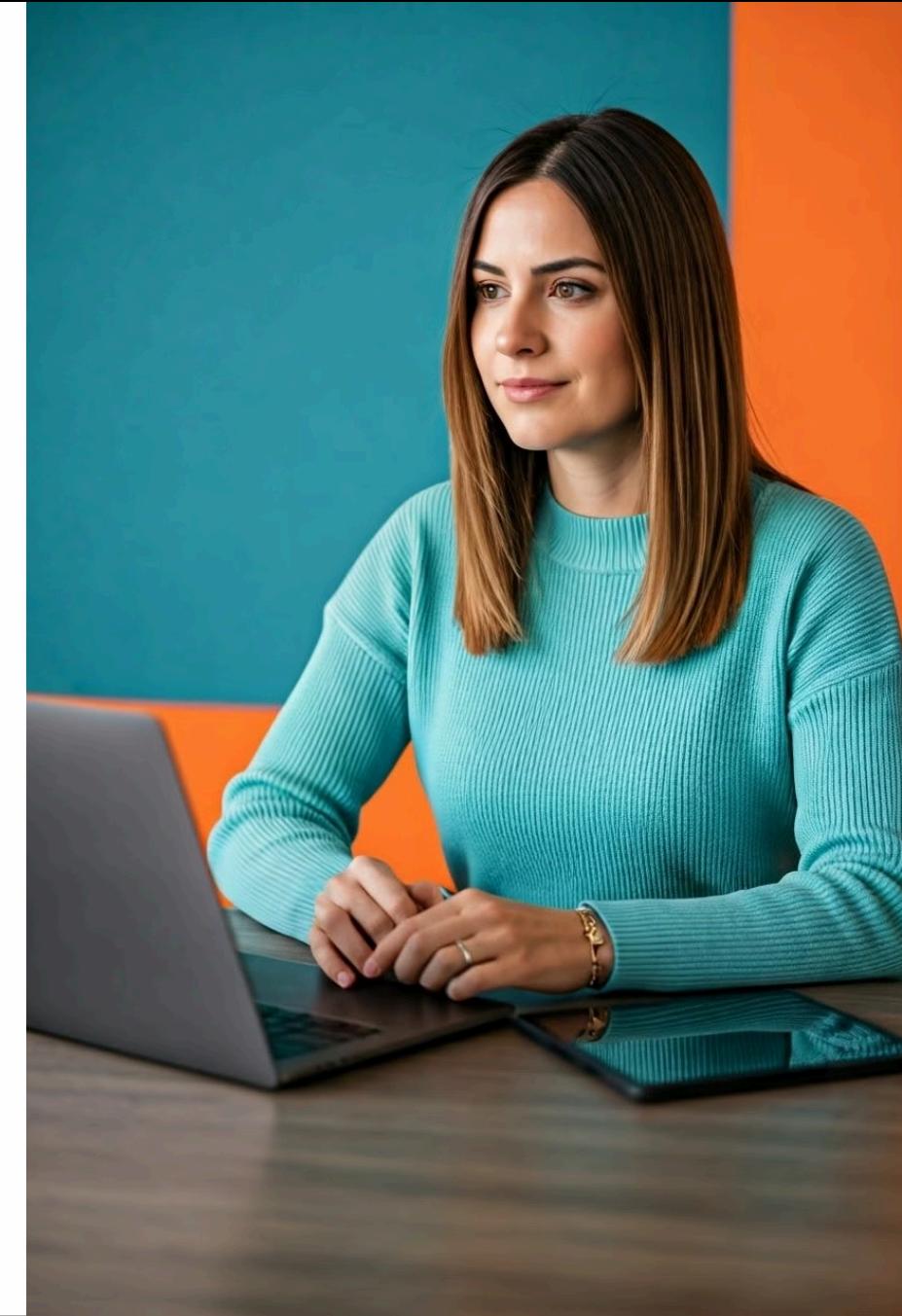


Retrieval Augmented Generation (RAG)



by Ivan Ruiz Rube



Contenidos

- Introducción
- Naive RAG
 - Indexación de información
 - Técnicas de recuperación
 - Generación de respuestas
- RAG avanzado
 - Técnicas de pre-recuperación
 - Técnicas de post-recuperación
 - Otras técnicas

Gemini



¿Cuándo exactamente serán las próximas jornadas CRUE-Digitalización a celebrar en Córdoba este otoño?

Mostrar versiones



¡Hola! Para obtener la información más actualizada sobre las próximas Jornadas CRUE-Digitalización en Córdoba este otoño, te recomiendo que consultes directamente las fuentes oficiales.

Aquí tienes algunas opciones:

- **Página web del evento:** La Universidad de Córdoba suele tener una página web específica para cada evento. Busca "49 Jornadas CRUE Digitalización" en el buscador de la universidad.
- **Sitio web de CRUE-Digitalización:** El Consorcio de Rectores Universitarios de España (CRUE) tiene una sección dedicada a sus eventos y jornadas. Allí podrás encontrar un calendario detallado.
- **Redes sociales:** Sigue las cuentas oficiales de CRUE-Digitalización y de la Universidad de Córdoba en redes sociales como Twitter, LinkedIn o Facebook. Suelen publicar actualizaciones y novedades sobre sus eventos.

Introducción

Calidad de la información: problemas

Información desactualizada

Los LLM tienen un conocimiento limitado del mundo. No pueden acceder a la información creada con posterioridad a su entrenamiento.

Información inaccesible

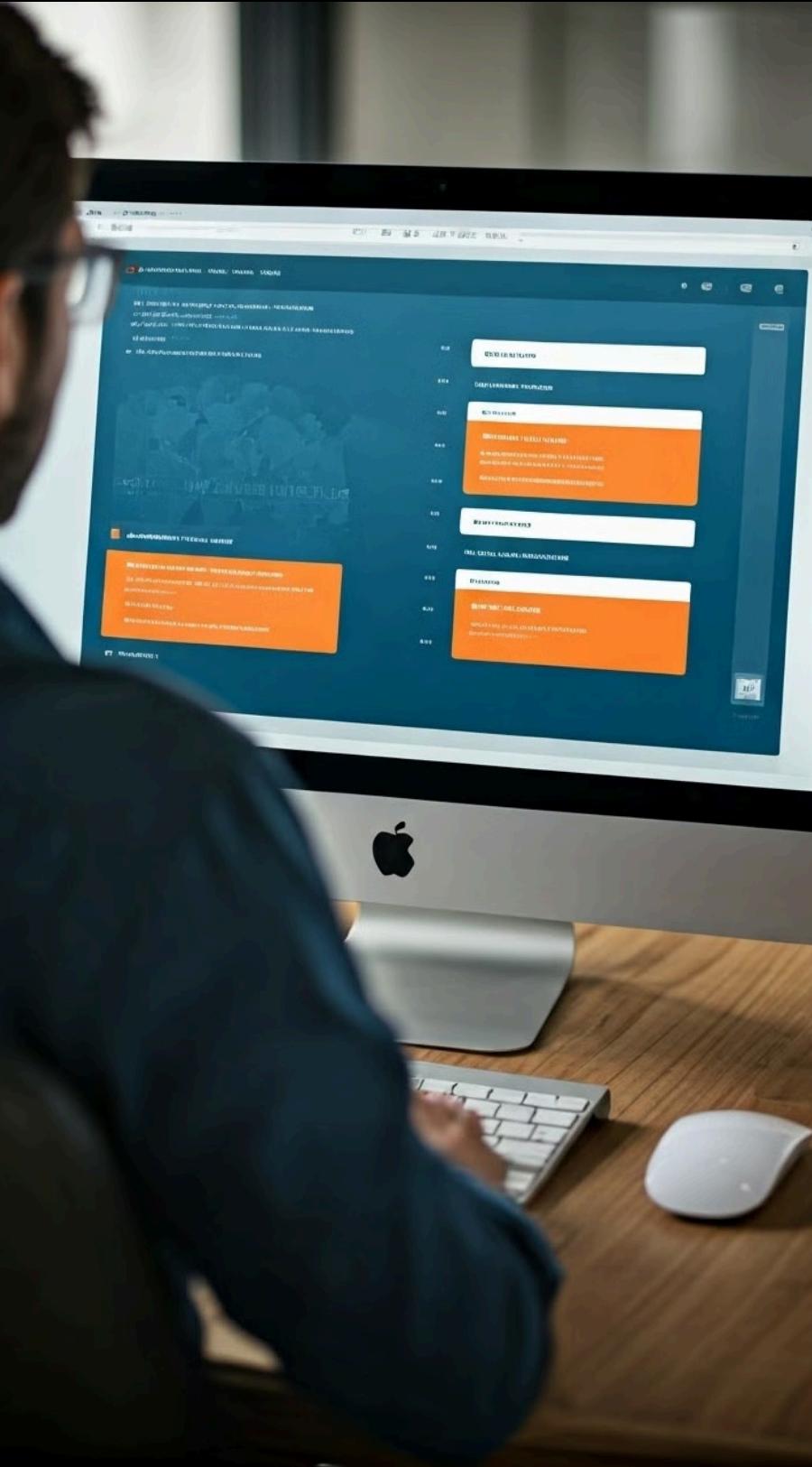
La información pública en la web que no fue capturada durante el proceso de extracción de datos previo a la generación del modelo también es inaccesible para los LLM.

Información privada

Los LLM no pueden acceder a la información protegida de las organizaciones, lo que limita su capacidad para brindar respuestas completas.

Alucinaciones

Los LLM pueden generar información incorrecta, inexacta o completamente inventada, pero que parece plausible.



Calidad de la información: remediación

Fine-tuning

El fine-tuning consiste en **ajustar** los **parámetros** de un modelo de lenguaje pre-entrenado para adaptarlo a un conjunto de datos específico.

Este método es ideal para mejorar la precisión del modelo en una tarea particular, como la generación de texto con **un estilo o tono específico**.

Retrieval-Augmented Generation

Técnica para reducir las alucinaciones de los LLM más **eficiente** que el fine tuning.

Consiste en aumentar el input que se le da al LLM, incorporando toda la base de conocimiento que necesite para poder dar una respuesta coherente.

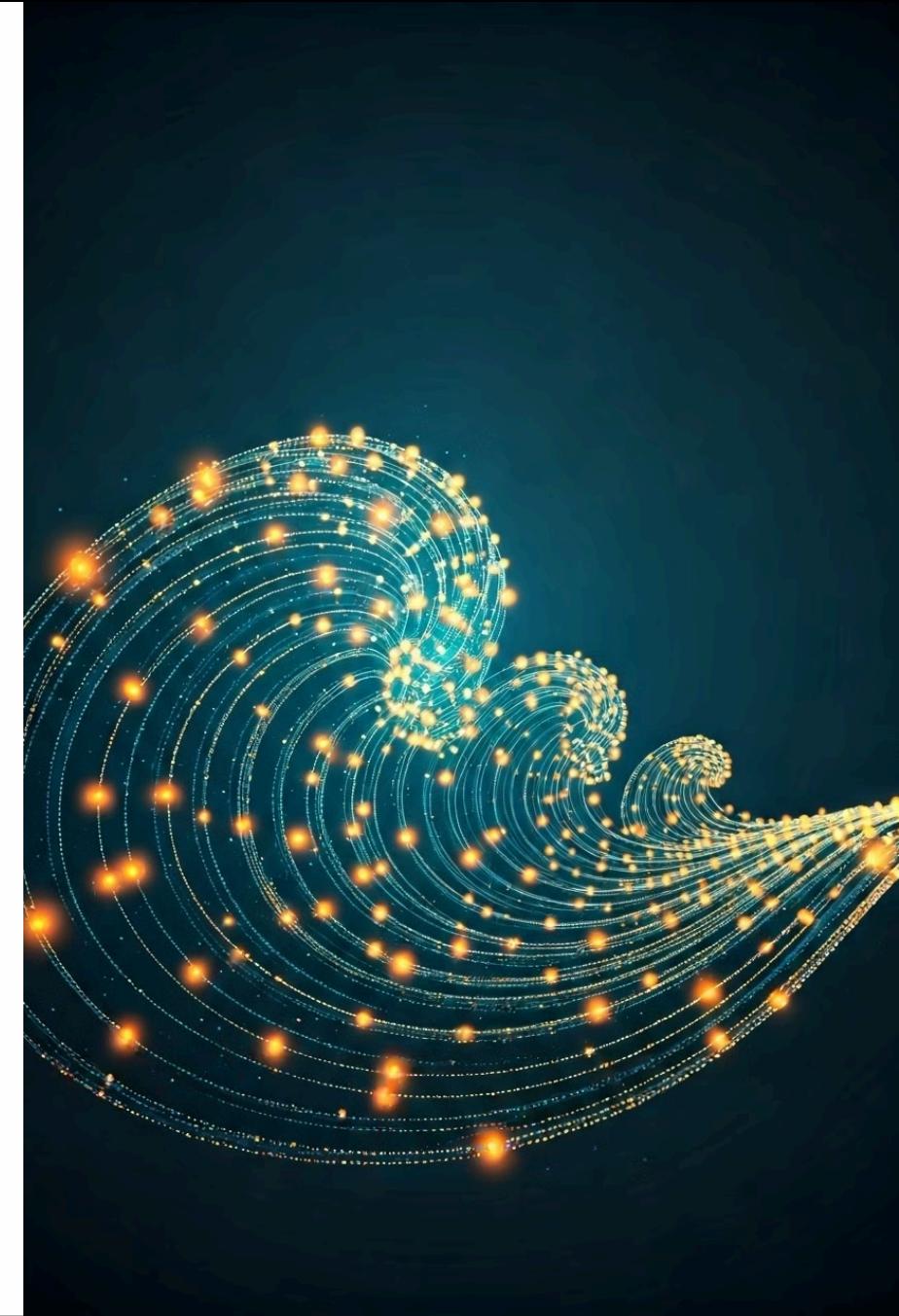
Retrieval-Augmented Generation (RAG)

Mejora la calidad

La generación de texto aumentada por recuperación busca mejorar la calidad de las respuestas de los modelos de lenguaje.

Base de datos de conocimiento

El modelo de lenguaje se conecta a una base de datos de conocimiento, recuperando información relevante para generar respuestas enriquecidas.



Es muy fácil hacer RAG..

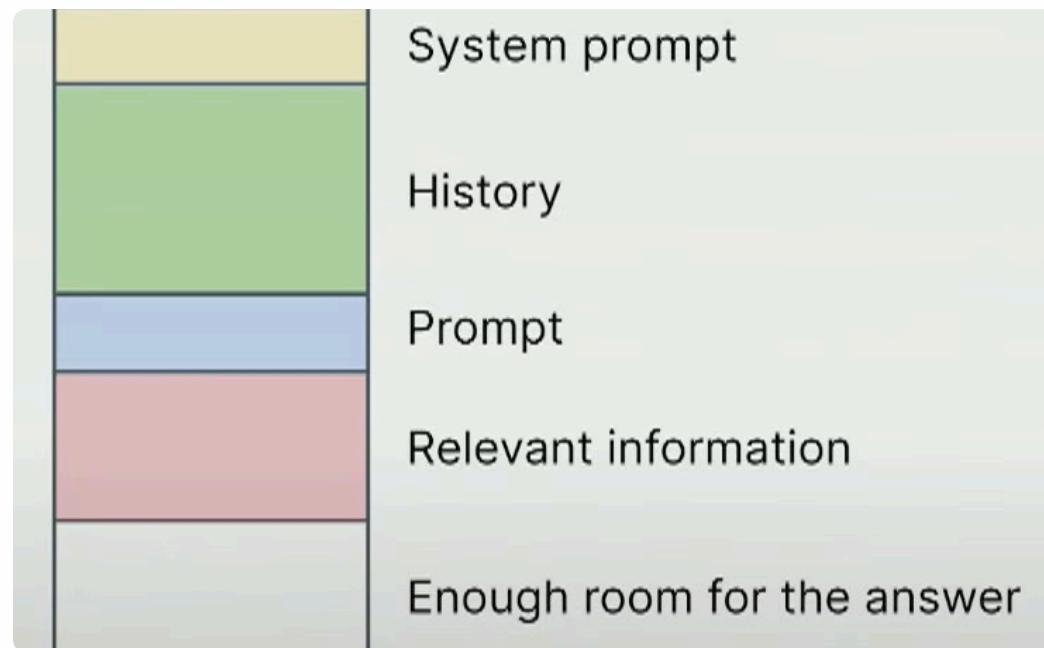
Ej 1. Creación de un asistente con EasyRAG



...pero conseguir buenos resultados no lo es tanto

Varios problemas que hay que abordar...

