DIGISOFT

RAG (Retrieval-Augmented Generation): Enriqueciendo a los Modelos LLM

Miguel Cotrina

Miguel Cotrina

- **Perfil Académico**
  - Ingeniero de software por la Universidad Tecnológica del Perú.
  - Maestría en ciencia de datos por la Universidad Ricardo Palma.
- **Perfil Profesional**
  - Arquitecto de datos región latam en Indra
  - Consultor e Instructor de Big Data, cloud, IA e IA Gen en empresas privadas y públicas

Agenda

- Que es RAG y por qué es importante
- Arquitectura general de un sistema RAG
- Casos de uso
- Buenas practicas
- Laboratorio practico

Que es RAG y por qué es importante

Miguel Cotrina

Que es RaG

Retrieval-Augmented Generation (RAG) es una técnica que combina modelos generativos de lenguaje (LLMs) con sistemas de recuperación de información. Esto permite que los modelos generativos accedan a información externa relevante en tiempo real, mejorando la precisión y actualidad de sus respuestas.

RAG integra dos componentes principales:

1. Recuperación (Retrieval): Busca documentos relevantes en una base de datos externa utilizando técnicas de búsqueda semántica.
2. Generación (Generation): El modelo generativo utiliza los documentos recuperados para generar respuestas más informadas y precisas.

Esta arquitectura fue introducida por investigadores de Meta AI en:

"Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks"

— Patrick Lewis et al., Meta AI

Por qué es importante

Los modelos de lenguaje tradicionales tienen limitaciones en cuanto a la actualización de conocimientos y pueden generar respuestas inexactas si la información no está presente en sus datos de entrenamiento. RAG aborda estas limitaciones al permitir que los modelos:

- Accedan a información actualizada sin necesidad de reentrenamiento.
- Reduzcan las "alucinaciones" al basar las respuestas en datos recuperados.
- Proporcionen respuestas más precisas y relevantes en contextos específicos.

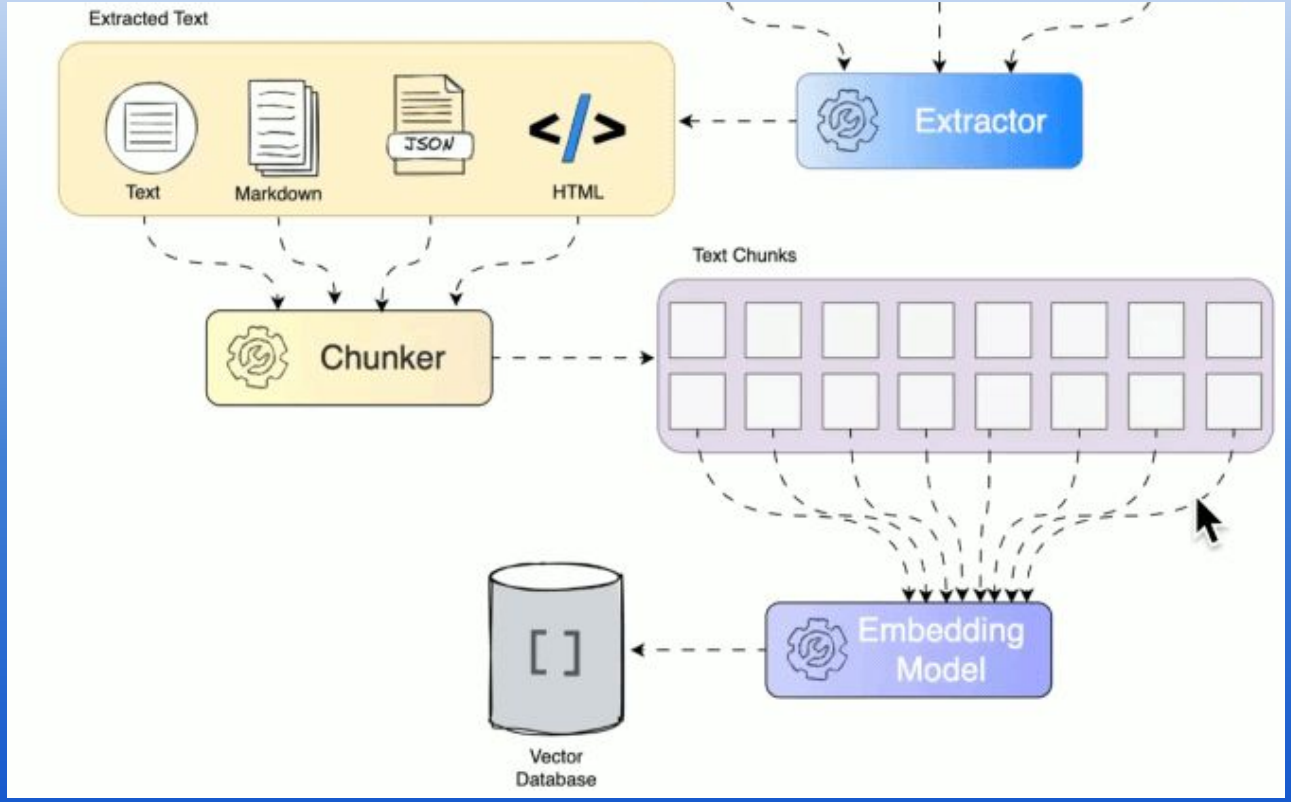
Según la documentación de LangChain:

