Advanced RAG

Iván Ruiz / Andrés Muñoz

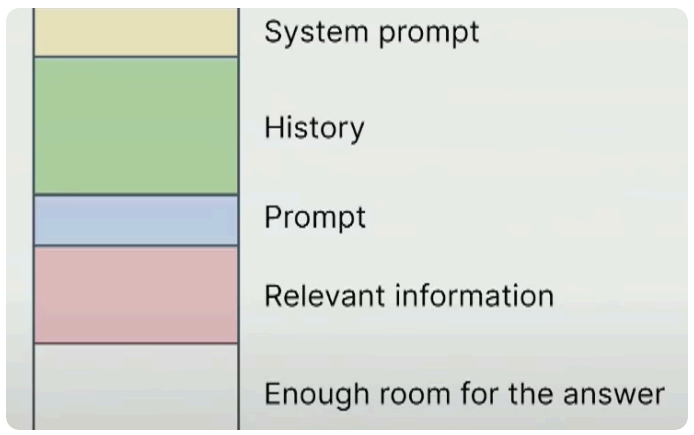
Contents

It's very easy to make RAG..

...but getting good results is not so easy

Several problems that need to be addressed...

Challenges in Facing a RAG System



Context Window

We should not include full documents as the context window is not unlimited*.

Token Consumption

Long documents can consume many tokens, generating high costs and/or longer response times

Irrelevant Information

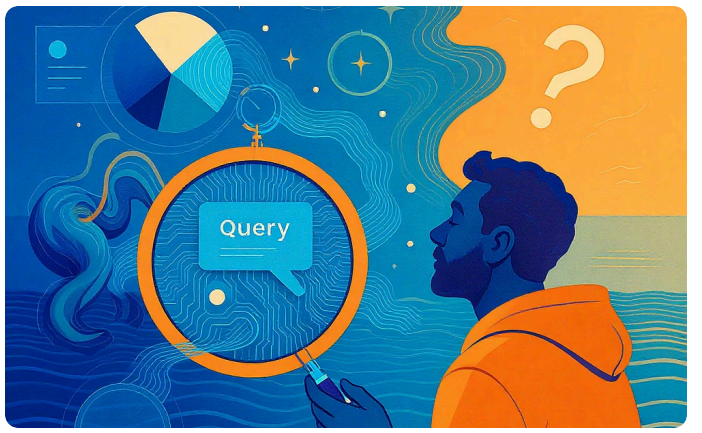
Including too much irrelevant information in the input prompt can reduce the accuracy of the responses provided by the LLM.

Question Complexity

The user might ask *"How do the advances made at the University of Cádiz in the field of biotechnology compare to those achieved by other universities, taking into account the publications in the last five years?"*

Stylistic Inconsistency

Variations in style and tone make it difficult to achieve coherence in the final result.



Implicit Questions

Queries may refer to implicit and unknown information for the retriever.

1

Software can be chaotic, but we make it work

Expert

Trying Stuff Until it Works

O RLY?

The Practical Developer
@ThePracticalDev

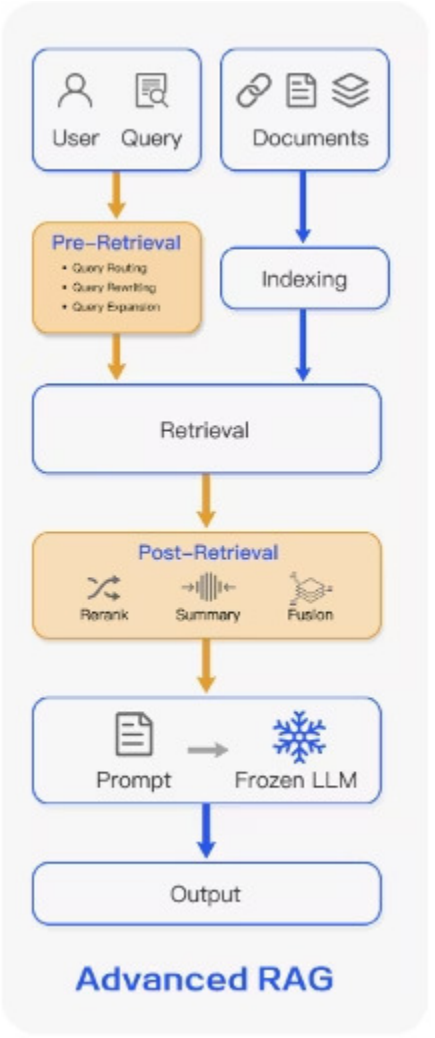
Advanced RAG Techniques

Pre-retrieval

Techniques applied before information retrieval.

Post-retrieval

Methods used after retrieval.



Pre-Retrieval Techniques

Pre-Retrieval Techniques

Query Rewriting

Query reformulation is a crucial technique for optimising search and information retrieval.

Query Expansion

Expanding the search involves adding additional terms to the original query to improve the precision and comprehensiveness of the results.

Query Compression

Query compression involves contextualising the original query to optimise the efficiency and precision of the search.

Query Routing

Routing the query to the most relevant data sources, optimising efficiency and relevance.

Query Rewriting: Query Reformulation

1

Identification

The original user query is analysed to detect areas for improvement.

2

Processing

An LLM modifies the query by including related terms and recognised entities.