Desafíos al afrontar un sistema RAG



Ventana de contexto

No deberíamos incluir los documentos completos ya que la ventana de contexto no es ilimitada*.

Consumo de tokens

Los documentos largos pueden consumir muchos tokens, generando elevados gastos y/o mayores tiempos de respuesta



Información irrelevante

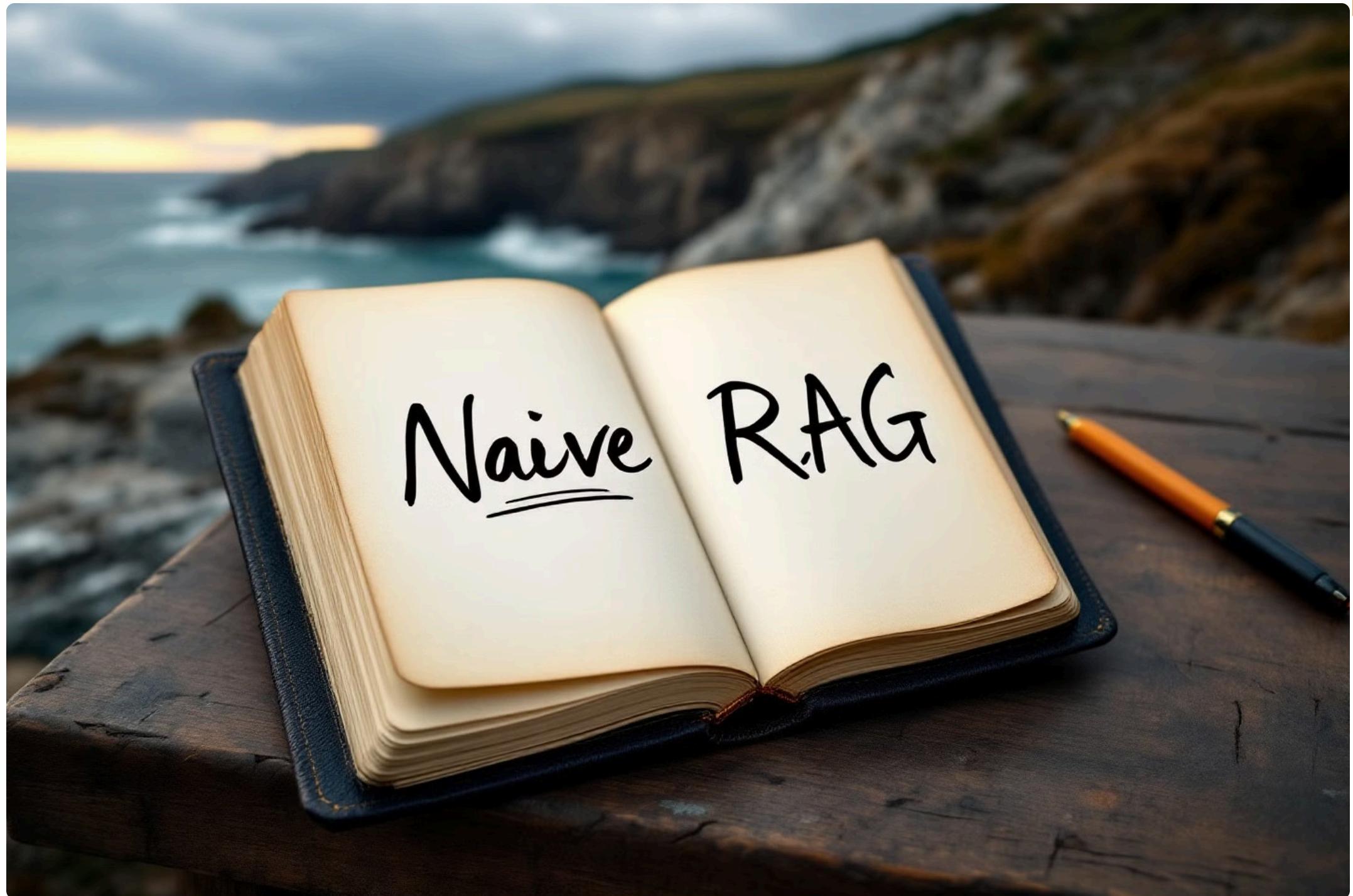
Incluir demasiada información en el prompt de entrada con información irrelevante puede reducir la precisión de las respuestas ofrecidas por el LLM.



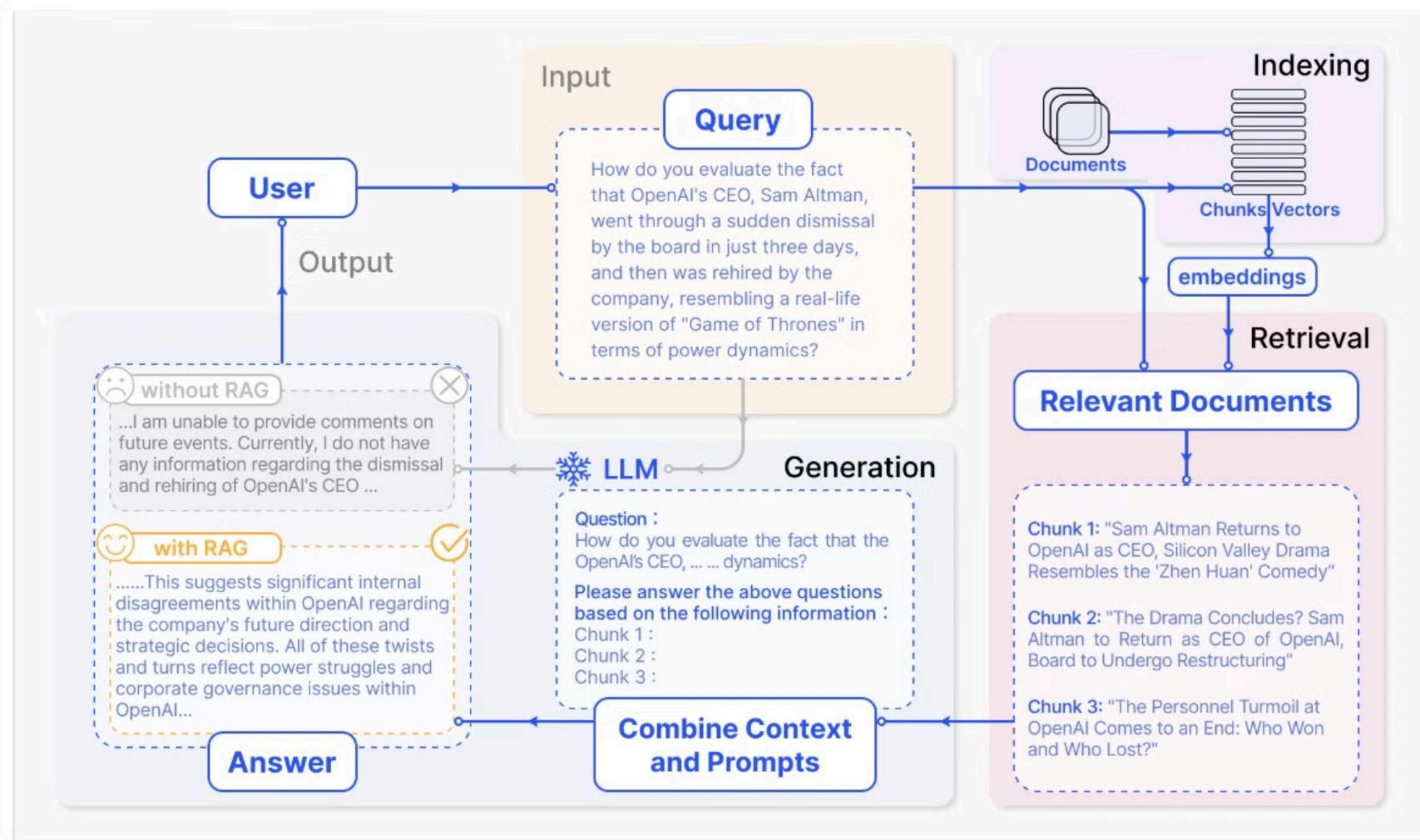
Complejidad de las preguntas

El usuario podría preguntar "*¿cómo se comparan los avances logrados en la UCA en el campo de la biotecnología en comparación con lo logrado por otras universidades, teniendo en cuenta las publicaciones en los últimos cinco años?*

¹ Imagen de @marchushellberg



Retrieval-Augmented Generation



Fases del RAG

Indexing

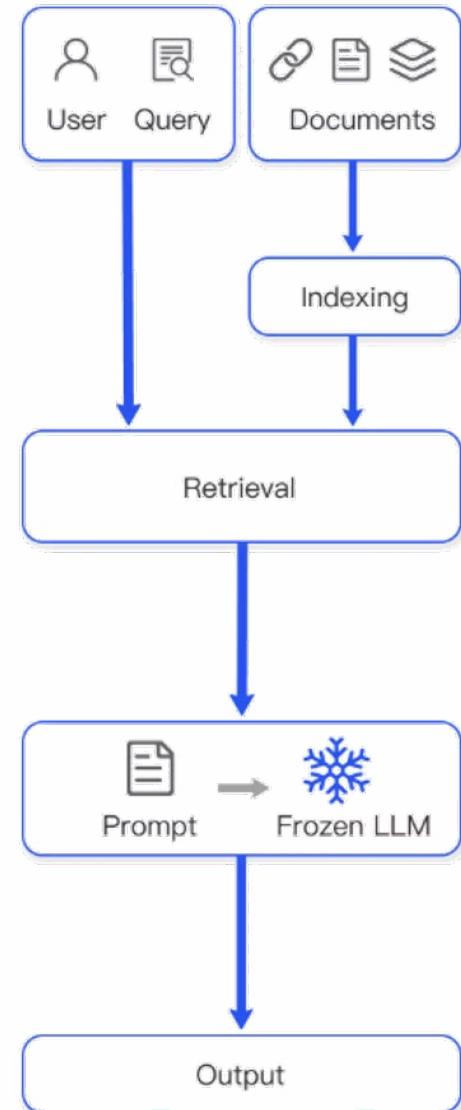
Pre-procesamiento *offline* de la información (documentos) para habilitar las fases siguientes

Retrieval

Recuperación de la información que se produce *online* cuando el usuario lanza la consulta al LLM

Generation

Respuesta sintetizada por el LLM como respuesta a la query y los textos recuperados



Naive RAG

Indexación de información

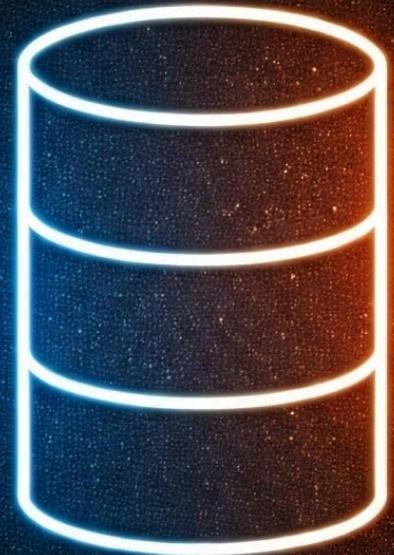
Indexación de información

Transformación

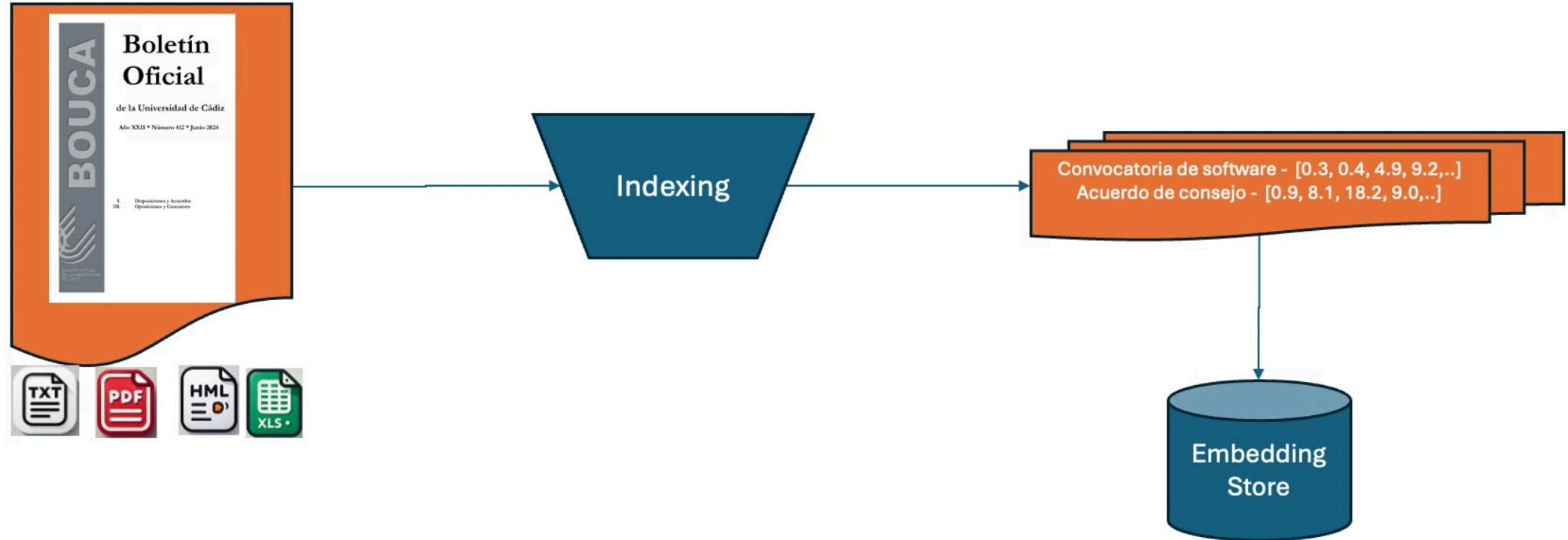
Convertir la información a un formato que permita realizar búsquedas eficientes.

Importancia

Este proceso es fundamental para la posterior recuperación y generación de texto.



Indexing (ejemplo)

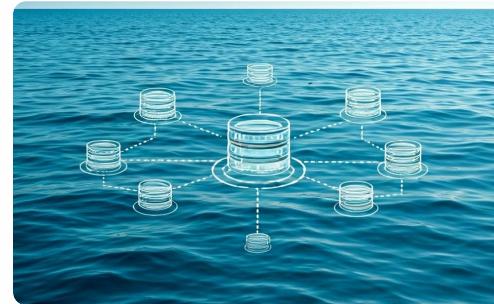


ETL

El proceso de indexación se puede dividir en etapas sucesivas que se integran en un pipeline de ETL (Extract, Transform, Load).

- 1 Los documentos se **cargan** desde fuentes externas.
- 2 Se **analizan** los documentos mediante algún parser.
- 3 Se **transforman** los documentos para limpiarlos o enriquecerlos con metadatos.
- 4 Los documentos se **dividen** en segmentos más pequeños.
- 5 Se generan vectores numéricos (**embeddings**) para cada segmento.
- 6 Los vectores se **almacenan** en una base de datos específica.