"RAG es una técnica para aumentar el conocimiento de los LLMs con datos adicionales. Si deseas construir aplicaciones de IA que puedan razonar sobre datos privados o introducidos después de la fecha de corte del modelo, necesitas aumentar el conocimiento del modelo con la información específica que necesita."

Arquitectura general de un sistema RAG

Miguel Cotrina

Arquitectura de carga



Ingesta y preprocesamiento de datos

Los datos, tanto estructurados como no estructurados (por ejemplo, documentos, bases de datos, páginas web), se recopilan y preparan para su procesamiento.

Procesos clave:

- Ingesta: Proceso de captura de datos desde el origen
- Chunking: Segmentación de documentos en fragmentos manejables para facilitar la recuperación.

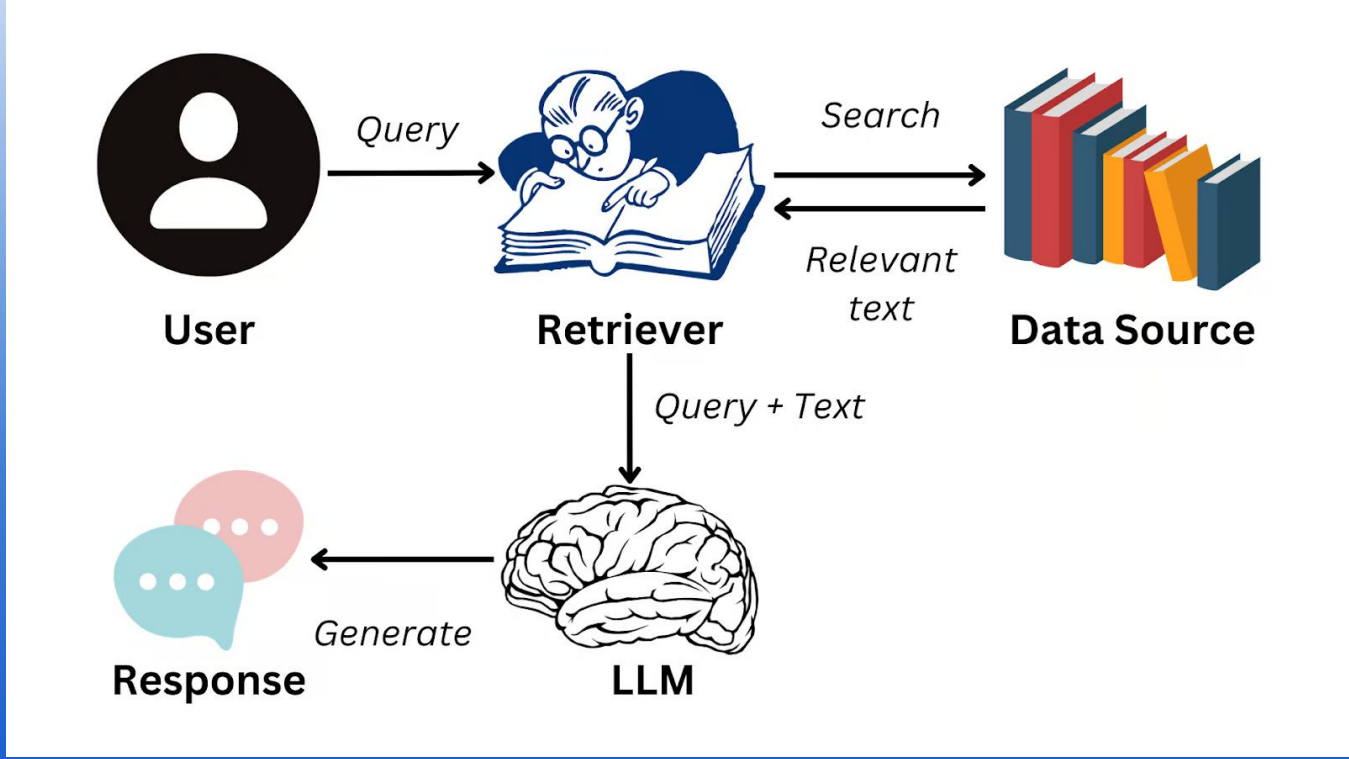
Embeddings y Almacenamiento Vectorial

Los fragmentos de texto se transforman en vectores numéricos (embeddings) que capturan su significado semántico.

Herramientas comunes:

- Modelos de Embeddings: OpenAI, Cohere, Hugging Face Transformers.
- Bases de Datos Vectoriales: FAISS, Chroma, Pinecone, Weaviate.

Arquitectura de recuperación



Recuperación de Información (Retriever)

Ante una consulta del usuario, el sistema busca en la base de datos vectorial los fragmentos más relevantes.

Técnicas utilizadas:

- Búsqueda Semántica: Comparación de la consulta con los embeddings almacenados.
- Re-ranking: Reordenamiento de los resultados para mejorar la relevancia.

Generación de respuesta

El modelo de lenguaje de gran tamaño (LLM) genera una respuesta basada en la consulta original y los fragmentos recuperados.

Modelos populares:

- GPT-4 (OpenAI), Claude (Anthropic), PaLM 2 (Google), LLaMA (Meta).

Casos de uso

Miguel Cotrina