3

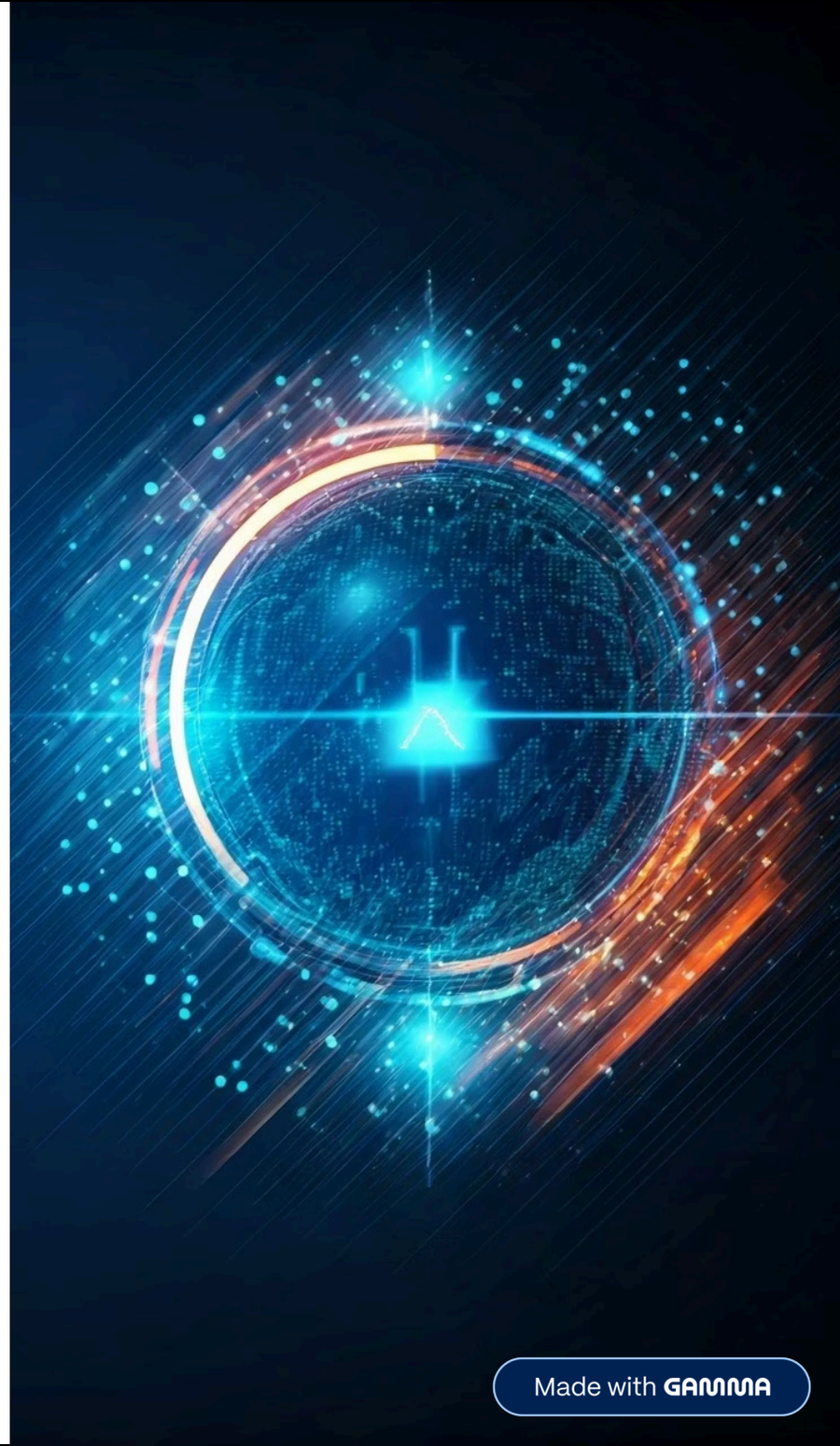
Optimisation

A paraphrase is performed to align with the vocabulary of the knowledge base.

4

Refinement

Spelling and grammatical corrections are applied to improve accuracy.



Query Rewriting: Example

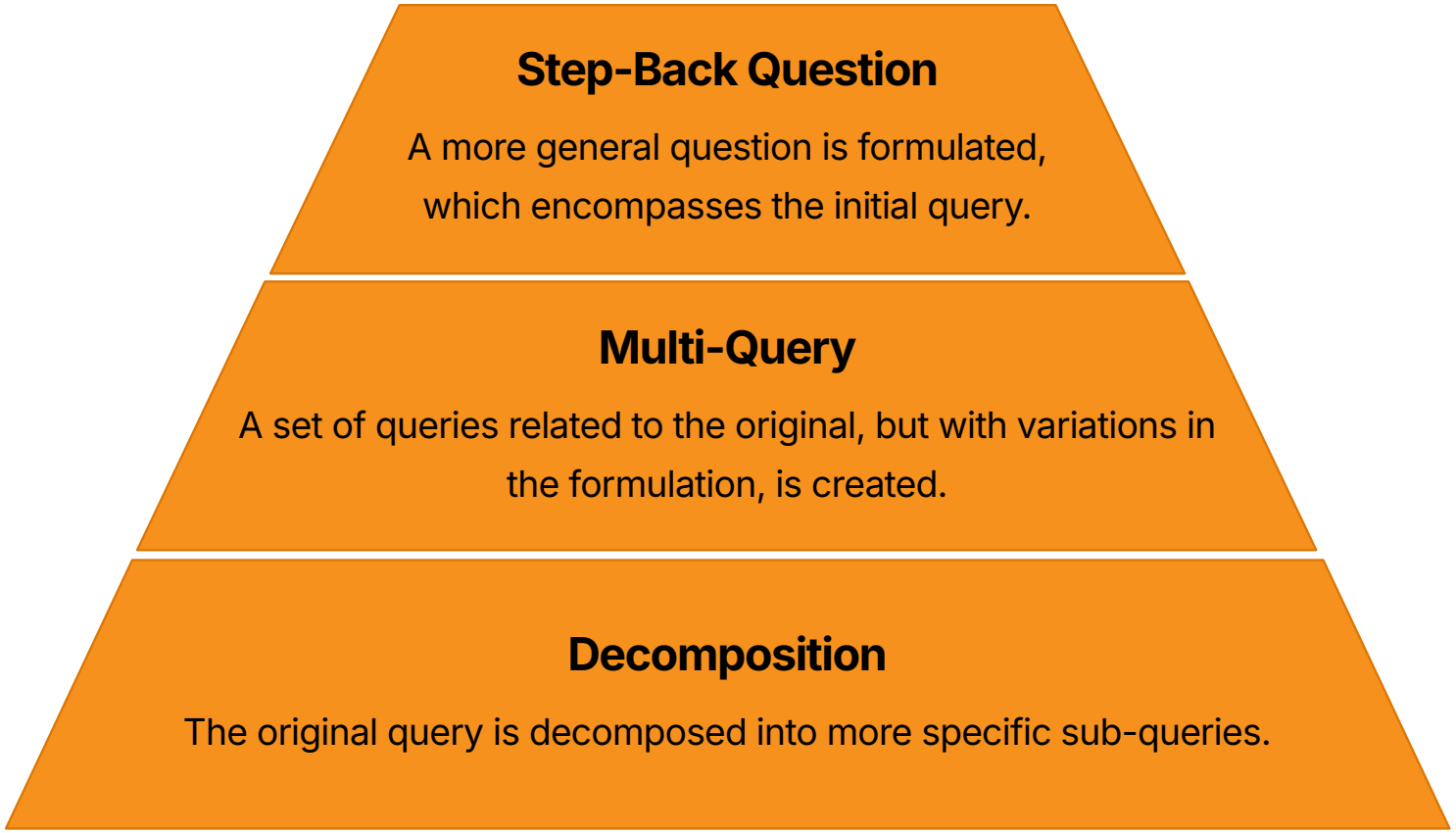
Original Query

"Renewable projects Cádiz university, research groups?"

Modified Query

"What are the renewable energy research projects being developed by the research groups at the University of Cádiz?"

Query Expansion: Broadening the Search



Step-Back Question

1

Specific Query

It starts from the original user's question.

2

Generalisation

A more abstract and broader question is formulated.

3

Dual Retrieval

Results are obtained from both the original question and the generalised one.

4

Synthesis

The information is combined to provide a more complete and contextualised answer.

Step-Back Question: Example

Original Query

"What specific strategies can a first-year engineering student apply to improve their performance on mathematics exams at the University of Cádiz?"

Additional Query

"What are the factors that influence the academic performance of first-year students at the University?"

Multi-Query

1

Initial Query

It starts from the original user's question.

2

Generation

Multiple variants are created while maintaining the level of abstraction.