Loading



Sistemas de ficheros

Información recogida en ficheros almacenados en sistemas de archivos locales o en sistemas remotos (v.g. NextCloud o GDrive)

Sistemas de bases de datos

Información recogida en tablas en bases de datos relacionales (v.g Oracle) o en otros formatos de bases de datos no relacionales (v.g. Documentos en MongoDB)

Servicios de almacenamiento

Información recogida en sistemas de almacenamiento de objetos open source (v.g. MinIO) o propietarios (v.g. AWS S3 o Google Cloud Storage)

Gestores documentales

Sistemas de gestión documental open source (v.g. Alfresco) o propietarios (v.g. SharePoint)

Parsing

Los documentos deben analizarse y extraer sus datos con ayuda de un parser adecuado.



Información no estructurada

Ficheros con maquetación e imágenes (v.g. DOC, PPT o PDF)



Información semi-estructurada

Ficheros con algún tipo de etiquetado (v.g. JSON, XML o HTML)



Información estructurada

Bases de datos accesibles con SQL, hojas de cálculo o ficheros CSV





Transforming: cleaning

■ Eliminar información irrelevante

Los encabezados o pies de página no relevantes, marcas de agua o firmas digitales deben eliminarse para optimizar el contenido.

■ Extraer elementos visuales

Las tablas, imágenes o gráficos deberían ser procesados adecuadamente para generar texto.

■ Normalizar el texto

Se eliminan los espacios extra, caracteres extraños y se normaliza el formato para un análisis coherente.

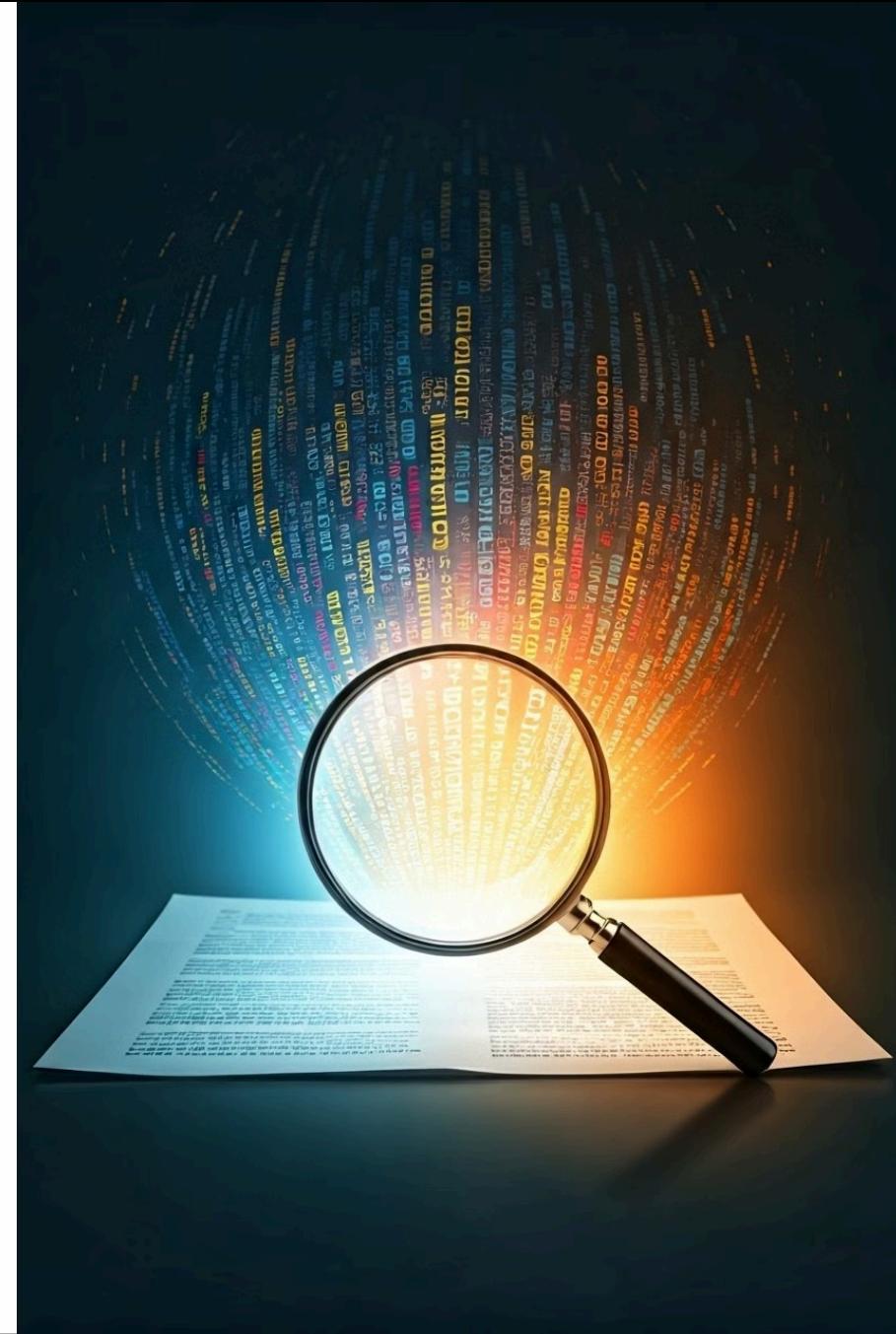
■ Eliminar marcado HTML

Para que el texto esté listo para su indexación, se eliminan etiquetas HTML.

Transforming: enriching

Enriquecer con metadatos es esencial para:

- Ofrecer contexto adicional al LLM.
- Optimizar las búsquedas con filtrado.
- Re-indexar la información tras actualizaciones de los documentos de origen.



Metadatos comunes

URL

Acceder al origen del documento y verificar su contenido.

Nombre del fichero y número de página

Identificar de forma única el documento y/o acceder a una sección específica.

Fecha de recuperación

Determinar la actualidad del documento.

Categoría

Facilitar la clasificación y búsqueda del documento por parte del LLM.

Autoría

Evaluar la confiabilidad del contenido.

Identificadores en origen

Localizar registros de información de manera eficiente en las fuentes de origen

Chunking (splitting)

- **Tamaño fijo:** división en segmentos de longitud fija de caracteres o tokens (300 aprox.).
- **Recursiva:** división en párrafos, líneas, espacios y, finalmente, en palabras.
- **Basada en documentos:** división por delimitadores estructurales del documento. Ejemplos:
 - **#** en Markdown
 - **<section>** en HTML
 - **\section** en LATEX
 - **class** en Python

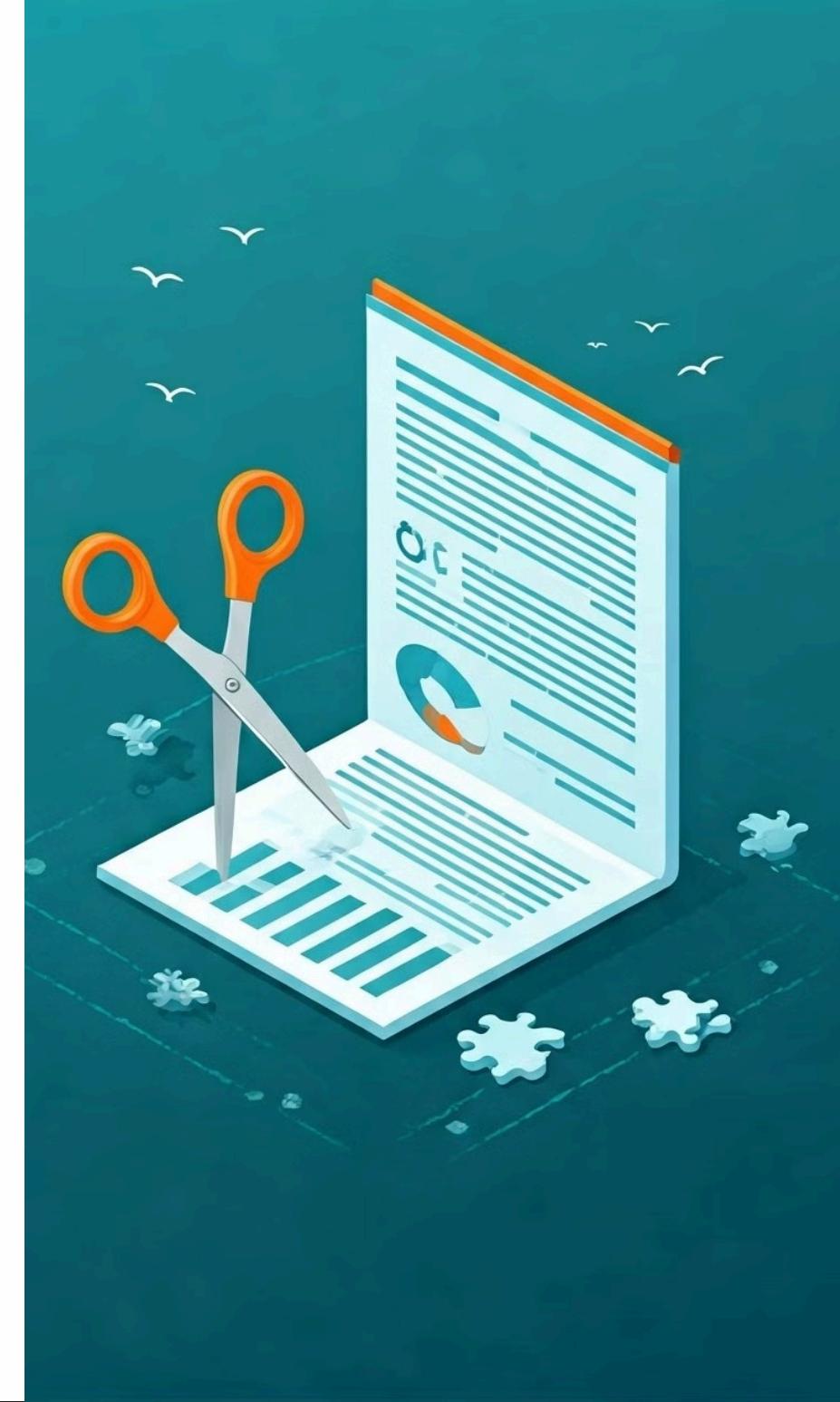
The screenshot shows a Streamlit application interface. At the top, there's a red header bar with the Streamlit logo. Below it, the title "Text Splitter Playground" is displayed. The main area contains a code editor with the following content:

```
Split a text into chunks using a Text Splitter. Parameters include:

- chunk_size: Max size of the resulting chunks (in either characters or tokens, as selected)
- chunk_overlap: Overlap between the resulting chunks (in either characters or tokens, as selected)
- length_function: How to measure lengths of chunks, examples are included for either characters or tokens
- The type of the text splitter, this largely controls the separators used to split on

Chunk Size: 1000    Chunk Overlap: 200    Length Function: Characters    Select a Text Splitter: RecursiveCharacterTextSplitter
```

Below the code editor is a preview area showing a file named "ui.png". The preview image displays a large orange pair of scissors cutting through a document on a laptop screen, symbolizing text splitting.



Indexing: embedding creation

Representación Semántica

Los embeddings capturan el significado del texto, representándolo como vectores numéricos de una determinada longitud (1536 floats).

Modelos de Embedding

Se utilizan modelos de aprendizaje automático específicos para generar los embeddings.

Word2Vec de Google

Word2Vec de Google fue uno de los primeros modelos de embeddings.

Tipos de tareas

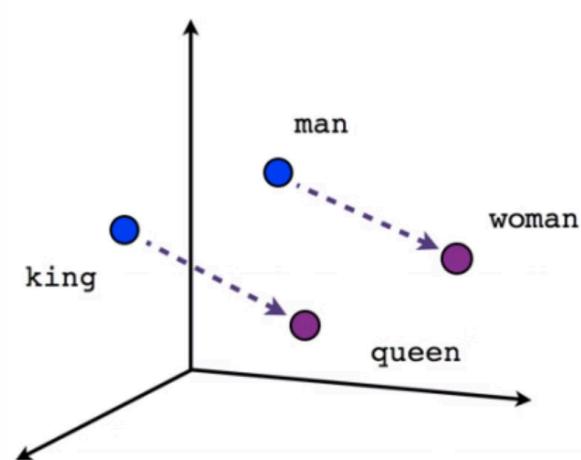
Además de para la recuperación de información, los embeddings pueden usarse para realizar tareas de clasificación o clustering.

Ejecución local

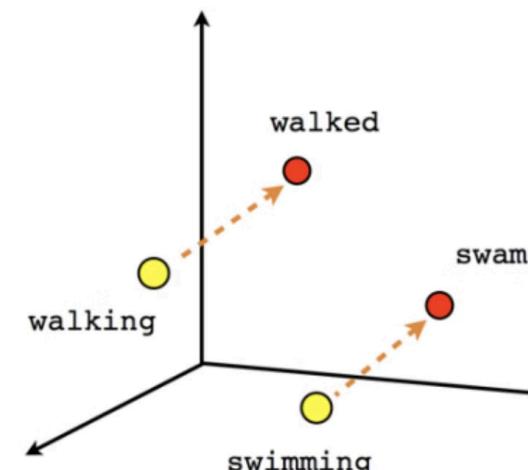
Hay modelos de embeddings que se pueden ejecutar localmente (ONNX Runtime o Jllama) en el mismo proceso de ejecución.

Ejecución remota

Podemos emplear APIs HTTP para acceder a modelos de embedding, como *text-embedding-3-small* de OpenAI.



Male-Female



Verb tense

Ranking de modelos de embeddings

Elegir el modelo de embedding más adecuado también es una decisión importante...



The screenshot shows the MTEB Leaderboard interface. On the left, there's a blue sidebar with the text "MTEB Leaderboard" and a gold medal icon. The main area has a light gray background. At the top right, there's a "huggingface" logo (a yellow smiley face) followed by the text "huggingface". Below that is the title "MTEB Leaderboard - a Hugging Face Space by mteb". Under the title, there's a subtitle "Discover amazing ML apps made by the community". In the top right corner of the main area, there's a small blue square icon with a white arrow pointing outwards.

Indexing: storing embeddings



Almacenamiento

Los embeddings generados deben almacenarse junto con los segmentos de texto correspondientes para permitir la recuperación de información.



Vector stores

Los vector stores son bases de datos especializadas para almacenar vectores de alta dimensionalidad, como embeddings.



Búsqueda por similitud

Estos sistemas permiten realizar búsquedas basadas en la similitud entre vectores.

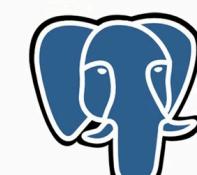


Integración

Los vector stores ofrecen APIs para su integración con diferentes lenguajes de programación, lo que facilita su uso en aplicaciones de IA.



Chroma



PostgreSQL



Comparativa de BD vectoriales

Y elegir la BD adecuada...



