



Introducción a RAG (Retrieval Augmented Generation)

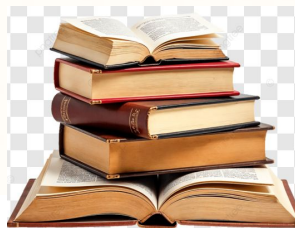
Autor: Ing. Ángel Nicolás Heredia. (LIDeSIA - FCEFYN)
Directora: Dra. Laura Cecilia Díaz Davila (LIDeSIA - FCEFYN)

¿Qué es RAG?

La generación aumentada por recuperación (RAG) es una técnica que permite a los grandes modelos de lenguaje (LLM) recuperar e incorporar nueva información.



Usuario con
pregunta de
un tema muy
específico o
actual



Mejor respuesta




¿Qué ventajas ofrece?

- Respuestas actualizadas
- Reducción de "alucinaciones"
- Conocimiento especializado o privado
- Verificabilidad
- Eficiencia y ahorro de costos




¿Cuáles son las partes de RAG?

Un RAG típico funciona en dos etapas principales:



Fase 1: Indexación de Datos (El trabajo previo)

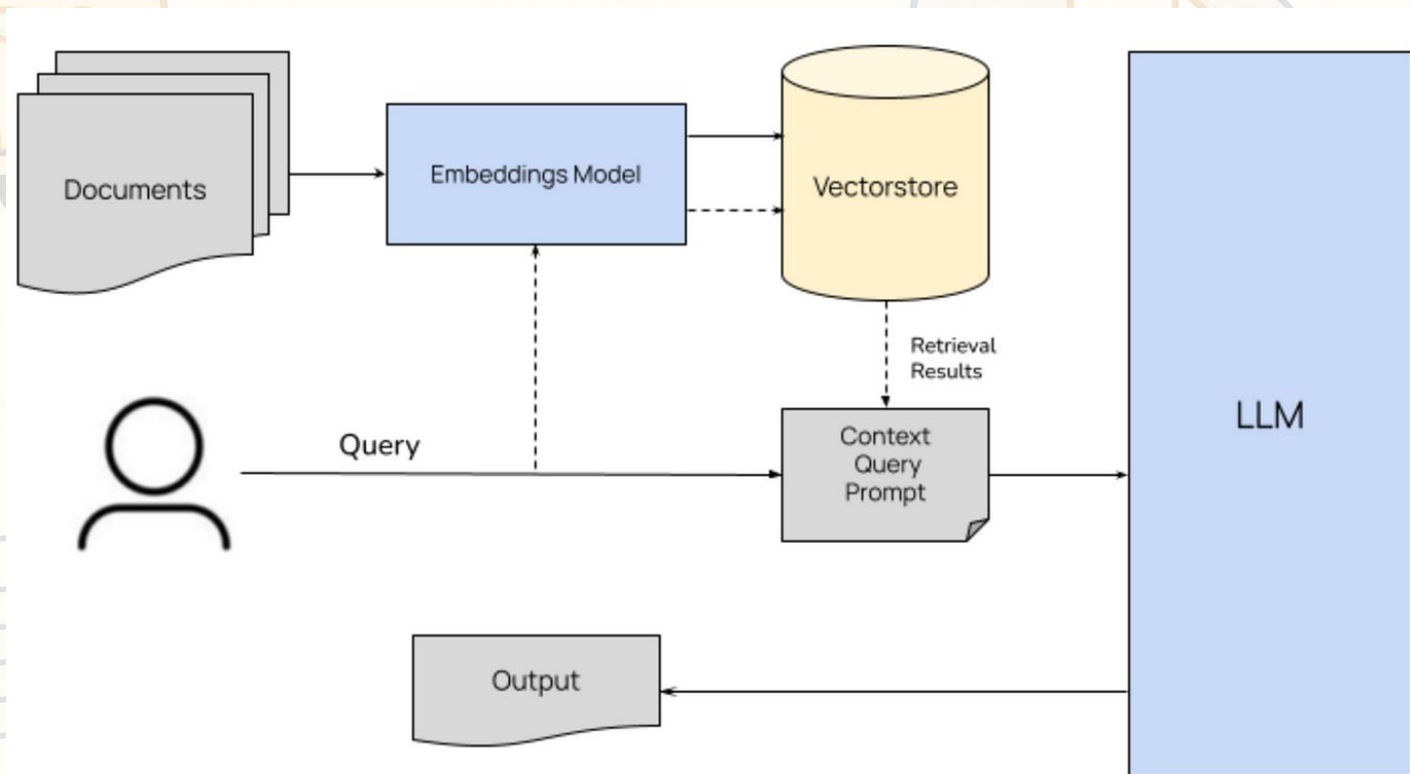
- 1. Cargador de Documentos**
- 2. Fragmentador (Chunker/Splitter)**
- 3. Vectorización (Embedding)**
- 4. Indexación y Almacenamiento**