3

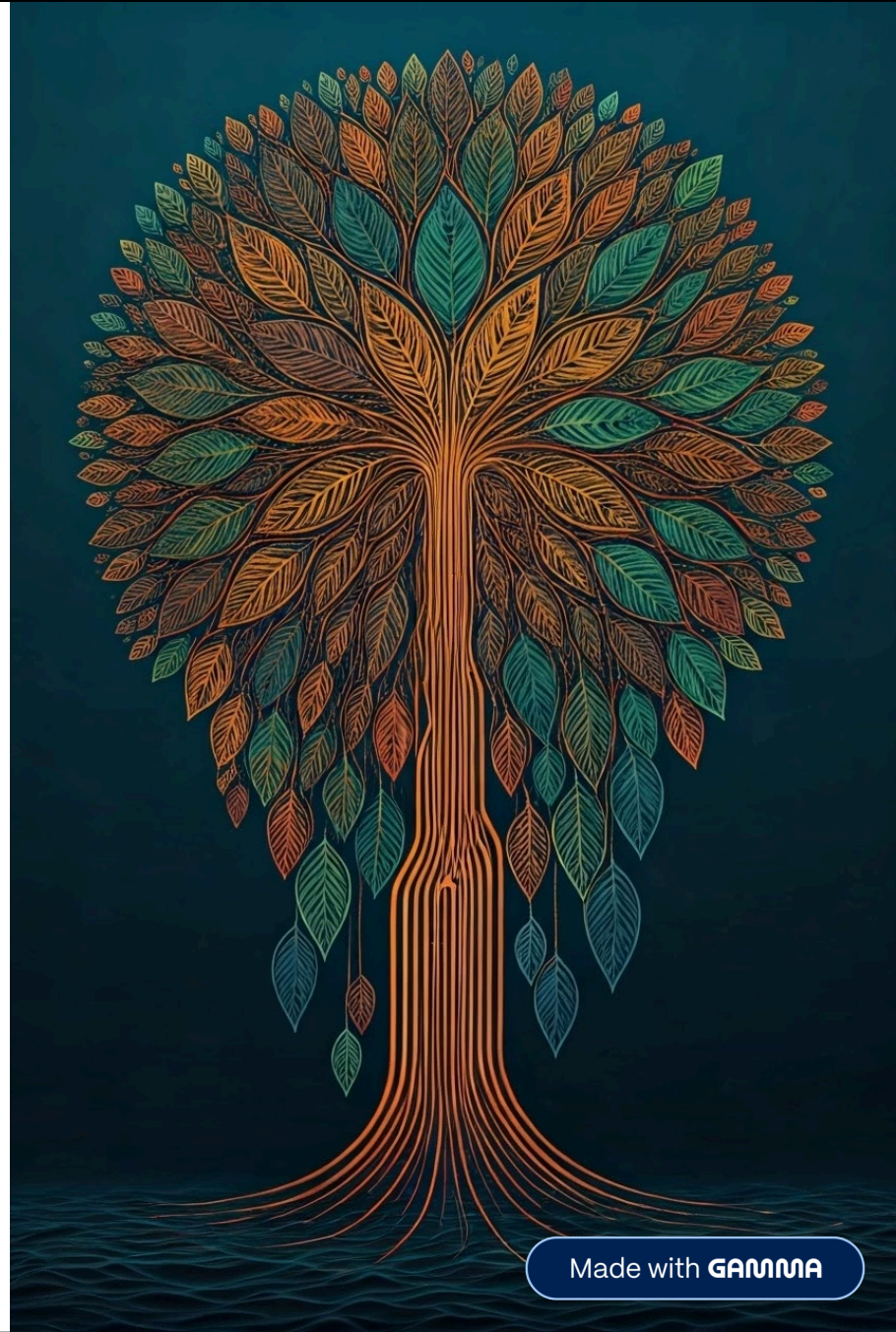
Diversification

Different perspectives of the same query are addressed.

4

Retrieval

All variants are used to obtain more complete results.



Multi-Query: Example

Original Query

"What exchange programmes does the University of Cadiz offer?"

Generated Queries

"What international agreements does the University of Cadiz have with European universities?"

"What requirements are needed to participate in the Erasmus+ programme?"

"Are there exchange opportunities outside of Europe, such as in America or Asia?"

"What financial support or scholarships does the University of Cadiz offer for exchange students?"

Query Decomposition

1

Analysis

The original query is examined to identify key components.

2

Decomposition

It is divided into more specific and concrete sub-questions.

3

Processing

Parallel or sequential queries are performed depending on the nature of the sub-questions.

4

Retrieval

All sub-questions are used to obtain more comprehensive results.

Query Decomposition: example

Original Query

"What are the most common evaluation methods in science degrees at the University of Cádiz and how do they compare to the methods used in humanities degrees?"

Derived Sub-queries

"What are the most common evaluation methods in science degrees at the University of Cádiz?"

"What are the evaluation methods used in humanities degrees at the University of Cádiz?"

"How do the evaluation methods compare between science and humanities degrees?"

Query Compression: Contextualising the Query

1

Context Analysis

The previous conversation history is examined.

2

Compression

The relevant context information is synthesised.