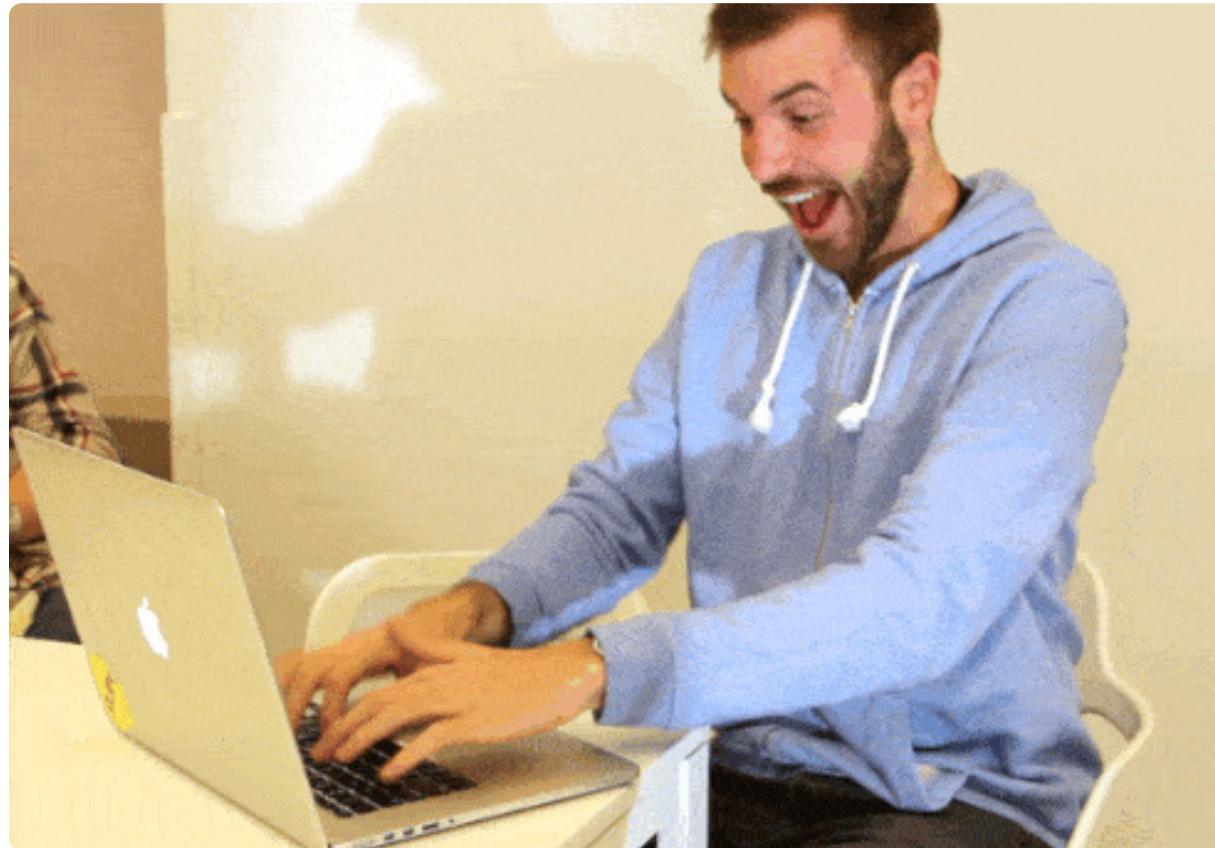
 superlinked.com

Vector DB Comparison

Vector DB Comparison is a free and open source tool from VectorHub to compare vector databases.



Ej 2. Uso de un embedding store



Base de datos vectorial Milvus

News

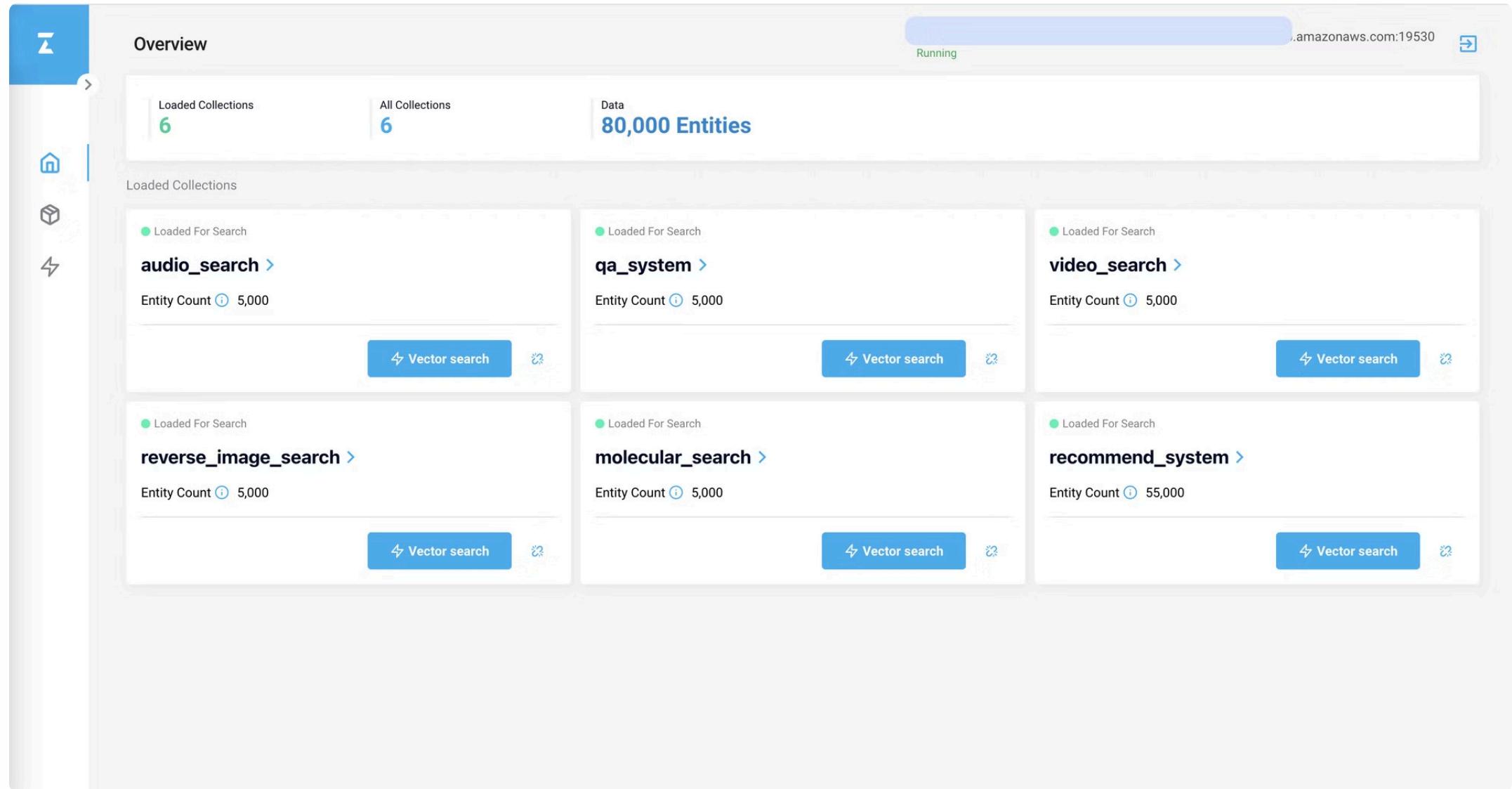
Unveiling Milvus 2.4: Multi-vector Search, Sparse Vector, CAGRA Index, and More!

The High-Performance Vector Database Built for Scale

Milvus is an open-source vector database built for GenAI applications. Install with pip, perform high-speed searches, and scale to tens of billions of vectors with minimal performance loss.

Get Started

Attu: cliente gráfico de Milvus



The screenshot shows the Attu interface for managing Milvus collections. At the top, there's a header with a 'Z' icon, the word 'Overview', and a status bar indicating 'Running' with the IP 'amazonaws.com:19530'. Below the header, it displays 'Loaded Collections' (6) and 'All Collections' (6), with a total data count of '80,000 Entities'.

The main area lists six collections, each with a status indicator (green dot for 'Loaded For Search'), the collection name, entity count (all 5,000), and a 'Vector search' button:

- audio_search >** Entity Count 5,000
- qa_system >** Entity Count 5,000
- video_search >** Entity Count 5,000
- reverse_image_search >** Entity Count 5,000
- molecular_search >** Entity Count 5,000
- recommend_system >** Entity Count 55,000

Instalación de Milvus y Attu

Descarga del script para Docker

```
curl -sfL https://raw.githubusercontent.com/milvus-io/milvus/master/scripts/standalone_embed.sh -o standalone_embed.sh
```

Añadir fichero user.yaml

```
# Extra config to override default milvus.yaml
common:
  security:
    authorizationEnabled: true
```

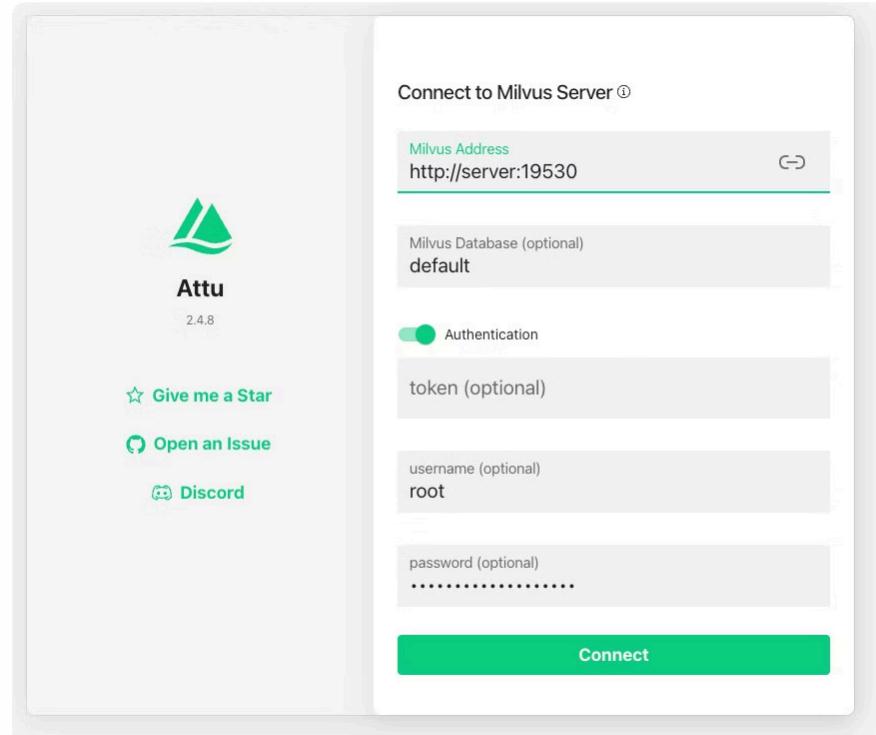
Arrancamos servidor Milvus

```
bash standalone_embed.sh start
```

Descargamos imagen de Attu y la arrancamos

```
docker run -d -p 8000:3000 -e MILVUS_URL=http://SERVER:19530 zilliz/attu:v2.4.8
```

Abrimos navegador y accedemos a <http://SERVER:8000> con usuario root y password Milvus



Recuperación de la información



Fuentes de datos



Bases de Datos Vectoriales

Realizar búsquedas semánticas con filtros.



Web

Motores de búsqueda como Google, Bing o DuckDuckGo.



Bases de Datos Relacionales

Acceso mediante consultas SQL a datos recogidos en motores (Oracle, PostgreSQL).



Bases de Datos NoSQL

Consultas en bases de datos NoSQL, como Neo4J, mediante lenguaje Cypher.



Servidores de búsqueda

Motores de búsqueda con capacidades de almacenamiento, como Solr o ElasticSearch.



Servicios Web

API HTTP desde donde obtener información actualizada.

Recuperación de la información

La recuperación de información es una parte fundamental del proceso de RAG, ya que permite acceder a los datos relevantes para la generación de información.

La calidad de la información recuperada afecta directamente la precisión y la calidad del texto generado.

Búsqueda SQL

- Con el lenguaje SQL se pueden realizar búsquedas en campos cadena de caracteres mediante operadores como LIKE o REGEXP_LIKE
- Se buscan coincidencias exactas o parciales de palabras clave dentro de los documentos, según los patrones definidos.



```
SELECT * FROM productos WHERE descripcion LIKE '%camara%';
```

```
SELECT * FROM productos WHERE REGEXP_LIKE(descripcion, 'camara|telefono');
```

```
SELECT * FROM productos WHERE CONTAINS(descripcion, 'camara') > 0;
```

Búsqueda full-text

Las búsquedas de texto completo buscan la aparición de palabras clave en los documentos.

Estos sistemas asignan una puntuación a los documentos según la frecuencia y relevancia (algoritmo TF-IDF) de las palabras clave en la consulta.

La puntuación se basa en factores como la frecuencia de aparición, la fecha de creación, la proximidad de las palabras clave entre sí, geolocalización, etc.

Tienen limitaciones al trabajar con consultas en lenguaje natural debido a la dificultad de identificar sinónimos, paráfrasis o contexto.





Búsqueda semántica en vector stores

■ Embeddings

Los documentos se convierten en vectores numéricos que representan su significado.

■ Vector Stores

Permiten búsquedas rápidas y eficientes en el espacio vectorial, basados en la similitud de vectores.

■ Distancia de Similitud

Los documentos se ordenan en función de su similitud de sus embedding con respecto al embedding de la query de entrada, utilizando la distancia angular (similaridad del coseno).

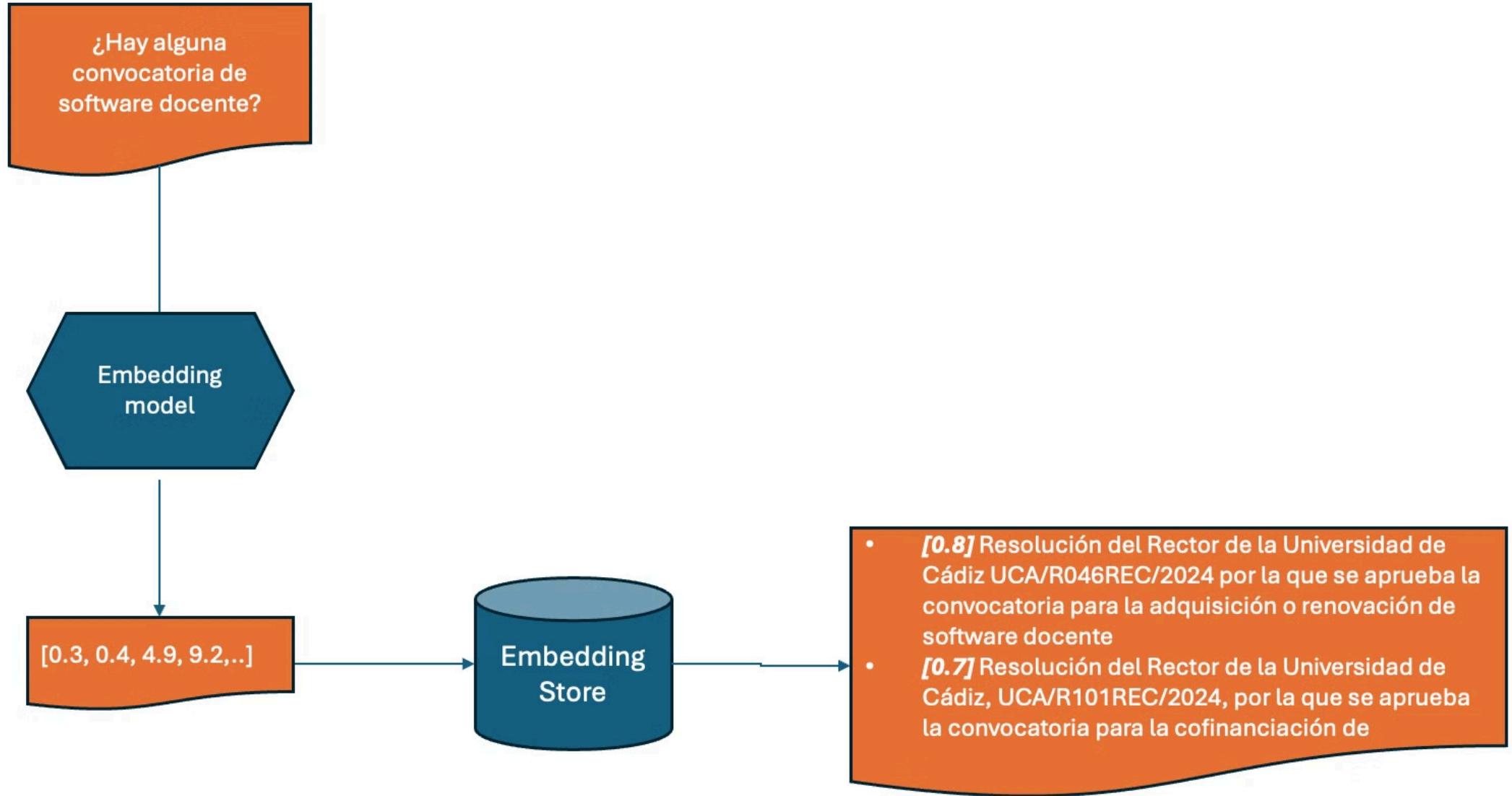
■ Diferentes modelos de embeddings producen diferentes salidas

No comparar embeddings generado con un determinado modelo con los embeddings generados por otros.