Fase 2: Recuperación y Generación (En tiempo real, al recibir una consulta)

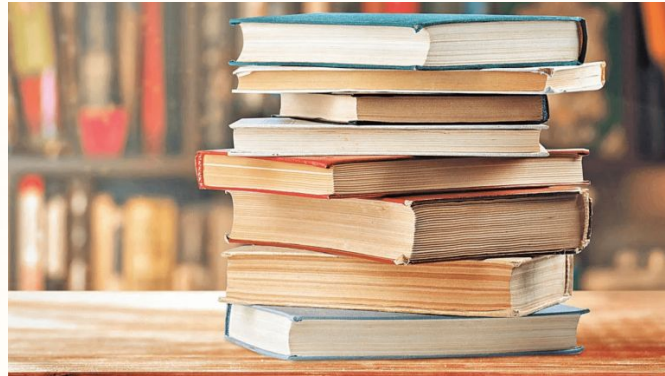
- 5. Recuperador (Retriever)**
- 6. Incorporación al Prompt (Prompt Augmentation)**
- 7. Generación de la Respuesta (LLM Generation)**

Diagrama



Cargador de Documentos (Document Loader):

Es el primer paso. Se encarga de leer y cargar los documentos desde múltiples fuentes: pueden ser archivos PDF, páginas de un sitio web, documentos de Word, registros de una base de datos, etc.



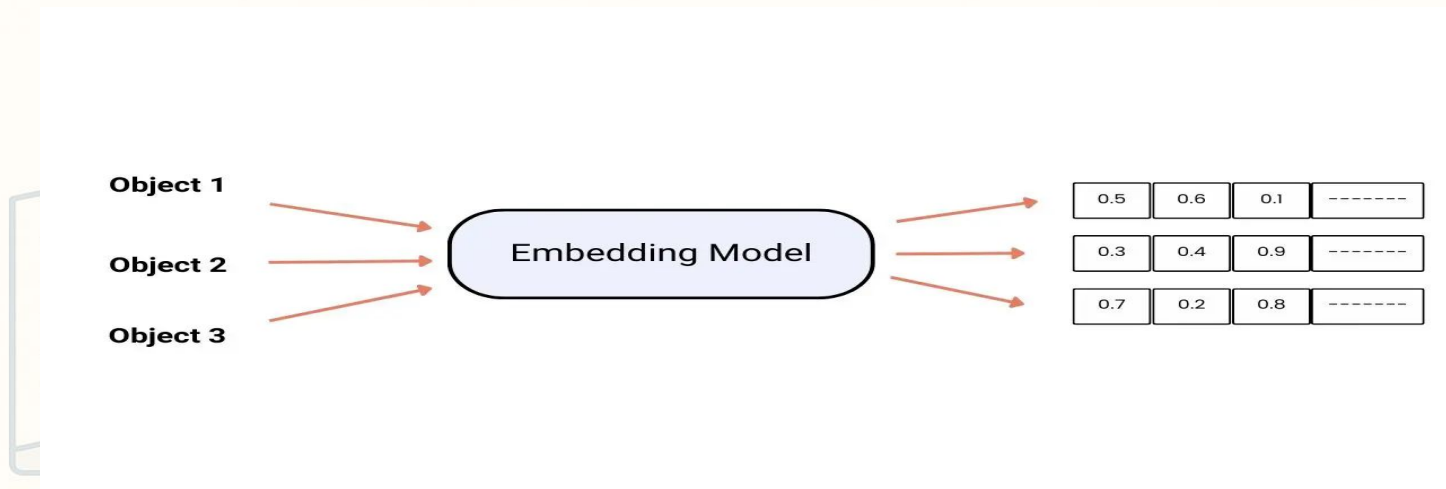
Fragmentador (Chunker/Splitter):

Los documentos cargados suelen ser muy largos. Este componente los divide en fragmentos de texto más pequeños y manejables (los "chunks").