3

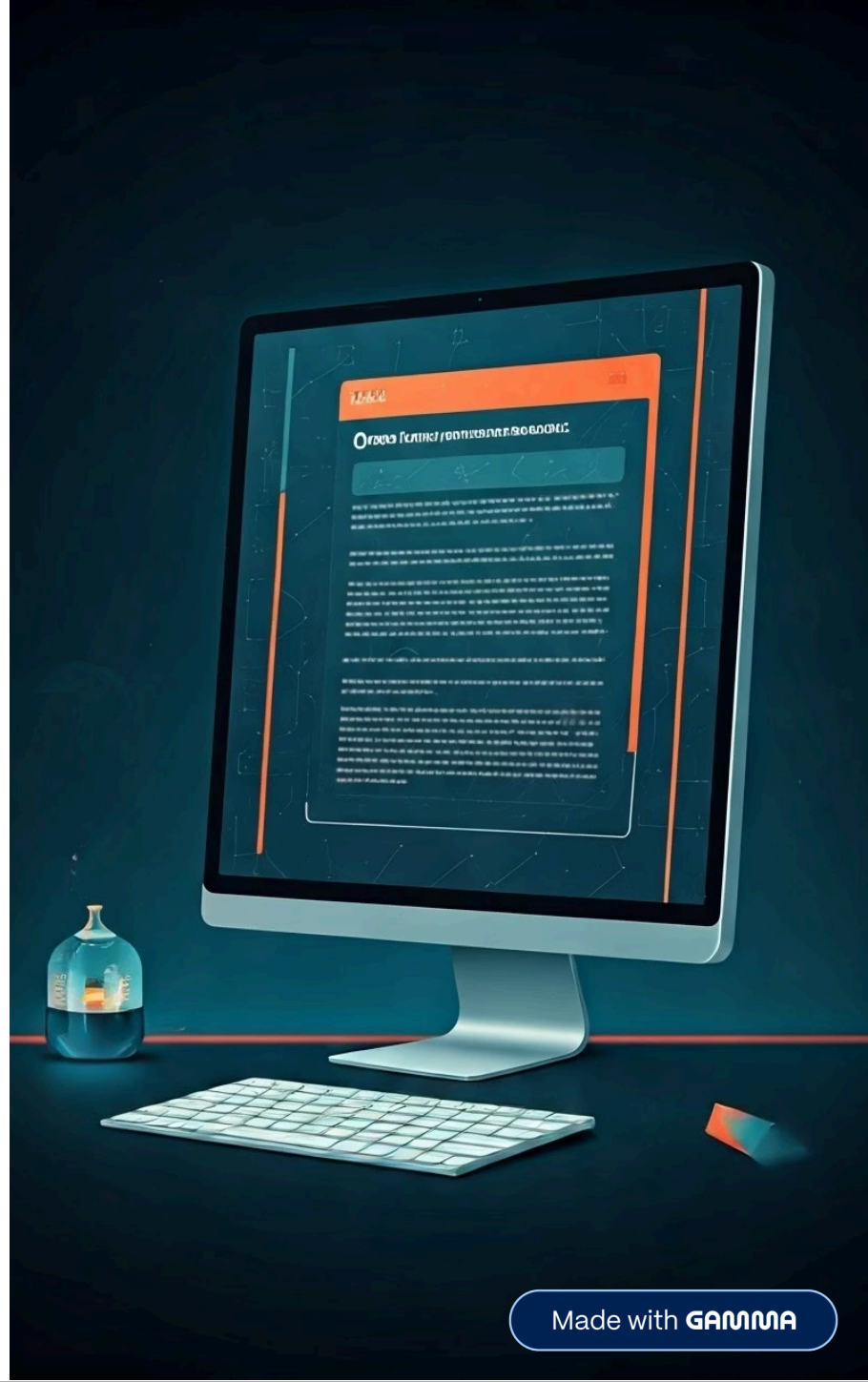
Reformulation

A new query that integrates the compressed context is created.

4

Optimised Retrieval

The compressed query is used to obtain more precise results.



Query Compression: Example

[... previous conversation about language training...]

Original Query

"What is the schedule of the course"

Derived Sub-queries

"What is the schedule of the C1 English course taught at the Puerto Real Campus during the first semester of the 2024/2025 academic year?"

Query Routing: Query Routing

1 Relevance

Ensures that the information retrieved is relevant for generating responses.

2 Efficiency

Directs the query to the most appropriate data sources, optimising resources.

3 Filtering

Can identify queries that the system should not respond to, improving security.

4 Hybrid Strategies

Combines keyword-based and semantic routing (using an LLM) for greater precision.

Query Routing: example

Original Query

"What cultural events are organised at the University of Cádiz?"

Router

The router decides to direct the query to a specific repository with data on events organised by the University of Cádiz, rather than searching the entire web

Query routing: system prompt

"Based on the user query, determine the most suitable data source(s)
to retrieve relevant information from the following options:
{{options}}
It is very important that your answer consists of either a single number
or multiple numbers separated by commas and nothing else!
User query: {{query}}"

Post-Retrieval Techniques

Post-Retrieval Techniques

Rerank

Reorganises the results to prioritise the most relevant ones.

Fusion

Combines results from multiple queries, resolving contradictions and creating coherent responses.

Summary

Synthesises and highlights the essential information, reducing token overload.

Rerank: Reordering the results

Similarity does not guarantee relevance: although semantic search already offers the results in order, it does not mean that the results it produces are the most relevant

"Lost in the middle": LLMs, like people, tend to focus on the beginning and end of texts.

Consider specific metadata: document popularity, source authority, temporal relevance, user preferences.

The aim is to introduce diversity in the results to avoid repetitions.

There are algorithms (TD-IDF, BM25) and specialised AI models for these tasks (Cohere, Jina, etc.)