■ Obtención de Resultados

Se obtienen los k chunks más similares a la query, proporcionando un conjunto de información relevante.

Retrieval (ejemplo)



Ej 3. Creación de un buscador semántico en base de datos vectorial



Búsquedas semánticas combinadas con filtros

Consulta del Usuario

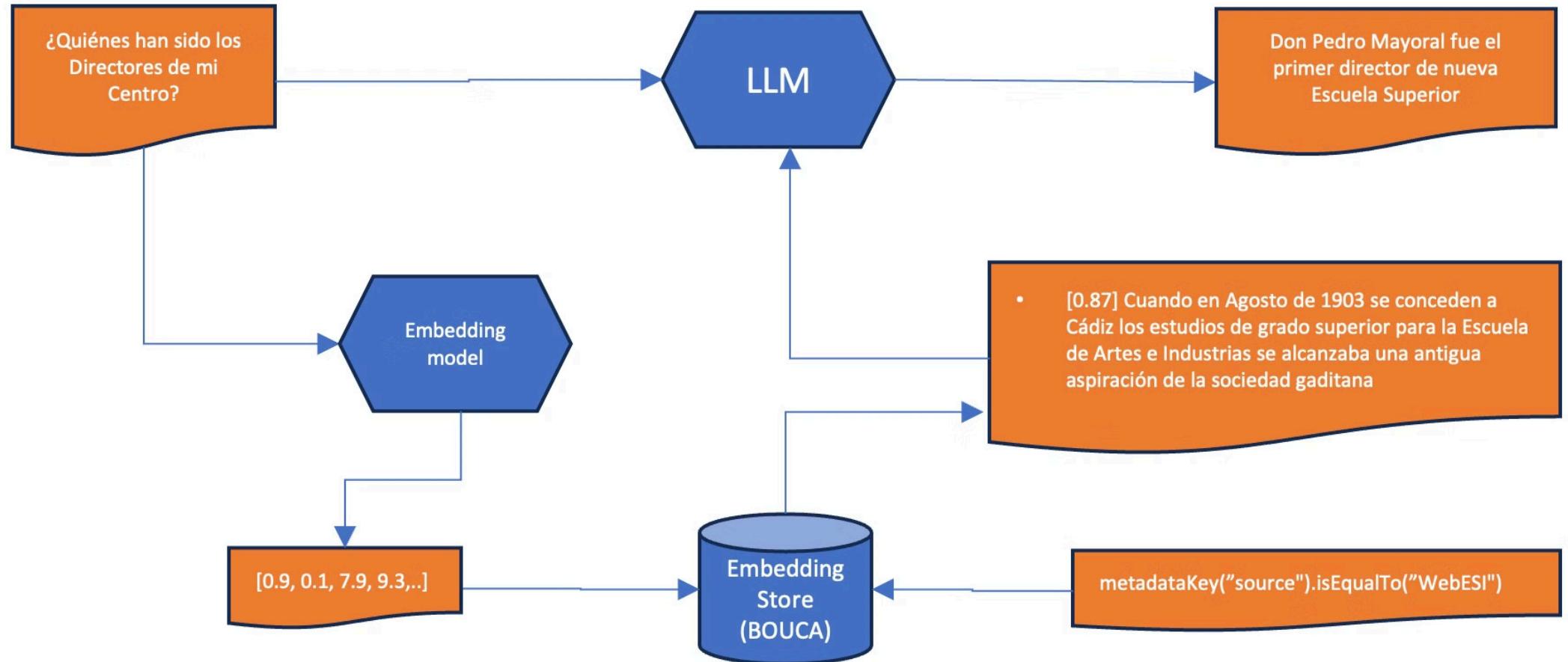
Los metadatos que acompañan a los segmentos de texto y sus embeddings, habilitan la posibilidad de aplicar filtros adicionales

Filtros de metadatos

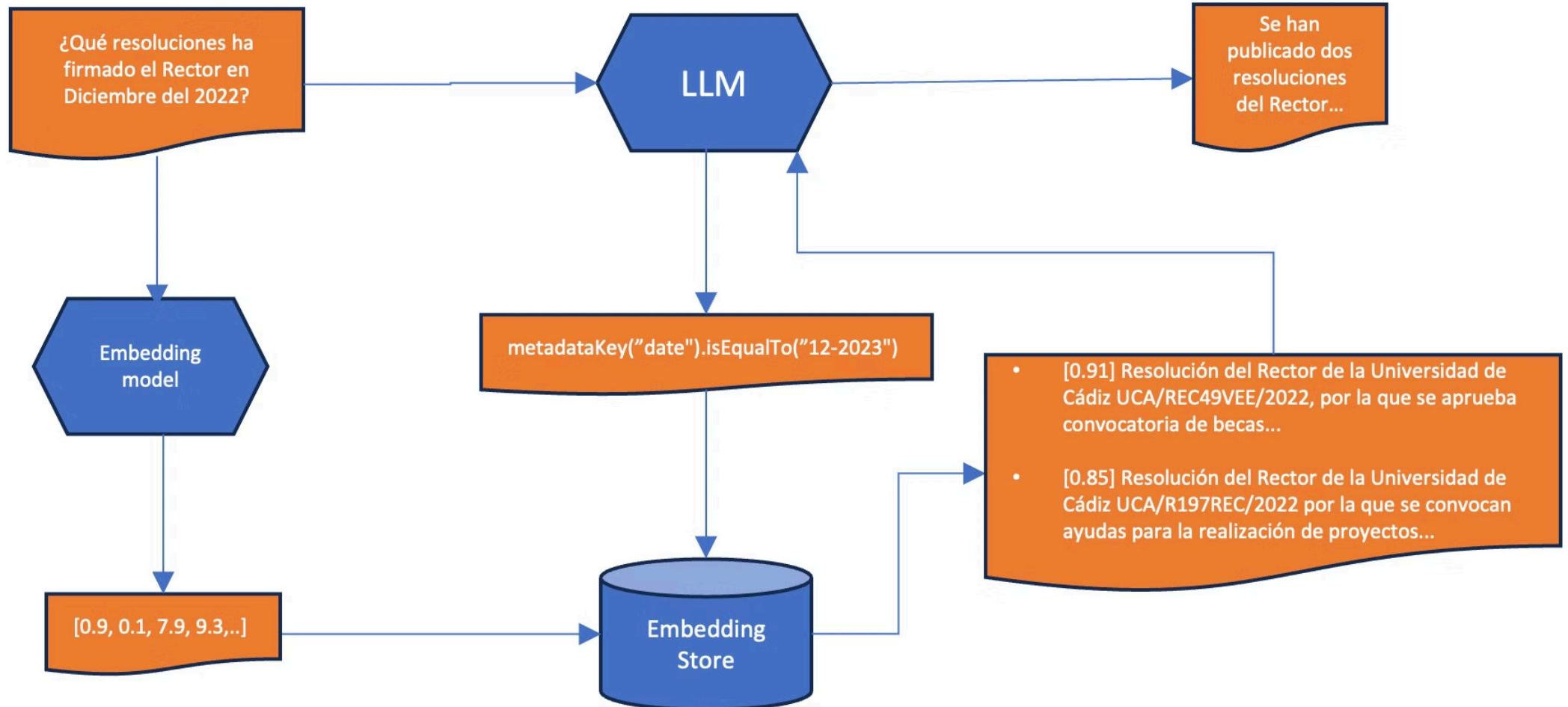
Los filtros se pueden definir de forma estática o dinámica (v.g. según permisos del usuario) o ser generados por un LLM



Filtros de metadatos

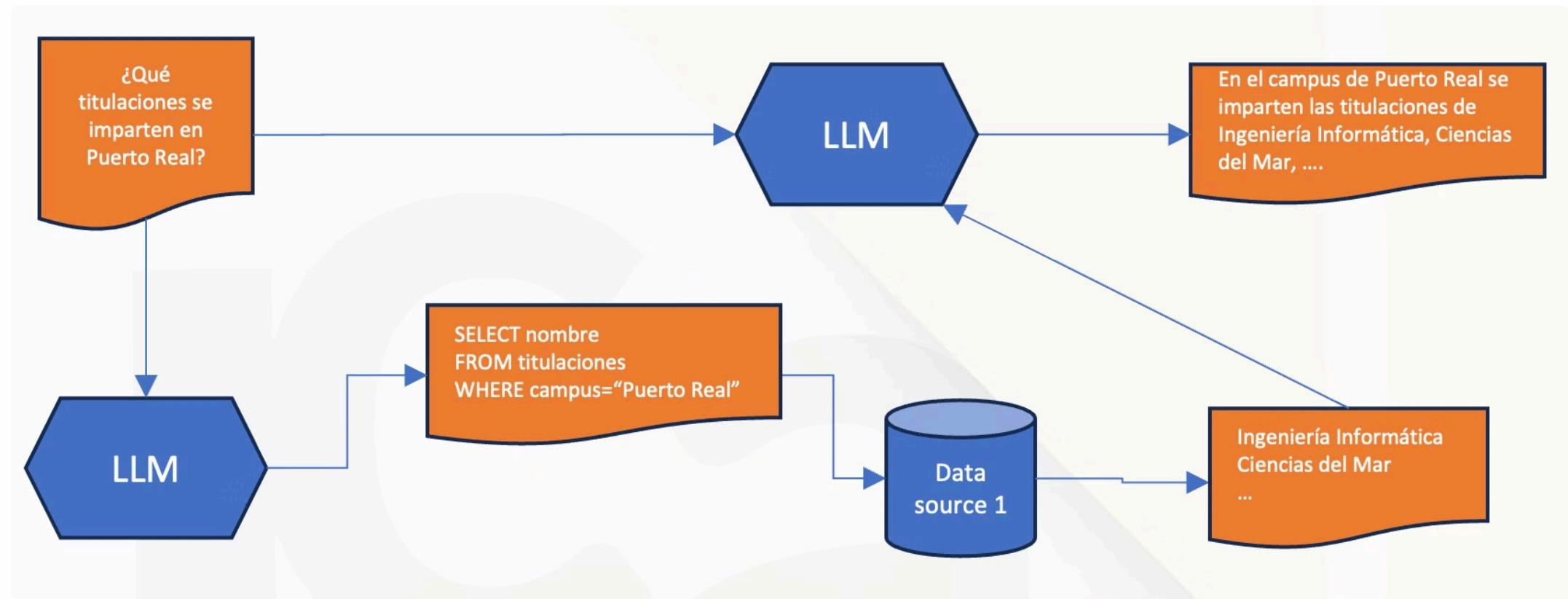


Filtros generados (Self-Querying)



Búsqueda semántica en BD Relacionales

El usuario realiza una consulta en lenguaje natural que un LLM traduce a una sentencia SQL válida.



Búsqueda semántica en BD Relacionales: system prompt

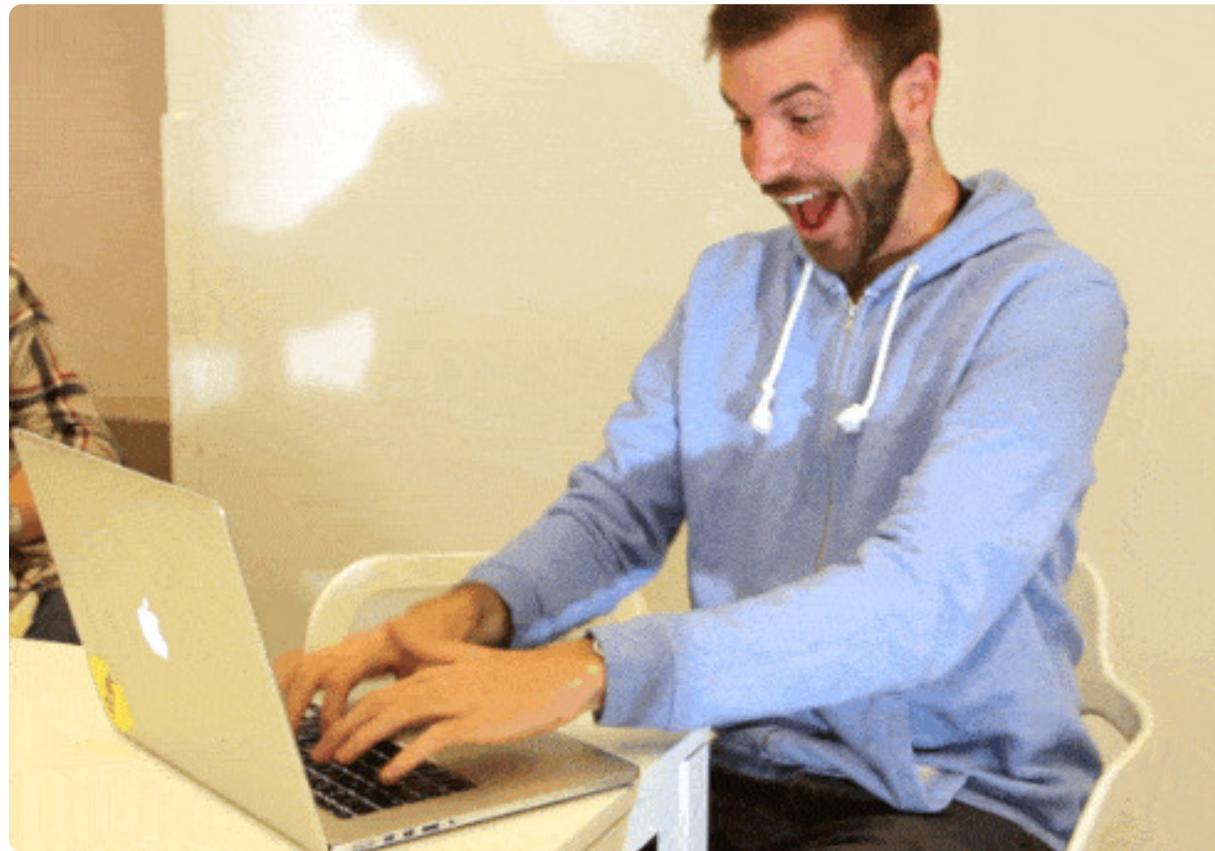
"You are an expert in writing SQL queries.

You have access to a {{sqlDialect}} database with the following structure:

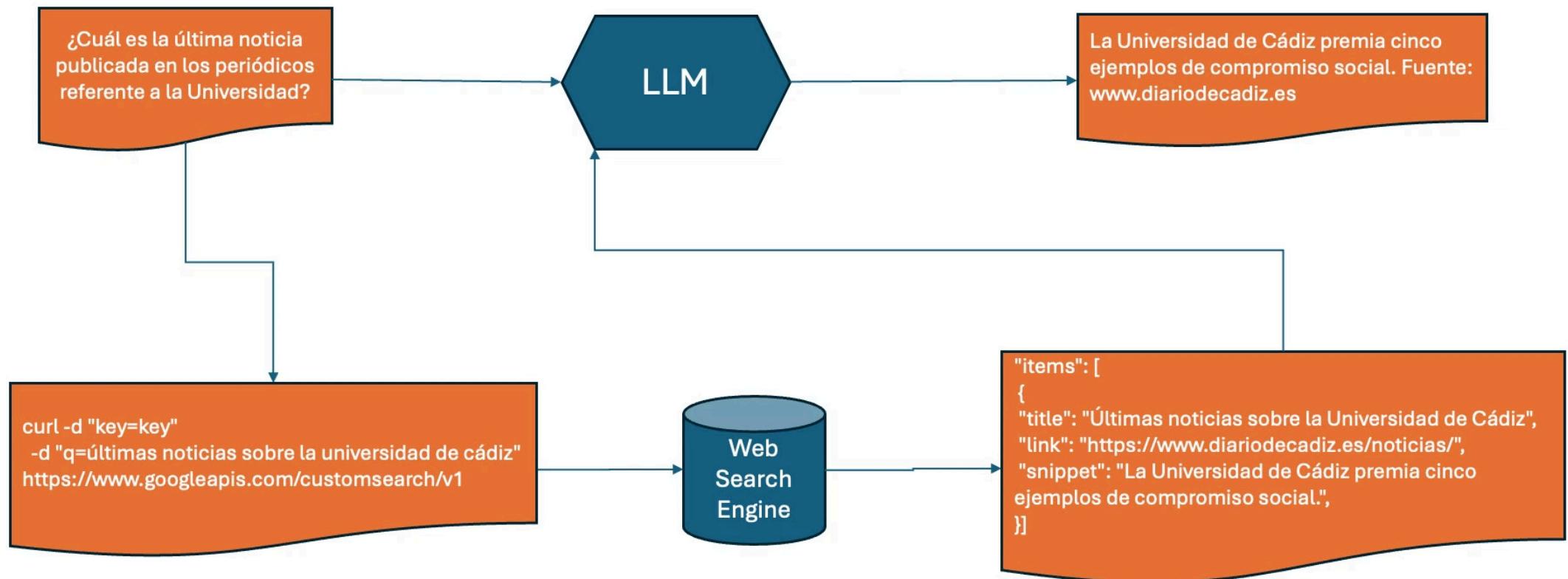
{{databaseStructure}}

If a user asks a question that can be answered by querying this database, generate an SQL SELECT query. Do not output anything else aside from a valid SQL statement!"

