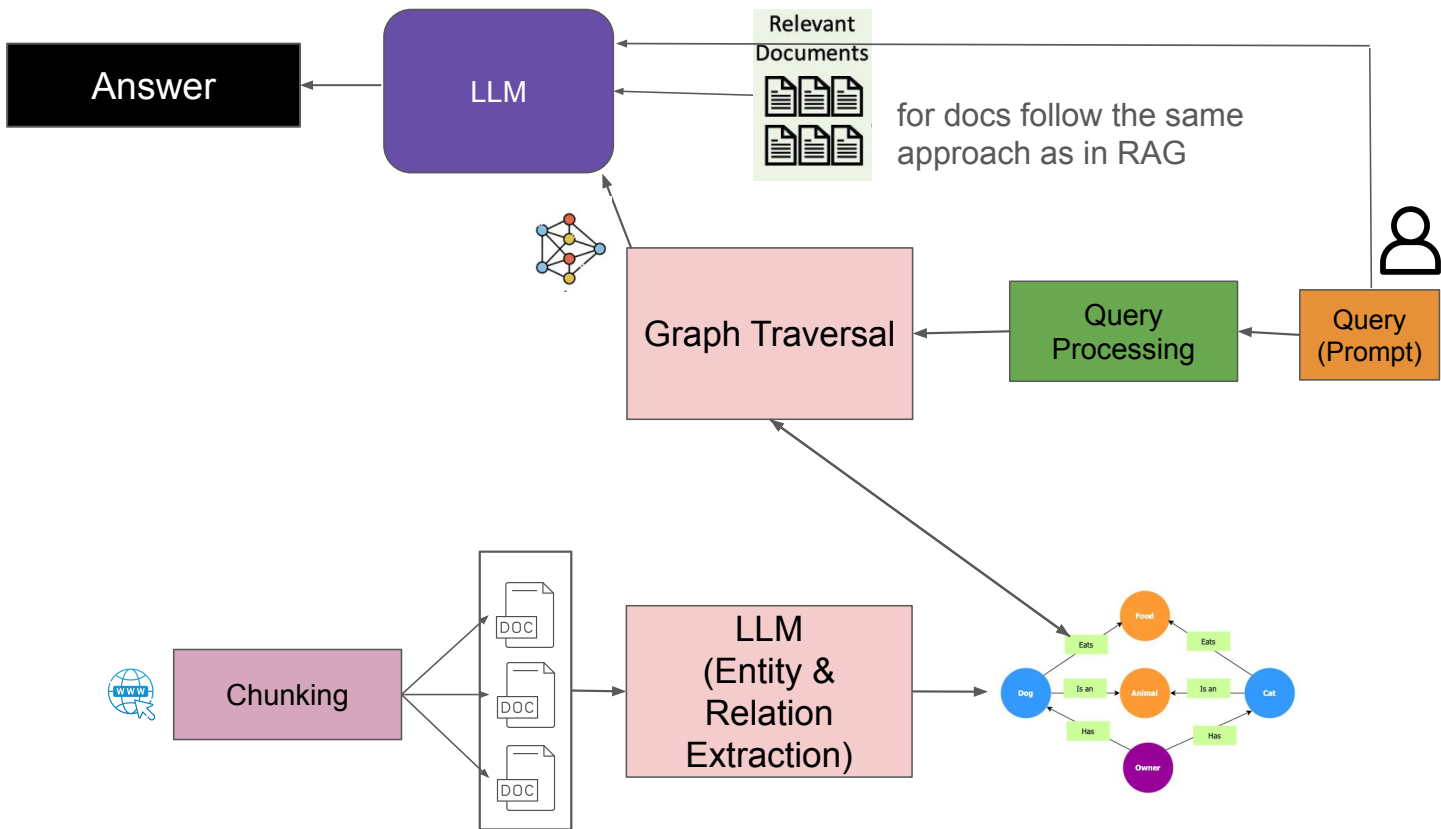
Traditional RAG retrieves articles on “Tesla” or “Battery Technology” separately.

Graph RAG understands the relationship: **Tesla** → **Uses** → **Lithium Batteries** → **Which Degrade Over Time** → **Due to High Charge Cycles**.