Rerank: Example

Semantic retriever results

Reranking results

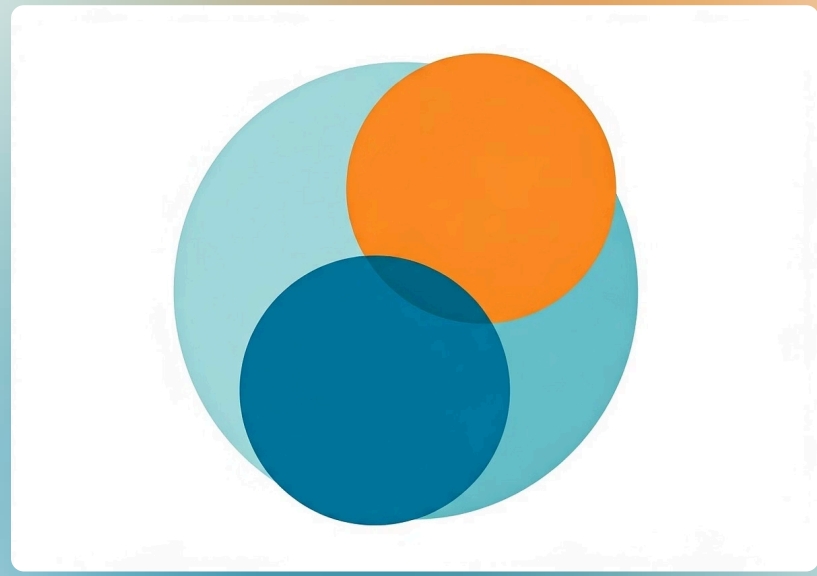
Fusion: Unifying Responses

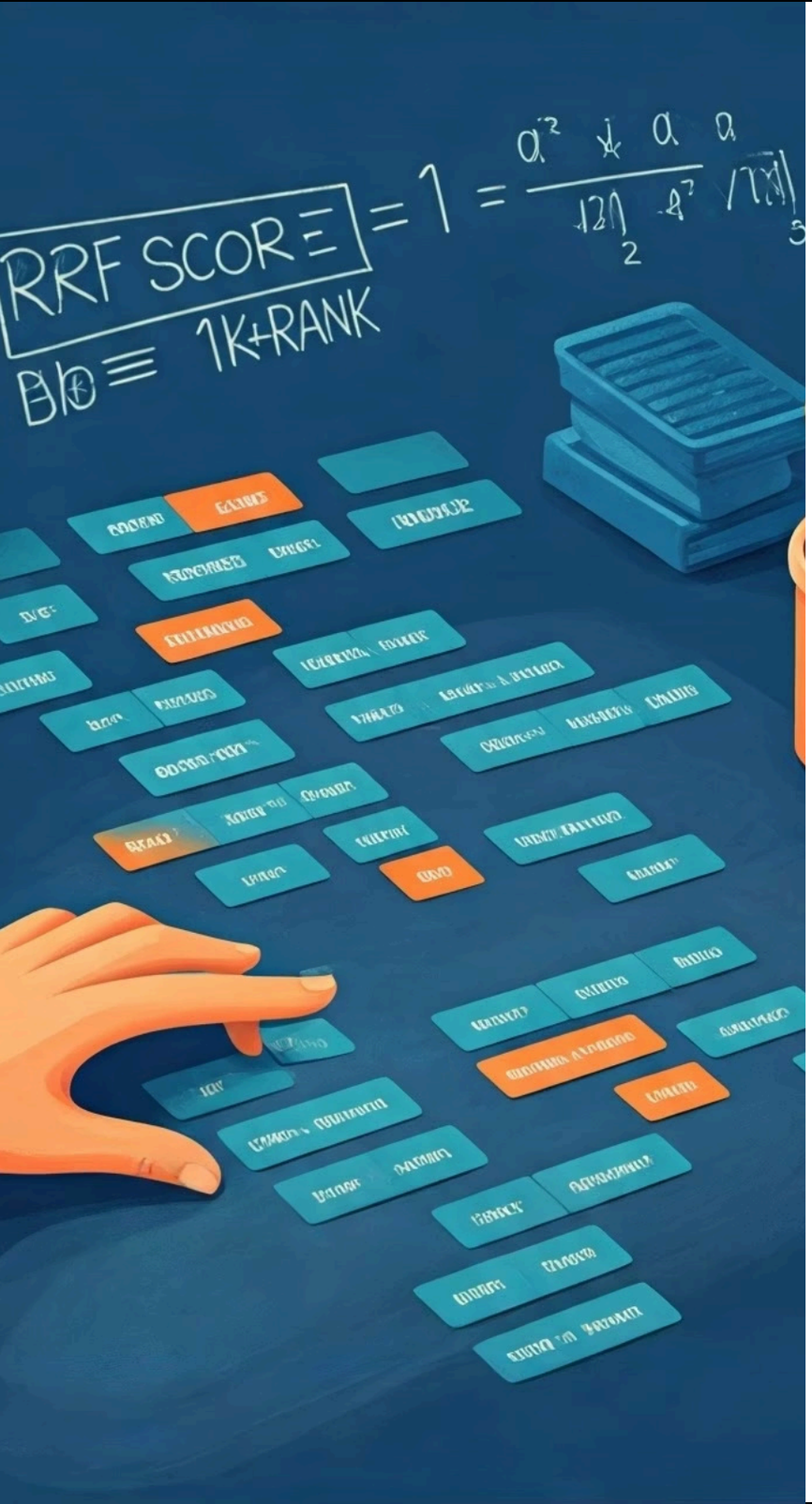
Combining Results

When using query expansion, we need to combine results to obtain complete and coherent answers.

Reciprocal Rank Fusion (RRF)

This technique assigns relevance scores to documents from different lists, then orders them to create a unified response.





Fusion: example

Semantic retriever results

	desarrolla software, aprende inteligencia artificial y redes."	
2	Documento G: "El Grado en Telecomunicaciones de la UCA abarca tecnología de redes, transmisión de datos y telecomunicaciones."	9.0
3	Documento H: "El Grado en Ingeniería Electrónica y Automatización ofrece formación en control de procesos industriales en la UCA."	8.5
4	Documento I: "La UCA imparte el Grado en Diseño Gráfico, con especial	6.1

	enfoca en redes y comunicaciones."	
2	Documento B: "El Grado en Ingeniería Electrónica Industrial en la UCA incluye automatización y control."	0.82
3	Documento C: "El Grado en Ingeniería Informática en la UCA cubre programación, redes y sistemas operativos."	0.80
4	Documento D: "El Grado en Diseño Gráfico en la UCA combina creatividad y	0.75

Fusion results

Documento	RRF Score	Ajuste final
Documento C / Documento F (Ingeniería Informática)	0.0323	**1er lugar**
Documento A / Documento G (Ingeniería Telecomunicaciones)	0.0325	2do lugar
Documento B / Documento H (Ingeniería Electrónica Industrial)	0.0320	3er lugar
Documento D / Documento I (Diseño Gráfico)	0.0312	4to lugar
Documento E / Documento J (Estudios Ingleses)	0.0306	5to lugar

Summary: Condensing contextual information

Distil, abstract and compact the retrieved texts, highlighting the key concepts and facts.

The aim is to avoid providing the LLM with superfluous or confusing information and reducing the number of tokens.

This helps the model to better understand the information and generate more accurate responses.

Summary: example

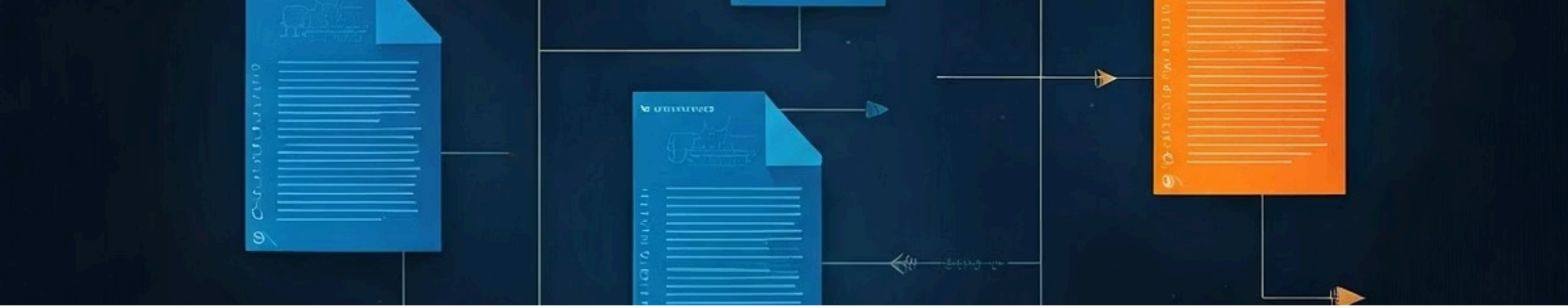
Results obtained

Entry to the LLM

Other advanced techniques

Other advanced techniques

- Summary embedding
- Parent document retrieval
- Multimodal RAG
- Hypothetical Questions Embedding
- Hypothetical Document Embedding
- Semantic chunking
- Contextual retrieval
- ...