Ej 4. Creación de un buscador semántico en una base de datos relacional



Búsqueda semántica en web



Plataformas para búsquedas web



Motores de búsqueda

Google, Bing, y DuckDuckGo ofrecen API para acceder a información en línea.



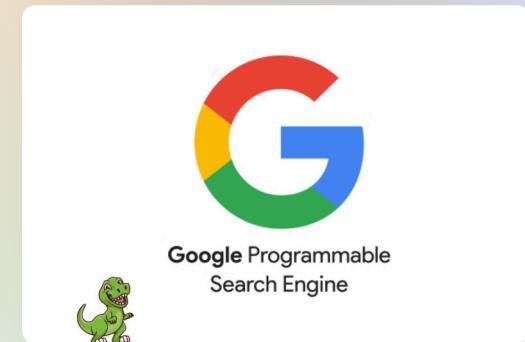
API de búsqueda web

Tavily, SerpApi, Serper y SearchApi son plataformas especializadas en la búsqueda de información web.



API de búsqueda web

Estas API permiten a las aplicaciones integrar funciones de búsqueda sin desarrollar su propio motor.



Generación de información

Generación de información

La generación de información es un proceso complejo que requiere comprender el lenguaje natural, procesar información y generar texto creativo, adaptando el lenguaje y el estilo a las necesidades del usuario.

El LLM combina el prompt de entrada con la información relevante para generar la salida.

La calidad de la salida depende de la calidad de la información recuperada y de la capacidad del propio LLM.

Generación de información

Hay que instruir al LLM con un prompt específico para que genere la respuesta deseada

"

...

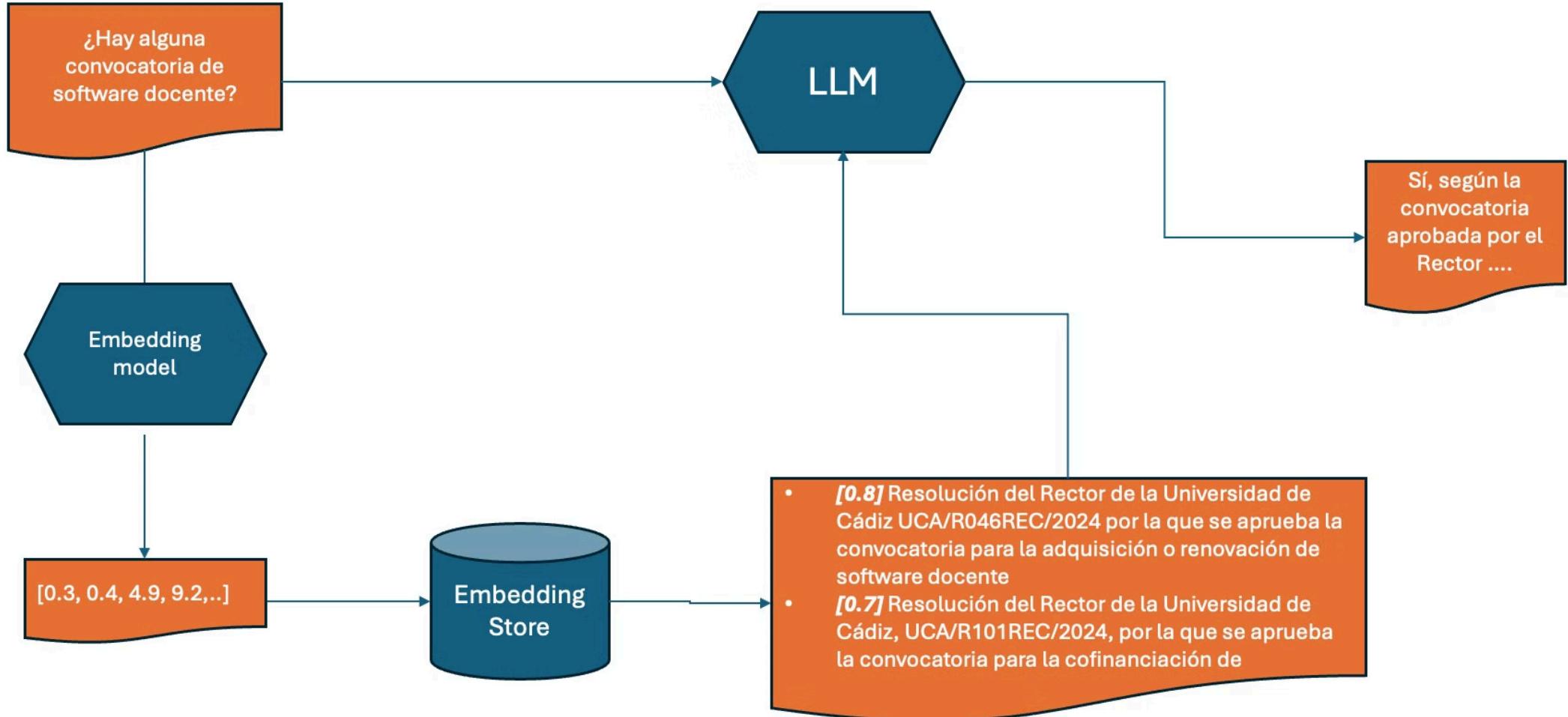
Answer the question based only on the following context:

{context}

Question: {question}

"

Generation (ejemplo)



Ej 6. Creación de un asistente RAG





Limitaciones del RAG básico



1 Irrelevancia

Textos recuperados pueden ser poco pertinentes o carecer de información vital.

3 Inconsistencia Estilística

Variaciones en estilo y tono dificultan coherencia del resultado final.

5 Preguntas inabordables

Las consultas complejas con múltiples partes no pueden ser resueltas

2 Redundancia

Información similar de diferentes fuentes genera duplicidad innecesaria.

4 Sobrecarga Informativa

Exceso de datos puede diluir la importancia de la información clave.

6 Preguntas no explícitas

Las consultas pueden hacer referencia a información implícita y desconocida para el *retriever*

Software can be chaotic, but we make it work



Expert

Trying Stuff Until it Works

O RLY?

The Practical Developer
@ThePracticalDev

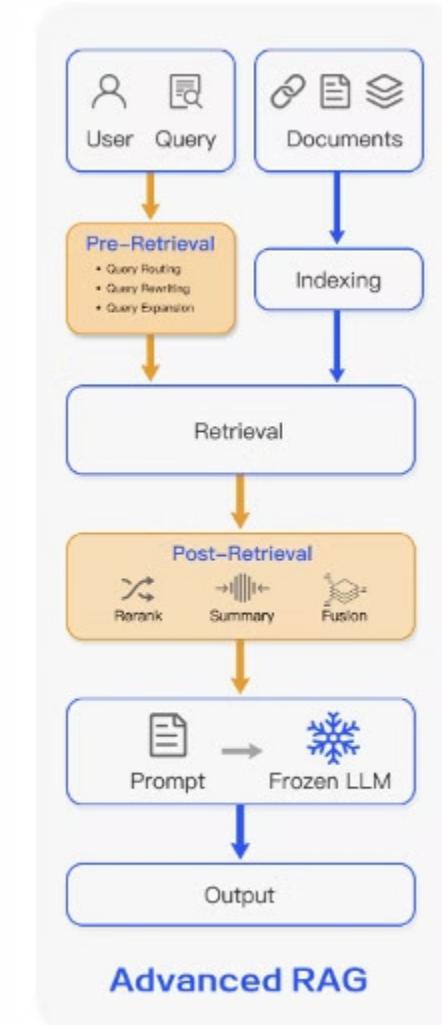
Técnicas del RAG Avanzado

Pre-retrieval

Técnicas aplicadas antes de la recuperación de información.

Post-retrieval

Métodos utilizados después de la recuperación.



Técnicas Pre-Retrieval

Técnicas Pre-Retrieval

Query Rewriting

La reformulación de consultas es una técnica crucial para optimizar la búsqueda y la recuperación de información.

Query Expansion

Ampliar la búsqueda implica agregar términos adicionales a la consulta original para mejorar la precisión y la exhaustividad de los resultados.

Query Compression

La compresión de consultas consiste en contextualizar la consulta original para optimizar la eficiencia y la precisión de la búsqueda.

Query Routing

El enrutamiento de consulta dirige la consulta a las fuentes de datos más relevantes, optimizando la eficiencia y la pertinencia.

Query Rewriting: reformulación de consultas

1

Identificación

Se analiza la consulta original del usuario para detectar áreas de mejora.

2

Procesamiento

Un LLM modifica la consulta incluyendo términos relacionados y entidades reconocidas.

3

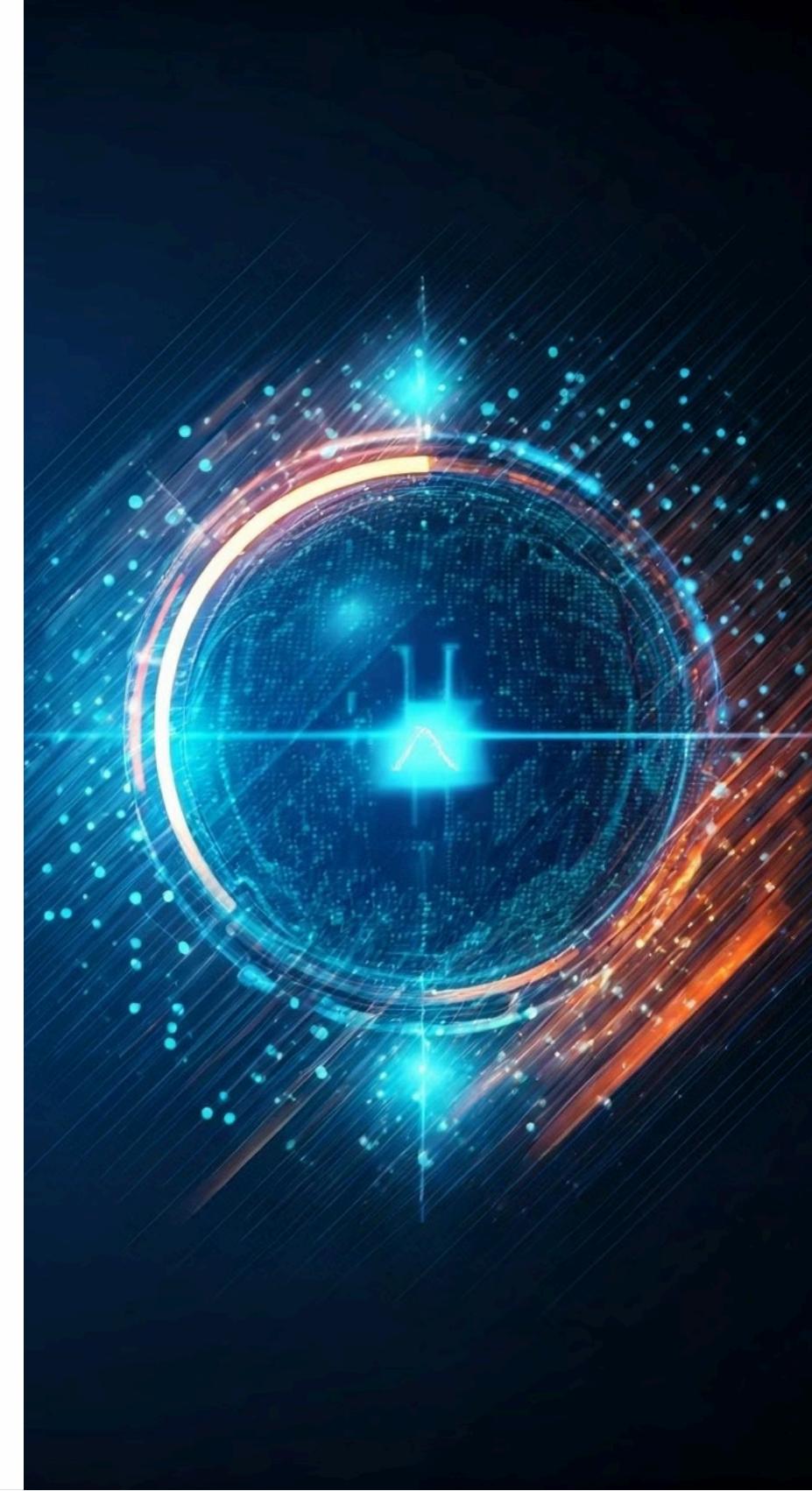
Optimización

Se realiza una paráfrasis para alinear con el vocabulario de la base de conocimientos.

4

Refinamiento

Se aplican correcciones ortográficas y gramaticales para mejorar la precisión.