Architecture

Inferencing

Pre-processing

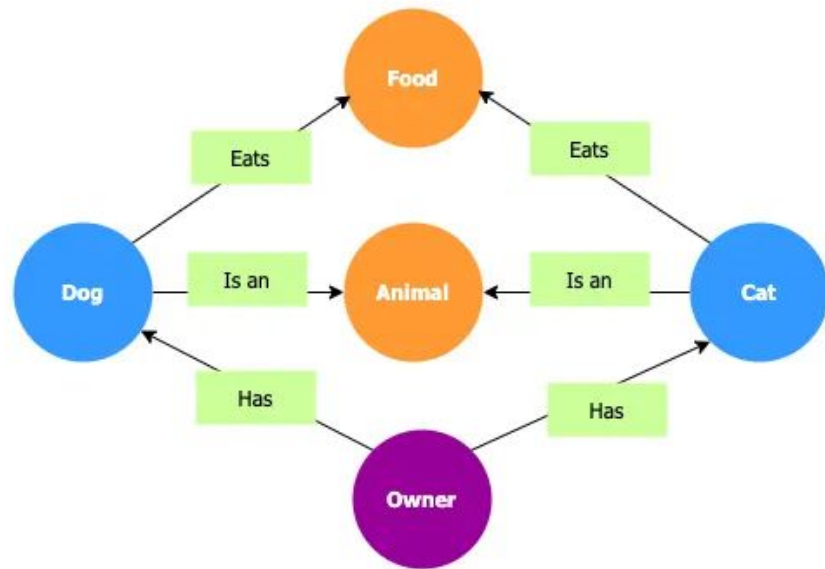


Preprocessing - KG Construction

- **LLM identify entities** from documents
- **LLM defines relationships** between entities

Doc: *The **dog** and the **cat** belong to the same **owner**, and they both love their **food***

Entity	Head Entity	Relation	Tail Entity
Dog	Dog	Has	Owner
Cat	Cat	Has	Owner
Owner	Dog	Eats	Food
Animal	Cat	Eats	Food
Food	Dog	Is an	Animal
	Cat	Is an	Animal



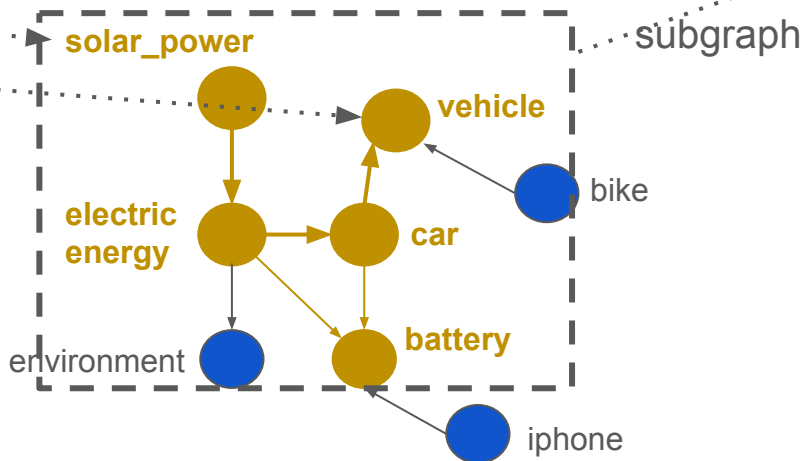
Preprocessing - Query Processing & Graph Traversal

- Identify entities and relations in query
- Graph traversing on KG (e.g., BFS)

Query Understanding

How does **solar power** help **vehicles**?

KG and Extracted Subgraph



LLM + Subgraph Answer

Solar power generates electric energy, which can be used by electric vehicles like cars through battery charging systems.

LLM vs RAG vs Graph-RAG

	LLMs (No Retrieval)	Standard RAG (Vector Search)	Graph RAG (Graph + Retrieval)
Retrieval Type	None	Vector similarity search	Vector similarity + graph traversal / graph reasoning
Knowledge Source	Fixed model parameters	External documents (unstructured text)	Documents + structured graph (entities + relationships)
Strengths	Fluent generation and general knowledge	Reduces hallucinations and handles long docs	Multi-hop reasoning, relational queries, structured + semantic retrieval
Weaknesses	Hallucinations, no access to new info, weak factuality	Misses relationships, limited multi-step reasoning, still may hallucinate	Requires graph construction, higher complexity, graph maintenance cost
Hallucination Risk	High	Medium	Low
Best At	Creativity, summarization, open-ended tasks	Open domain Q&A, factual lookup, content summarization	Complex reasoning, biomedical/legal tasks, entity-heavy queries

Challenges

- Retrieval quality issues
 - irrelevant, incomplete, or poorly chunked data reduces answer accuracy
- Graph construction is fragile
 - errors in entity/relation extraction
 - ongoing graph drift make it hard to build and maintain a clean KG
- Scalability
 - embedding, indexing and graph-building are costly and complex.

What's next ?

- **Adaptive, self-optimizing retrieval** that automatically learns optimal **chunking** and reduced **retrieval noise**
- **Robust, self-maintaining** knowledge graphs where models continuously validate entities/relations, repair graph drift, and propose schema improvements
- **Long-context** and **hierarchical reasoning models** that summarize retrieved evidence that fit within context limits

Conclusions

- LLMs are easy to use, scalable and cost efficient
 - usually hallucinate due to lack of understanding, real-time and domain-specific/personalised knowledge
- RAGs enhance LLMs with knowledge from external sources
 - hard to manage long unstructured docs → low efficiency and effectiveness
 - limited semantic understanding and poor reasoning
- GraphRags build on top of RAGs
 - use also structured knowledge (knowledge graphs)
 - better reasoning and semantic understanding
 - more complex and less scalable

Thank you !

Q&A

Nikolaos Fanourakis

PhD in Computer Science & AI/ML

Data Scientist @SATALIA

Email: nikosfanourakis5@gmail.com



LinkedIn