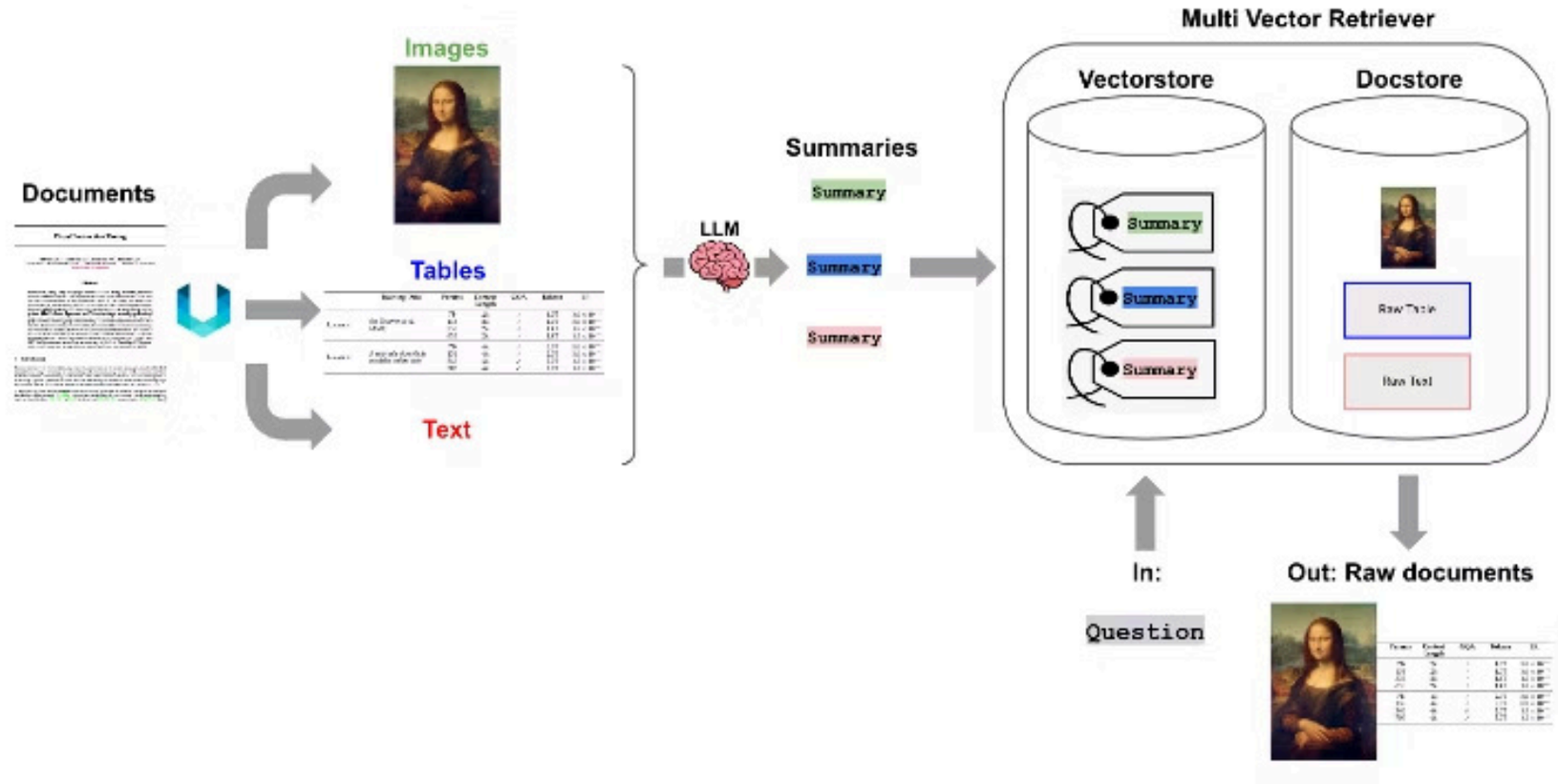
Summary Embedding + Parent

- When dividing documents for retrieval, there are often conflicting desires:
- The quality of **retrieval** improves when embeddings are generated from shorter texts. If they are too long, the embeddings may lose the full semantics of the text.
- The quality of **generation** improves when a long and detailed context is available, as this carries more semantic relationships and concepts. If we use short texts, the responses of the LLM may be less accurate due to lack of sufficient information.
- Documents, in addition to **text**, may include other types of content.

https://python.langchain.com/docs/how_to/parent_document_retriever/

Multi-modal RAG

Documents, in addition to text, can include images and tables, which can enrich the knowledge of the LLM.



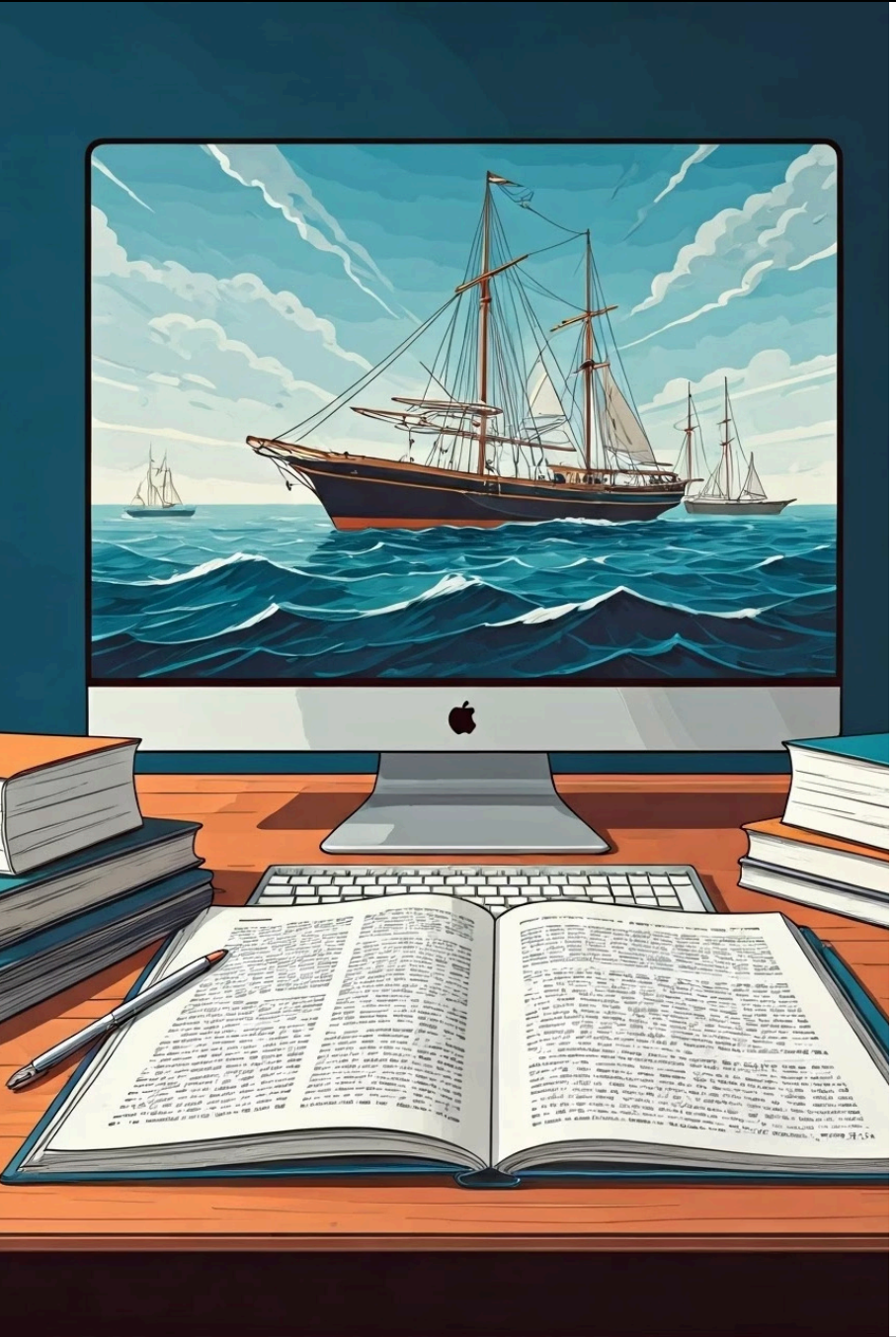
https://blog.langchain.dev/semi-structured-multi-modal-rag/

Hypothetical Document Embedding (HyDE)

Document similarity search can fail if the vector representation (embedding) of the query is not sufficiently similar to the relevant documents.