Query Rewriting: ejemplo

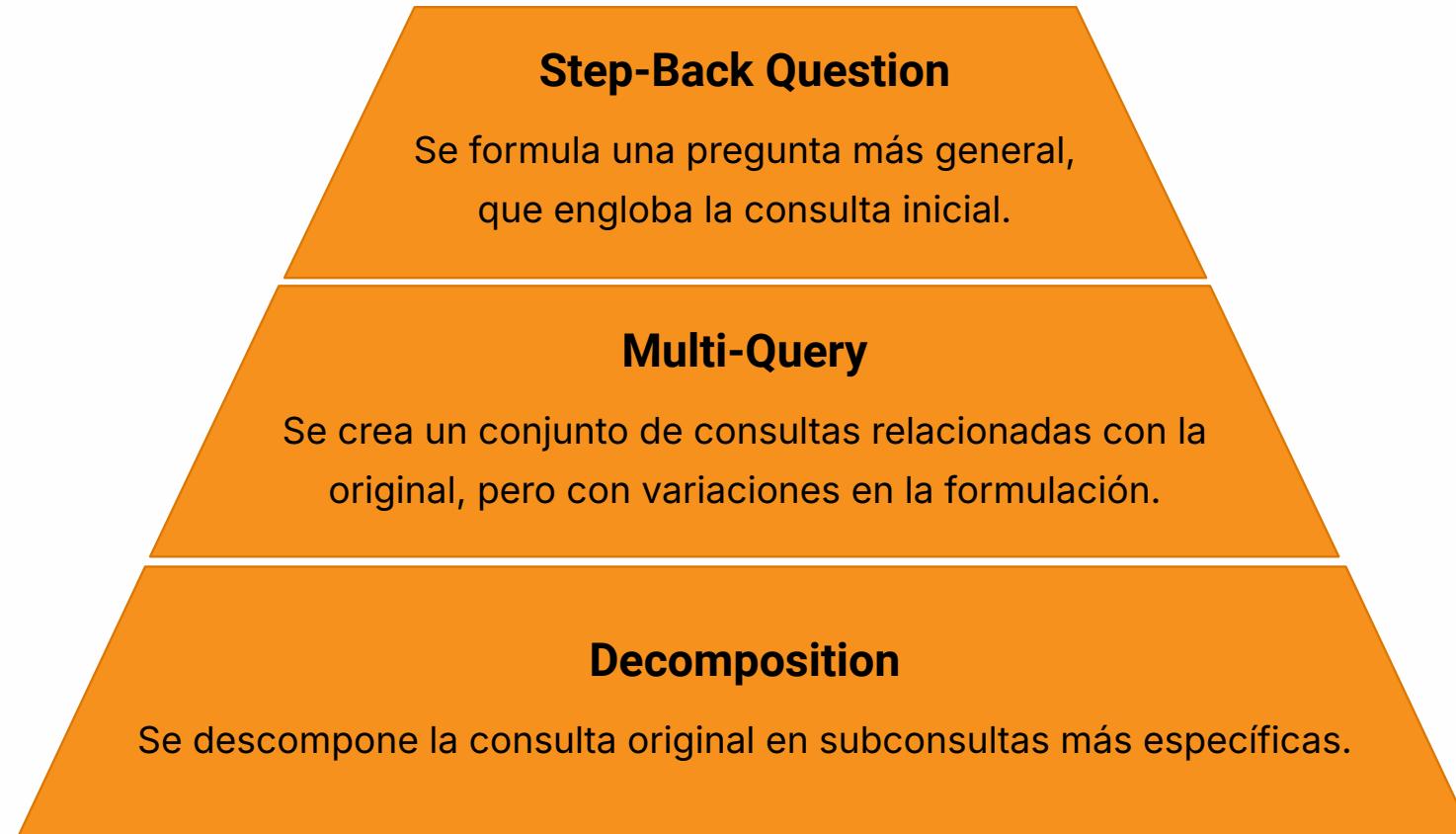
Consulta Original

"Proyectos renovables Cádiz universidad, grupos investigando?"

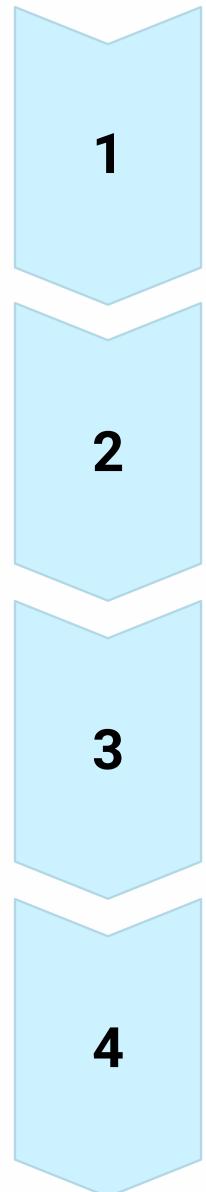
Consulta Modificada

"¿Cuáles son los proyectos de investigación en energías renovables que están desarrollando los grupos de investigación en la Universidad de Cádiz?"

Query Expansion: ampliar la búsqueda



Step-Back Question



Consulta Específica

1

Se parte de la pregunta original del usuario.

Generalización

2

Se formula una pregunta más abstracta y amplia.

Recuperación Dual

3

Se obtienen resultados tanto de la pregunta original como de la generalizada.

Síntesis

4

Se combina la información para proporcionar una respuesta más completa y contextualizada.





Step-Back Question: ejemplo

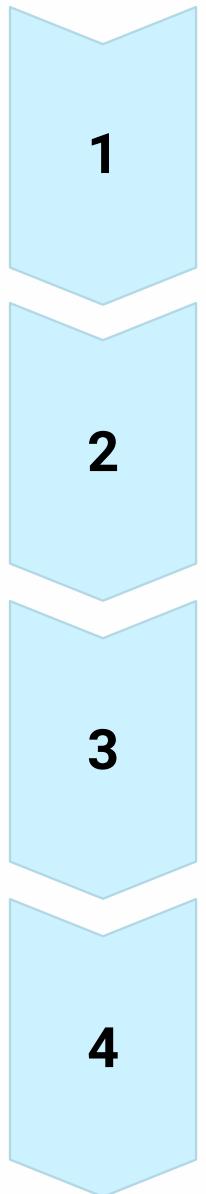
Consulta Original

"¿Qué estrategias específicas puede aplicar un estudiante de primer año de ingeniería para mejorar su rendimiento en los exámenes de matemáticas en la Universidad de Cádiz?"

Consulta Adicional

"¿Cuáles son los factores que influyen en el rendimiento académico de los estudiantes de primer año en la Universidad?"

Multi-Query



1 Consulta Inicial

Se parte de la pregunta original del usuario.

2 Generación

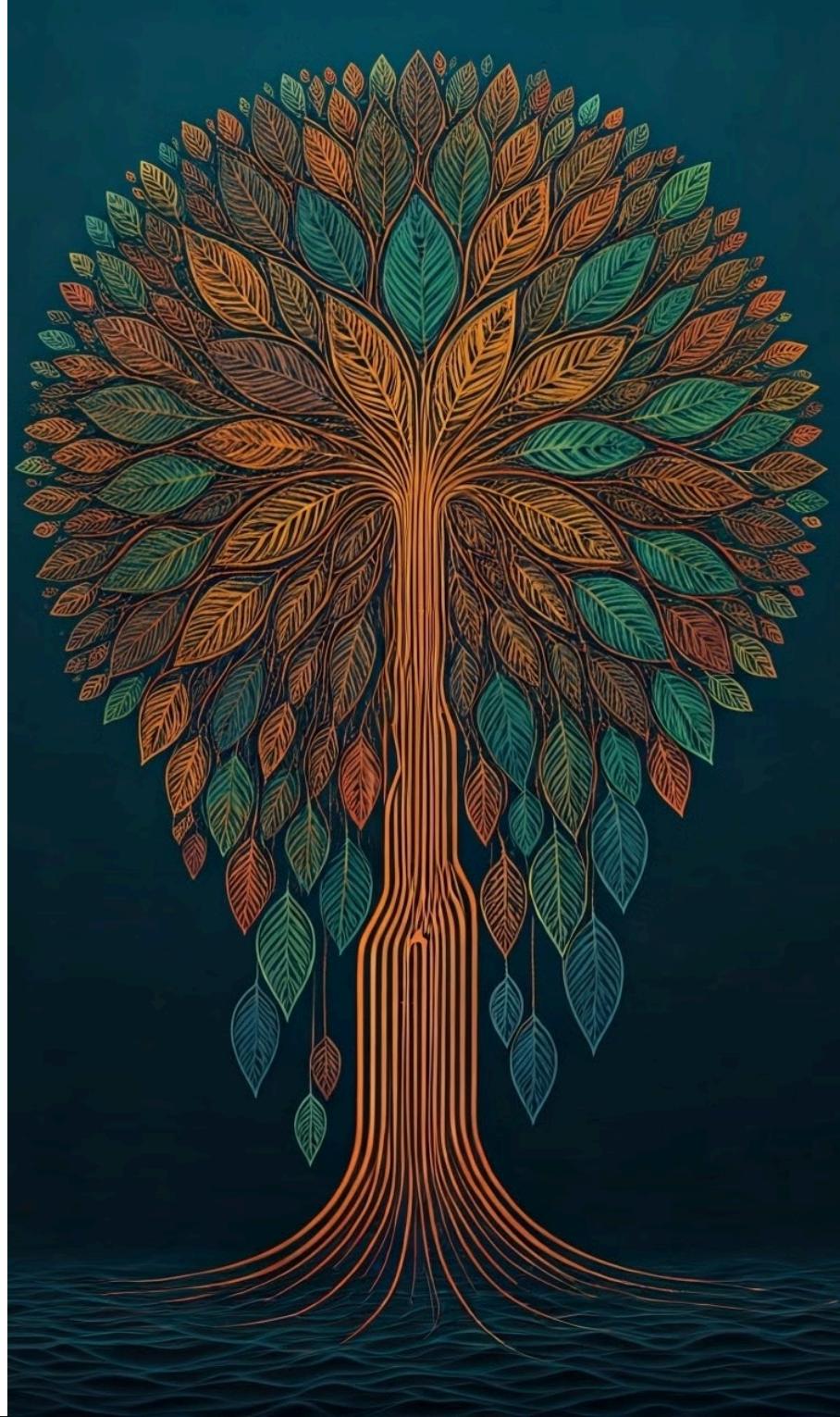
Se crean múltiples variantes manteniendo el nivel de abstracción.

3 Diversificación

Se abordan diferentes perspectivas de la misma consulta.

4 Recuperación

Se utilizan todas las variantes para obtener resultados más completos.





Multi-Query: ejemplo

Consulta Original

"¿Qué programas de intercambio ofrece la Universidad de Cádiz?"

Consultas Generadas

"¿Cuáles son los convenios internacionales que tiene la Universidad de Cádiz con universidades europeas?"

"¿Qué requisitos se necesitan para participar en el programa Erasmus+?"

"¿Existen oportunidades de intercambio fuera de Europa, como en América o Asia?"

"¿Qué apoyo financiero o becas ofrece la Universidad de Cádiz para estudiantes de intercambio?"

Query Decomposition

1

Análisis

Se examina la consulta original para identificar componentes clave.

2

Descomposición

Se divide en subpreguntas más específicas y concretas.

3

Procesamiento

Se realizan consultas paralelas o secuenciales según la naturaleza de las subpreguntas.

4

Recuperación

Se utilizan todas las subpreguntas para obtener resultados más completos.





Query Decomposition: ejemplo

Consulta Original

"¿Qué métodos de evaluación son más comunes en las carreras de ciencias en la Universidad de Cádiz y cómo se comparan con los métodos utilizados en las carreras de humanidades?"

Subconsultas Derivadas

"¿Cuáles son los métodos de evaluación más comunes en las carreras de ciencias en la Universidad de Cádiz?"

"¿Cuáles son los métodos de evaluación utilizados en las carreras de humanidades en la Universidad de Cádiz?"

"¿Cómo se comparan los métodos de evaluación entre las carreras de ciencias y humanidades?"

Query Compression: contextualizar la consulta

1

Análisis del Contexto

Se examina el histórico de la conversación previa.

2

Compresión

Se sintetiza la información relevante del contexto.

3

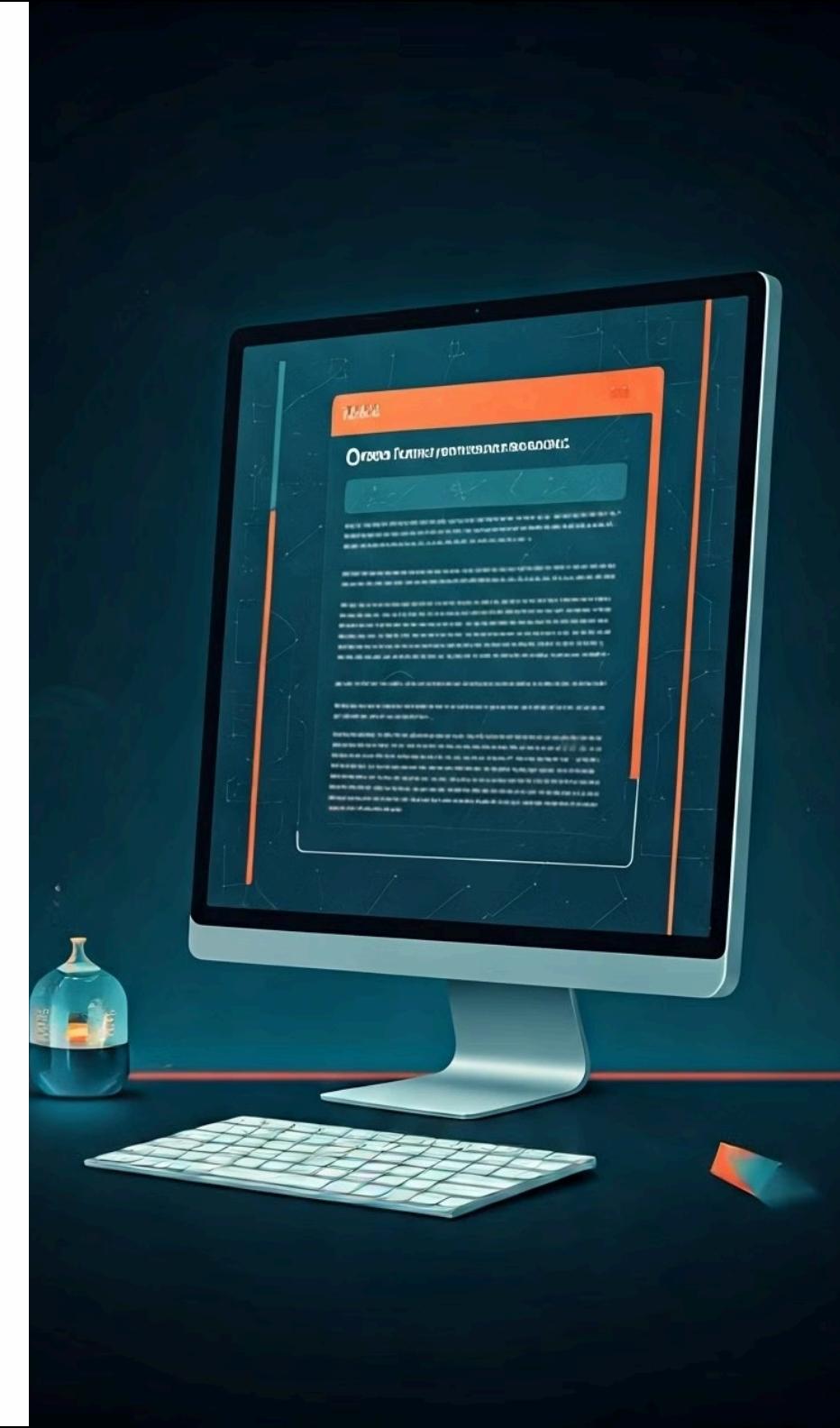
Reformulación

Se crea una nueva consulta que integra el contexto comprimido.

4

Recuperación Optimizada

Se utiliza la consulta comprimida para obtener resultados más precisos.





Query Compression: ejemplo

[... conversación previa sobre formación en idiomas..]

Consulta Original

"¿Cuál es el horario del curso"

Subconsultas Derivadas

"¿Cuál es el horario del curso de Inglés C1 que se imparte en el Campus de Puerto Real durante el primer semestre del curso 2024/2025?"



Query Routing: enrutamiento de consultas

1 Pertinencia

Asegura que la información recuperada sea relevante para la generación de respuestas.

3 Filtrado

Puede identificar consultas que el sistema no debe responder, mejorando la seguridad.

2 Eficiencia

Dirige la consulta a las fuentes de datos más adecuadas, optimizando recursos.

4 Estrategias Híbridas

Combina enrutamiento basado en keywords y semántico (mediante un LLM) para mayor precisión.



Query Routing: ejemplo

Consulta Original

"¿Qué eventos culturales se organizan en la Universidad de Cádiz? "

Enrutador

El enrutador decide dirigir la consulta a un repositorio específico con datos de eventos organizados por la Universidad de Cádiz, en lugar de buscar sobre toda la 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}}`"

Técnicas Post-Retrieval

Técnicas Post-Retrieval



Rerank

Reorganiza los resultados para priorizar los más relevantes.



Fusion

Combina resultados de múltiples consultas, resolviendo contradicciones y creando respuestas coherentes.



Summary

Sintetiza y destaca la información esencial, reduciendo la sobrecarga de tokens.

Rerank: reordenando los resultados

La similitud no garantiza la relevancia: aunque la búsqueda semántica ya ofrezca los resultados ordenados, no significa que los resultados que produzca sean los más relevantes

"Lost in the middle": Los LLM, como las personas, tienden a enfocarse en el principio y final de los textos.

Considerar metadatos específicos: popularidad del documento, autoridad de la fuente, relevancia temporal, preferencias del usuario.