Vectorización (Embedding)

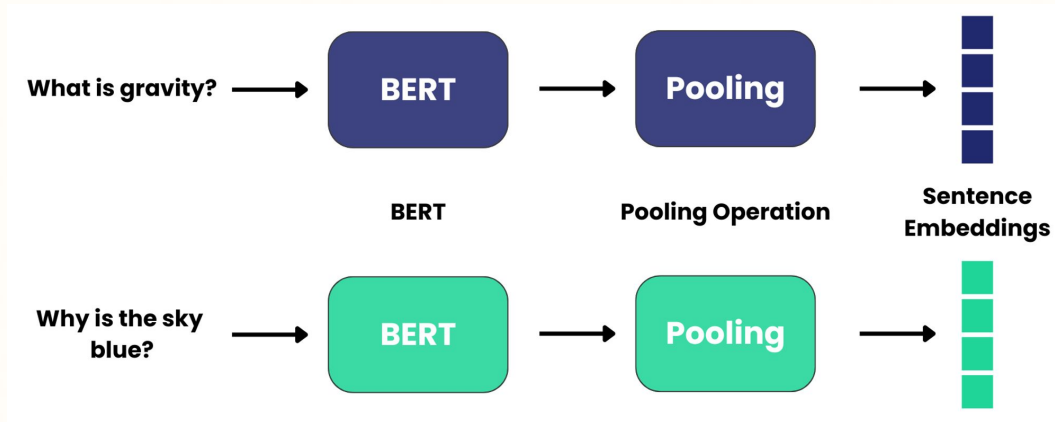
Cada fragmento de texto se convierte en un vector numérico usando un modelo de lenguaje especializado (un embedding model). Este vector es como una "huella digital" que captura el significado semántico del texto.



Vectorización (Embedding)

Lo más utilizado para generar un vector de todo un pedazo de texto son los “Sentence Transformer”

1. La Base: Un Modelo de Lenguaje Pre-entrenado (como BERT)
2. BERT no produce un único vector de "significado"
3. Pooling
4. Usar Red Siamesa (Siamese Network)



Indexación y Almacenamiento (Vector Store)

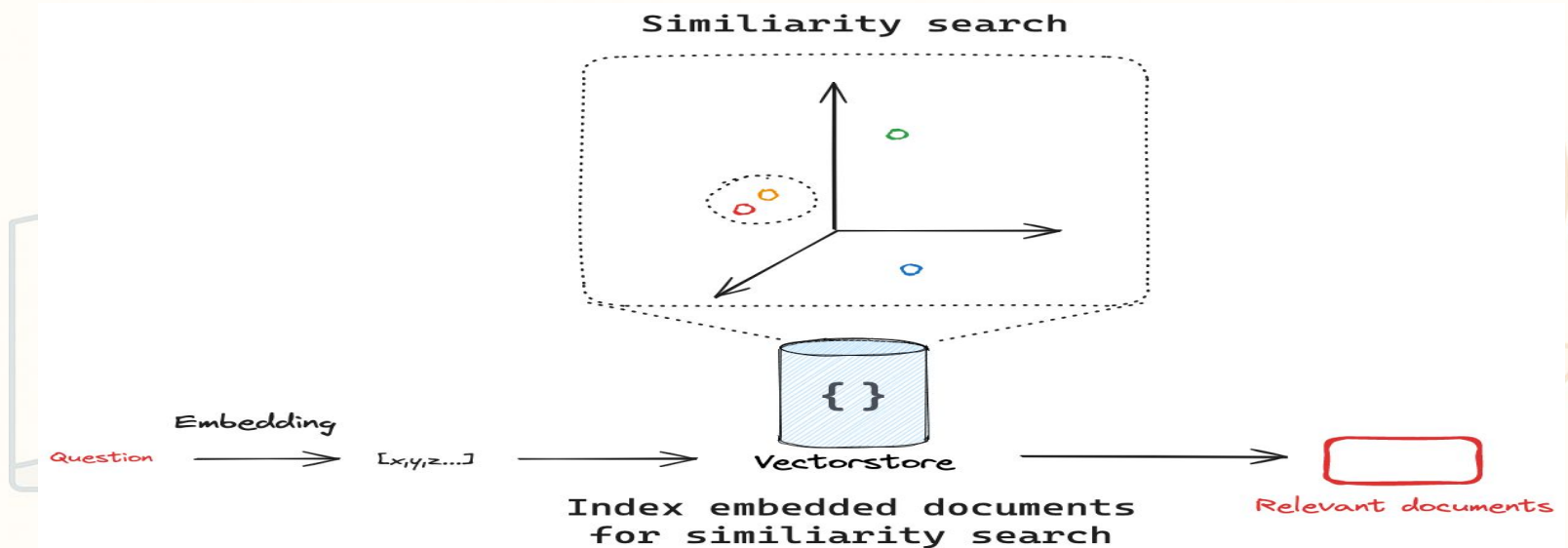
Estos vectores, junto con una referencia al fragmento de texto original, se cargan y se guardan en una base de datos vectorial. Esta base de datos está optimizada para buscar vectores semánticamente similares de forma muy eficiente.