This technique generates a hypothetical document, which is then used to perform a similarity search in the knowledge base. This helps to find more relevant information and generate more accurate responses.

https://js.langchain.com/docs/integrations/retrievers/hyde/



Example of HyDE

Original Query

"How does the increase in ocean temperature affect the biodiversity of coral reefs?"

Generated Document

> "Global warming has caused a sustained increase in ocean temperatures, which negatively affects coral reef ecosystems. The rise in sea water temperature accelerates the phenomenon of coral bleaching, in which corals expel the symbiotic algae that nourish them, reducing biodiversity in these ecosystems. The loss of key species in the reefs has cascading effects that impact the entire marine food chain."

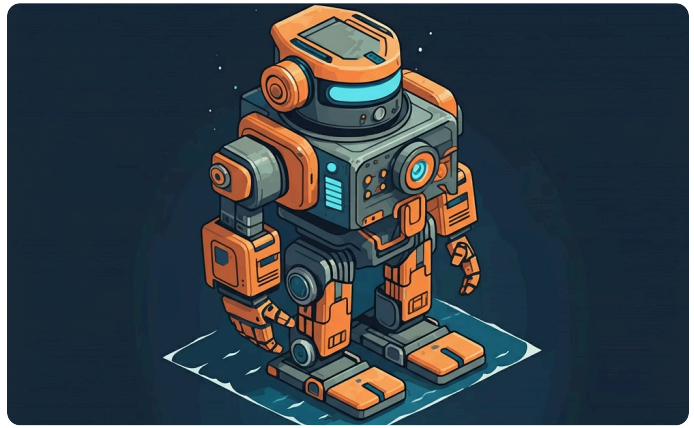
RAG modular

Modularity

This technique allows adding or replacing RAG components according to the specific needs of the task.

Adaptability

The modularity of RAG allows adapting the system to various tasks and contexts, offering greater flexibility and control.



RAG Patterns

In addition to the *Naive* and *Advanced* patterns, other patterns have emerged that allow increasing and refining the context in order to handle more complex tasks.

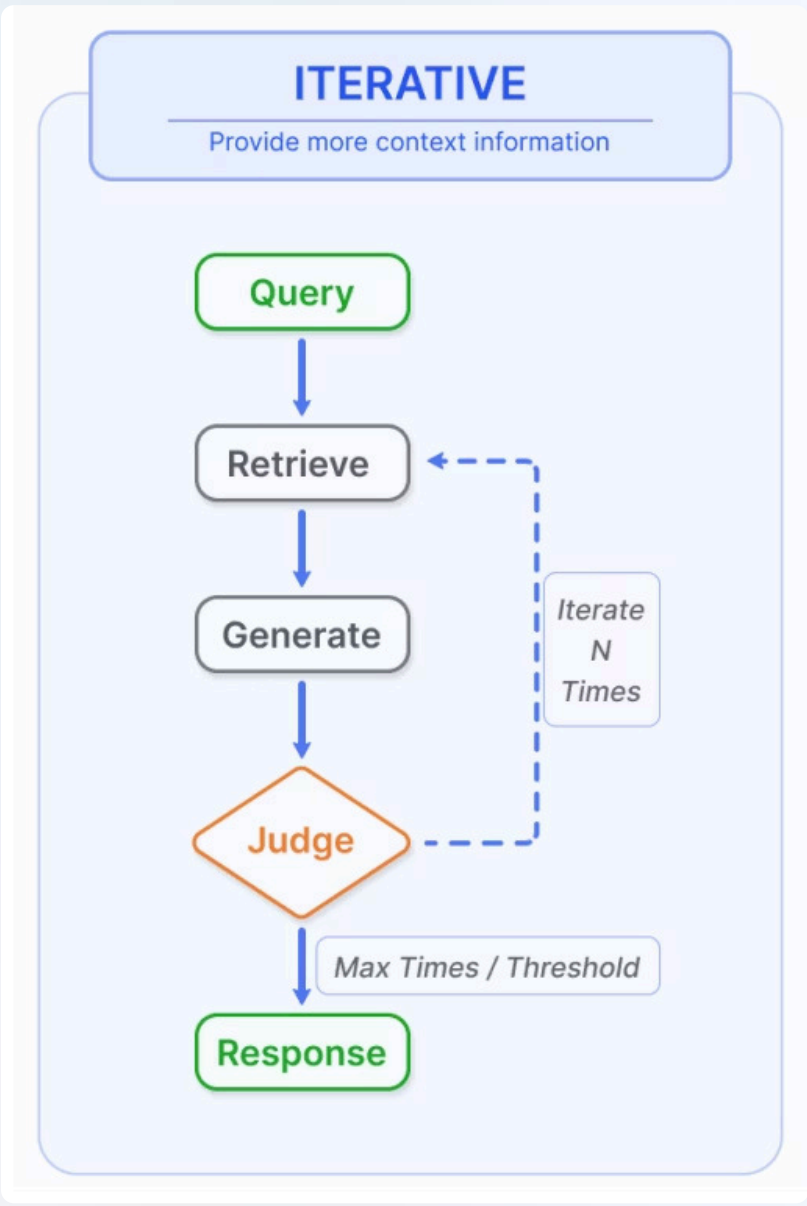
Multi-step augmentation: iterative

Query: "I would like information about the master's programmes offered at the ESI"
Retrieve: The system searches for and retrieves a list of available master's programmes.
Generate: "Master's in Cybersecurity, Master's Research in Engineering."
Judge: The response is incomplete, as details such as duration and admission requirements are missing.

Query: "What is the duration and admission requirements for the Master's in Cybersecurity?"
Retrieve: The system searches for and retrieves information about this master's programme.
Generate: "The Master's in Cybersecurity consists of 60 credits and requires a Degree in Computer Engineering"
Judge: The response is incomplete, as I still lack information about the Master's Research in Engineering.

Query: "What is the duration and admission requirements for the Master's Research in Engineering?"
Retrieve: The system searches for and retrieves information about this master's programme.
Generate: The Master's Research in Engineering consists of 60 credits and requires any Engineering Degree.
Judge: The response is now complete.

Response: "I will now provide you with the available master's programmes and the information I have about them..."