Se busca introducir diversidad en los resultados para evitar repeticiones.

Existen algoritmos (TD-IDF, BM25) y modelos de IA especializados para estas tareas (Cohere, Jina, etc.)





Rerank: ejemplo

Resultados del retriever semántico

Documento recuperado	semántica
Documento 1: "La Universidad de Cádiz ofrece el Grado en Ingeniería de Telecomunicaciones, que se enfoca en redes y sistemas de comunicación."	0.85
Documento 2: "El Grado en Ingeniería Electrónica Industrial en la UCA incluye formación en automatización y control, con énfasis en sistemas."	0.82
Documento 3: "El Grado en Ingeniería Informática en la UCA cubre programación, redes, sistemas operativos y software."	0.80
Documento 4: "La UCA ofrece el Grado en Diseño Gráfico, que combina creatividad y tecnología digital."	0.75
Documento 5: "El Grado en Estudios Ingleses en la UCA es perfecto para los amantes de la literatura y los idiomas."	0.65

Resultados del reranking

Documento recuperado	Puntuación (BM25)
Documento 3: "El Grado en Ingeniería Informática en la UCA cubre programación, redes, sistemas operativos y software."	13.2
Documento 1: "La Universidad de Cádiz ofrece el Grado en Ingeniería de Telecomunicaciones, que se enfoca en redes y sistemas de comunicación."	9.0
Documento 2: "El Grado en Ingeniería Electrónica Industrial en la UCA incluye formación en automatización y control, con énfasis en sistemas."	8.5
Documento 4: "La UCA ofrece el Grado en Diseño Gráfico, que combina creatividad y tecnología digital."	6.1
Documento 5: "El Grado en Estudios Ingleses en la UCA es perfecto para los amantes de la literatura y los idiomas."	2.5

Fusion: unificando respuestas

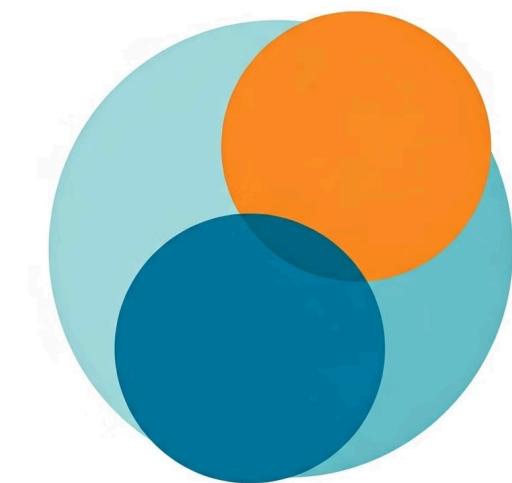
Combinación de Resultados

Cuando usamos expansión de consultas, necesitamos combinar resultados para obtener respuestas completas y coherentes.

Reciprocal Rank Fusion (RRF)

Esta técnica asigna puntuaciones de relevancia a los documentos de las diferentes listas para luego ordenarlos y crear una respuesta unificada.

$$\text{RRF}(d) = \sum_{i=1}^N \frac{1}{k + r_i(d)}$$





Fusion: ejemplo

Resultados del retriever semántico

Posición	Documento	Puntuación BM25
1	Documento F: "Estudia Ingeniería Informática en la Universidad de Cádiz: desarrolla software, aprende inteligencia artificial y redes."	13.2
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 énfasis en la tecnología aplicada al arte digital."	6.1
5	Documento J: "Grado en Estudios Ingleses en la Universidad de Cádiz, orientado a la filología y el análisis literario."	2.5

Posición	Documento	Similitud semántica
1	Documento A: "El Grado en Ingeniería de Telecomunicaciones en la UCA se enfoca en redes y comunicaciones."	0.85
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 tecnología digital."	0.75
5	Documento E: "El Grado en Estudios Ingleses en la UCA se orienta a los amantes de la literatura y los idiomas."	0.65

Resultados de la fusión

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: condensando la información de contexto

Destila, abstrae y compacta los textos recuperados, resaltando los conceptos y hechos clave.

El objetivo es evitar proporcionar al LLM información superflua o confusa y reducir el número de tokens.

Esto ayuda al modelo a comprender mejor la información y generar respuestas más precisas.



Summary: ejemplo



Resultados obtenidos

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

Entrada al LLM

Resumen de titulaciones en la Universidad de Cádiz relacionadas con tecnología y ordenadores:

1. **Grado en Ingeniería Informática** (1er lugar): Este grado es ideal si te gustan los ordenadores. Incluye formación en programación, desarrollo de software, redes y sistemas operativos. Es el programa más alineado con el trabajo directo con ordenadores y tecnología informática.
2. **Grado en Ingeniería de Telecomunicaciones** (2do lugar): Si te interesan las redes y los sistemas de comunicación, este grado es una excelente opción. Aquí aprenderás sobre transmisión de datos y tecnologías de comunicaciones.
3. **Grado en Ingeniería Electrónica Industrial** (3er lugar): Aunque más orientado a la automatización y el control de procesos industriales, este programa incluye formación en electrónica aplicada y sistemas tecnológicos, lo cual puede ser interesante si te gustan las máquinas y los sistemas controlados por ordenadores.

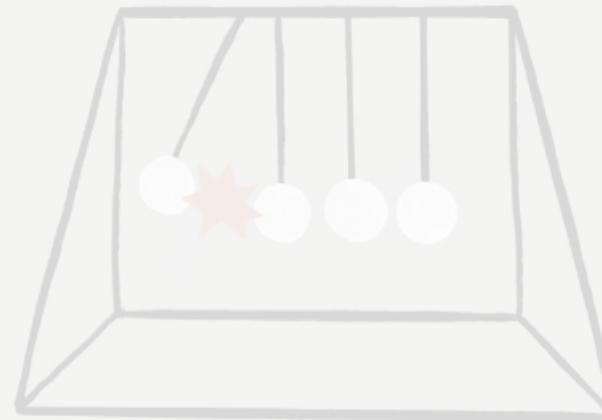
Otras técnicas avanzadas

Introducing Contextual Retrieval

Otras técnicas avanzadas

19 Sept 2024 • 10 min read

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





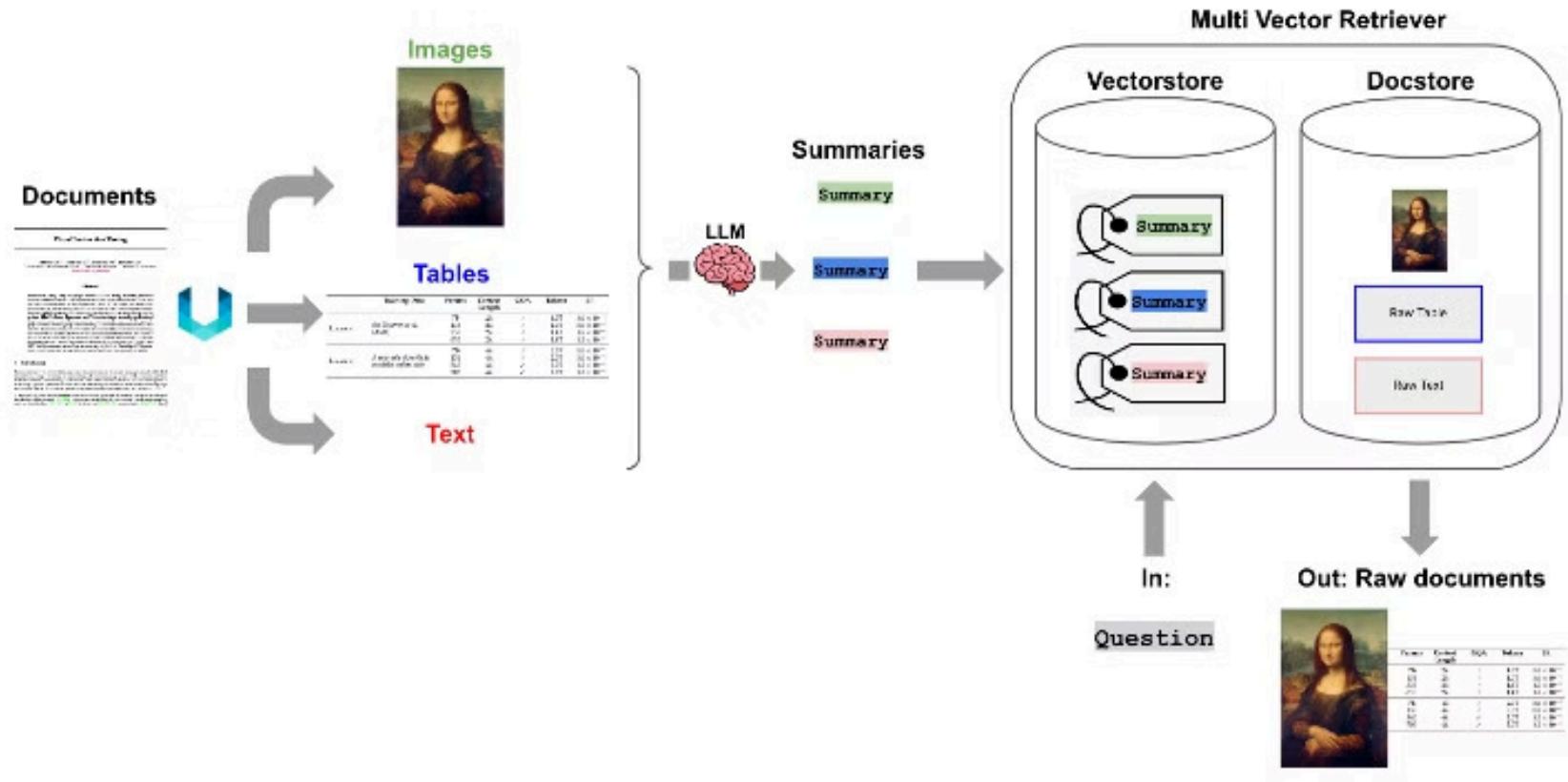
Summary Embedding + Parent

- Al dividir documentos para la recuperación, a menudo hay deseos contradictorios:
- La calidad de la **recuperación** mejora cuando los embeddings son generados desde textos cortos. Si son demasiado largos, los embeddings pueden perder la semántica completa del texto.
- La calidad de la **generación** mejora cuando se dispone de un contexto largo y detallado, ya que éste conlleva más relaciones y conceptos semánticos. Si usamos textos cortos las respuestas del LLM pueden ser menos precisas al carecer de suficiente información.
- Los documentos, además de **texto**, pueden incluir otros tipos de el

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

Multi modal RAG

Los documentos, además de texto, pueden incluir imágenes y tablas, lo que puede enriquecer el conocimiento del LLM.



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

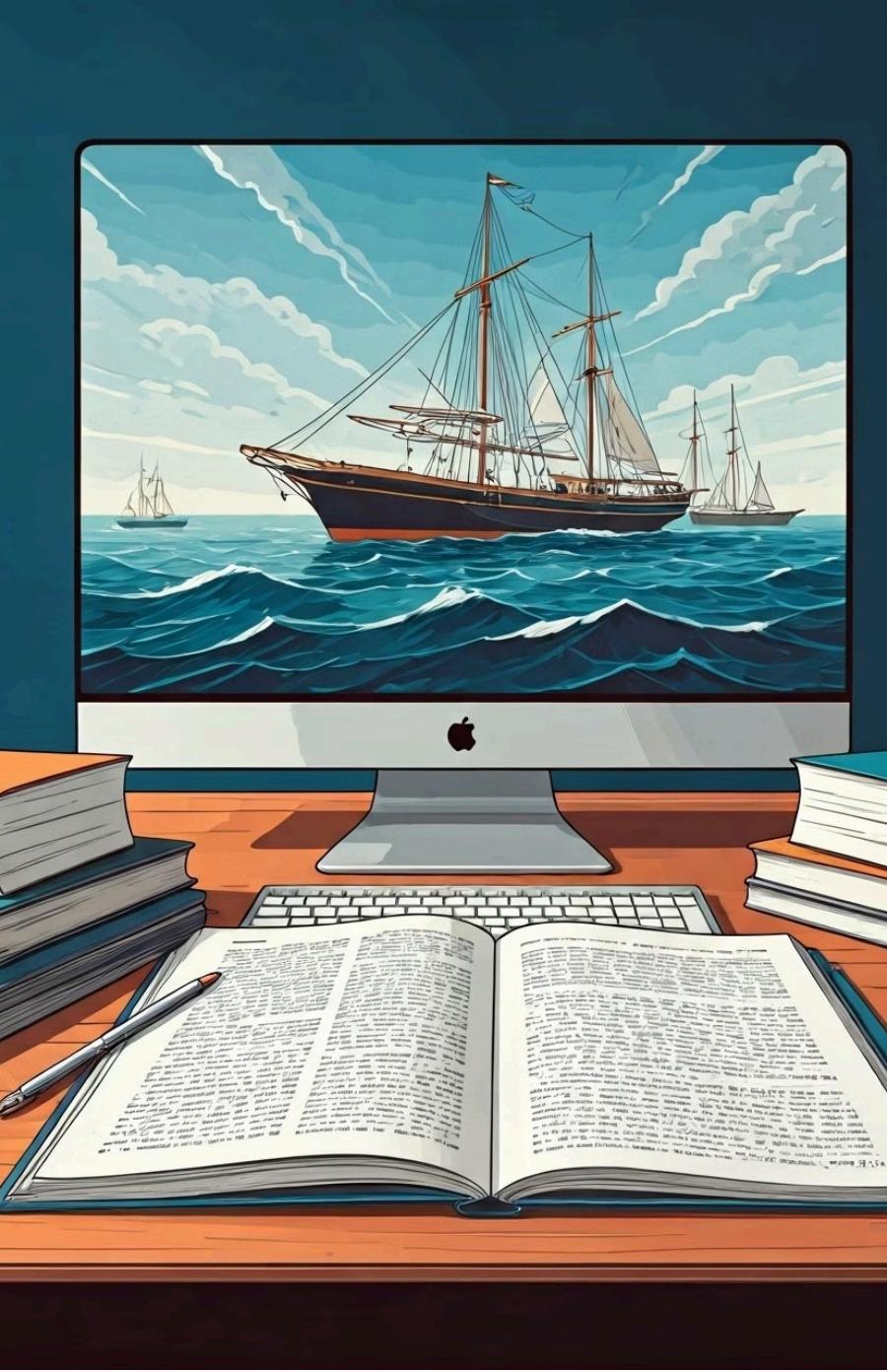


Hypothetical Document Embedding (HyDE)

La búsqueda por similaridad de documentos puede fallar si la representación vectorial (embedding) de la pregunta no es lo suficientemente similar a los documentos relevantes.

Con esta técnica se genera un documento hipotético, que luego es utilizado para realizar una búsqueda por similitud en la base de conocimiento. Esto ayuda a encontrar información más relevante y a generar respuestas más precisas.

- ❑ <https://js.langchain.com/docs/integrations/retrievers/hyde/>



Ejemplo de HyDE

Consulta Original

"¿Cómo afecta el aumento de la temperatura del océano a la biodiversidad de los arrecifes de coral?"

Documento generado

"El calentamiento global ha provocado un aumento sostenido de las temperaturas en los océanos, lo que afecta negativamente a los ecosistemas de los arrecifes de coral. La elevación de la temperatura del agua marina acelera el fenómeno de blanqueamiento del coral, en el cual los corales expulsan a las algas simbióticas que los nutren, reduciendo la biodiversidad en estos ecosistemas. La pérdida de especies claves en los arrecifes tiene efectos en cadena que afectan a toda la cadena alimentaria marina."

Resumen

Esta presentación trató sobre la evolución de la información en el ámbito de la inteligencia artificial (IA), específicamente en el contexto de la búsqueda y la generación de información.

Exploramos cómo RAG, una tecnología que combina la recuperación de información con la generación de texto, revolucionó la forma en que las máquinas comprenden e interactúan con el conocimiento.