Indexación y Almacenamiento (Vector Store)

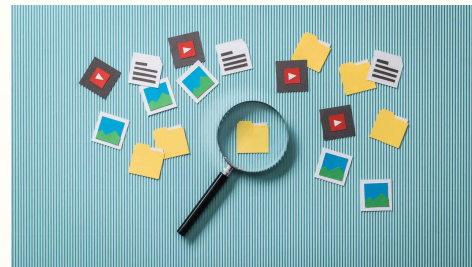
Los algoritmos más comunes y famosos son:

- HNSW (Hierarchical Navigable Small World)
- **IVF (Inverted File System)** — Faiss (Facebook AI Similarity Search)
- Annoy (Approximate Nearest Neighbors Oh Yeah)



Recuperador (Retriever)

La pregunta del usuario (el query) también se convierte en un vector usando el mismo modelo de embedding. El recuperador toma este vector y busca en la base de datos vectorial los fragmentos de texto cuyos vectores son más "cercaños" o similares semánticamente a la pregunta.



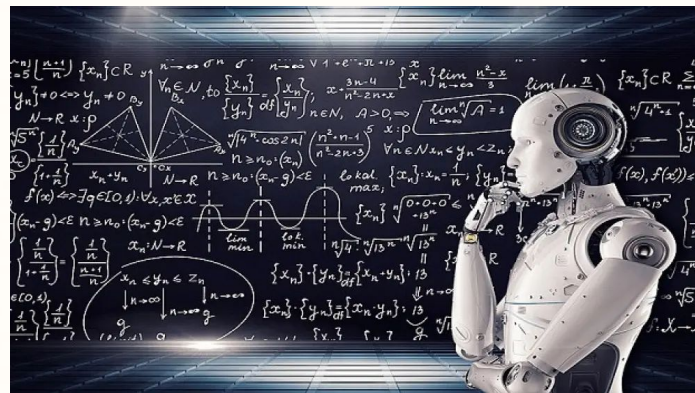
Incorporación al Prompt (Prompt Augmentation)

Los fragmentos de texto recuperados, que son el contexto más relevante para la pregunta, se insertan en una plantilla de prompt junto con la pregunta original del usuario.



Generación de la Respuesta (LLM Generation)

El prompt aumentado (pregunta + contexto recuperado) se envía finalmente al LLM. El modelo de lenguaje utiliza la información proporcionada para formular una respuesta completa, coherente y basada en los hechos recuperados.



Ir al colab

Bibliografía

https://en.wikipedia.org/wiki/Retrieval-augmented_generation

<https://www.marqo.ai/course/introduction-to-sentence-transformers>

<https://medium.com/@j13mehul/rag-part-4-indexing-1985f4000f72>

<https://www.oracle.com/ar/artificial-intelligence/generative-ai/retrieval-augmented-generation-rag/>