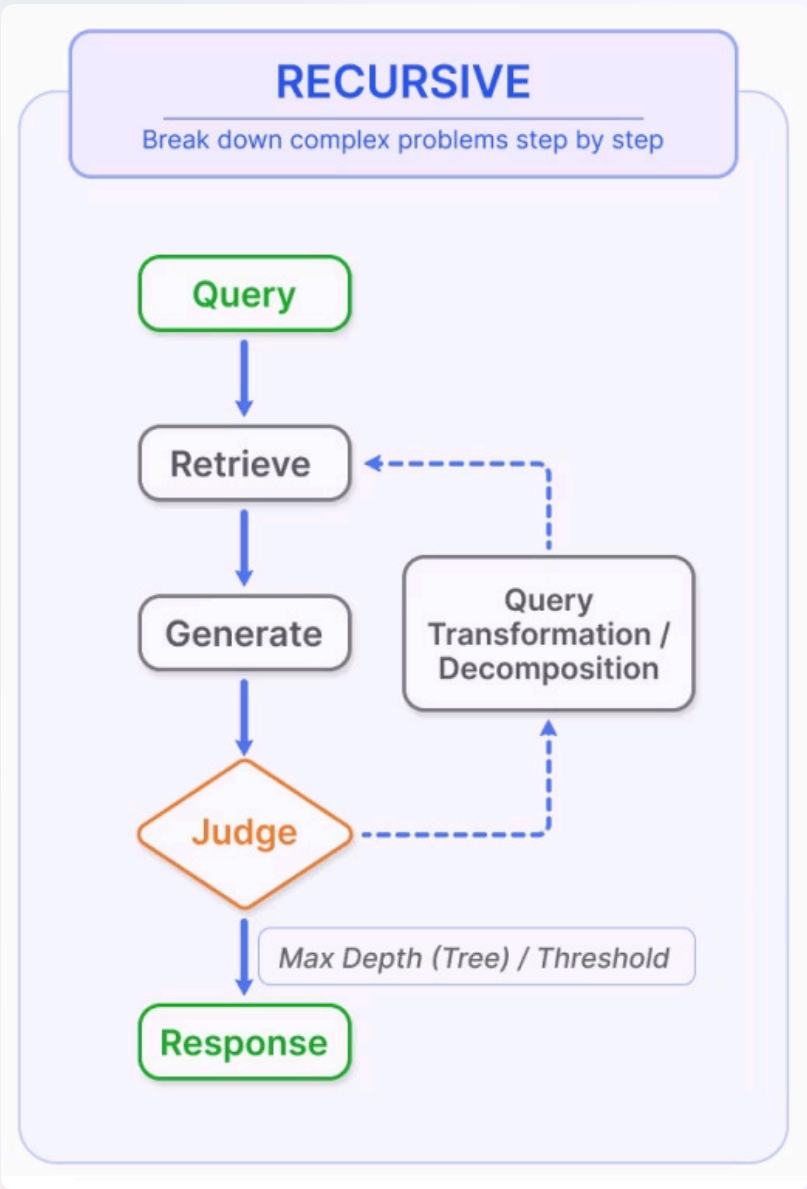
Multi-step augmentation: recursive

Query: "What research projects is the University of Cádiz (UCA) carrying out in the area of renewable energy?"
Retrieve: The system searches for and retrieves general information indicating that the UCA has a research project on offshore wind energy.
Generate: "The University of Cádiz is carrying out a research project on offshore wind energy."
Judge: The response is incomplete, as it lacks detailed information about this specific project.

Query: "What is the main objective of the University of Cádiz's offshore wind energy project?"
Retrieve: The system searches and finds the main objective of the project.
Generate: "The offshore wind energy project aims to study the feasibility of installing wind farms on the coast of Cádiz."
Judge: It is detected that more detail is still needed on the project's results.

Query: "What are the results obtained so far in the offshore wind energy project?"
Retrieve: The system finds the preliminary results of the project.
Generate: "The preliminary results show a high potential for the installation of wind farms in certain areas of the Cádiz coast."
Judge: The response is now complete and provides detailed information about the project.

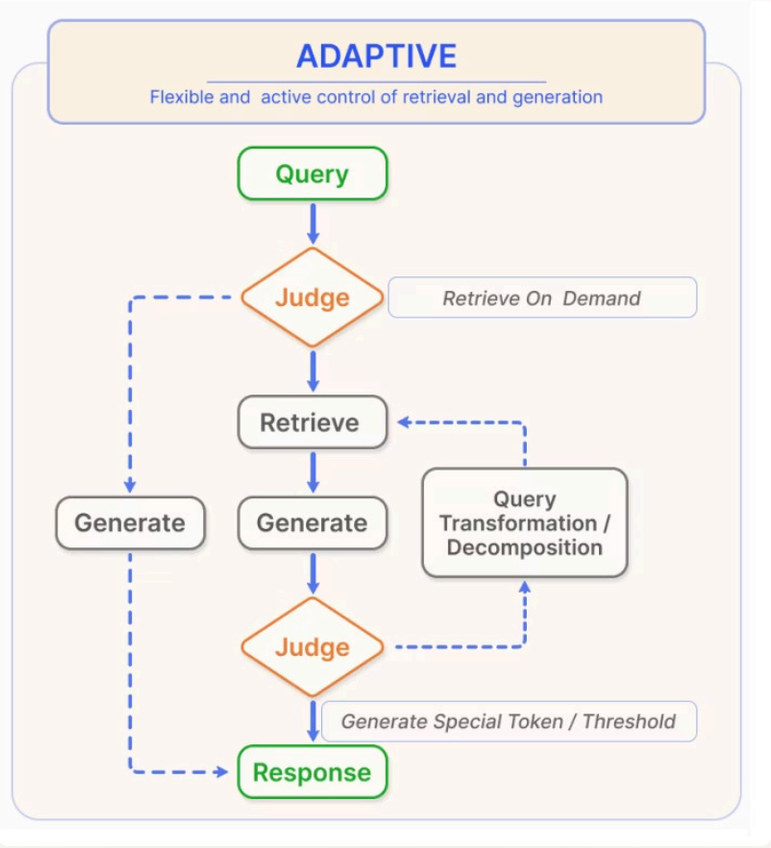
Response: "The University of Cádiz is carrying out a research project on offshore wind energy. The objective is... The preliminary results indicate..."



Multi-step augmentation: adaptive

Query: "What research projects is the UCA carrying out in the area of renewable energy?"
Retrieve: The system searches for and retrieves general information indicating that the UCA has a research project on offshore wind energy.
Generate: "The University of Cádiz is carrying out a research project on offshore wind energy."
Judge: The model wonders if it is necessary to obtain more details to adequately answer the original question. Given that the original question is general, it decides that it has enough information and does not perform any further searches. However, recognising that the user may want more details about these projects, the system prepares to offer additional information if the user requests it.

Response: "The University of Cádiz is carrying out a research project in the area of renewable energy, specifically on offshore wind energy. Would you like to know more details about any of these projects?"



Summary

This presentation explored the evolution of Retrieval-Augmented Generation (RAG) systems, moving beyond basic implementations to tackle common challenges and enhance performance. We delved into a variety of advanced techniques designed to improve both the retrieval of relevant information and the quality of the generated responses from Large Language Models.

Key advanced RAG techniques covered include:

- **Pre-Retrieval Techniques:** Enhancing the query before searching, such as query rewriting, expansion (e.g., Step-Back Question, Multi-Query), decomposition, compression, and routing.
- **Post-Retrieval Techniques:** Optimizing retrieved documents before generation, including reranking, fusion, and summarization of context.
- **Other Advanced Techniques:** Innovations like Summary Embedding + Parent, Multi-modal RAG for diverse document types, and Hypothetical Document Embedding (HyDE) to bridge semantic gaps.

By implementing these sophisticated approaches, RAG systems can achieve greater accuracy, relevance, and robustness in their responses.