Soporte Técnico y Atención al Cliente

Automatización de respuestas a consultas técnicas complejas, mejorando la eficiencia del soporte al cliente.

Ejemplo: Chatbots empresariales para responder preguntas específicas sobre productos, como especificaciones técnicas

Búsqueda Semántica en Documentación

Facilita la búsqueda de información relevante en grandes volúmenes de documentos internos, como políticas de la empresa o manuales técnicos.

Ejemplo: Sistema RAG avanzado para responder preguntas sobre documentación específica, utilizando LangChain.

Educacion y formacion

Desarrollo de herramientas educativas que proporcionan información actualizada y precisa, mejorando la experiencia de aprendizaje.

Ejemplo: RAG fortalece las herramientas de aprendizaje al proporcionar información actualizada sobre diversos temas.

Laboratorio practico

Miguel Cotrina

Laboratorio aplicado

Codigo compartido

- Flujo RAG
  - Carga
  - Fragmentacion
  - Almacenamiento
- Creación de un agente RAG
  - Recuperación con Tool

https://github.com/macespinoza/programa7genai/tree/main/Clase%2006

Tarea actividad 05

Actividad 01: Desplegar un agente RAG con otro origen de datos

Miguel Cotrina

Agradecimiento y preguntas

Muchas gracias a todas las personas que están interesadas en aprender sobre estas nuevas tecnologías, el camino comienza pero el destino aún es desconocido.

Todas sus preguntas consultas o feedback son bienvenidos y lo pueden dejar en los comentarios del video de cada clase

Redes sociales:

- https://www.linkedin.com/in/mcotrina/
- https://www.youtube.com/@macespinozaonline
- https://github.com/macespinoza/

Programa de Introducción a la IA Generativa con Modelos de Gran Tamaño de 7 clases

Miguel Cotrina