ChatGPT sin alucinaciones con la arquitectura RAG

Contenidos

1. Qué es una alucinación en IA?
2. Por qué ocurren alucinaciones?
3. Qué es una RAG?
4. Cómo construir una aplicación RAG?
5. Conclusiones.
6. Ejemplo práctico.

Qué es una alucinación en IA?

📌 Definición:
"Las alucinaciones en modelos de lenguaje son respuestas incorrectas, inventadas o fuera de contexto generadas con confianza, pero sin base en datos reales."

Qué es una alucinación en IA?

📌 Definición:
"Las alucinaciones en modelos de lenguaje son respuestas incorrectas, inventadas o fuera de contexto generadas con confianza, pero sin base en datos reales."

⚠️ **Ejemplo real**:
👉 Un chatbot respondiendo con datos falsos en medicina o leyes.
👉 Modelos generando referencias a artículos científicos inexistentes

Por qué ocurren alucinaciones?

1️⃣ *Predicción basada en patrones:* ChatGPT no accede a una base de datos en tiempo real, sino que genera respuestas basadas en patrones aprendidos durante su entrenamiento.

Por qué ocurren alucinaciones?

1. *Predicción basada en patrones:* ChatGPT no accede a una base de datos en tiempo real, sino que genera respuestas basadas en patrones aprendidos durante su entrenamiento.

2. *Falta de datos actualizados*: Si se le pregunta algo fuera de su conocimiento o sobre eventos recientes, ChatGPT puede "inventar" respuestas que parecen creíbles.

Por qué ocurren alucinaciones?

1. *Predicción basada en patrones:* ChatGPT no accede a una base de datos en tiempo real, sino que genera respuestas basadas en patrones aprendidos durante su entrenamiento.

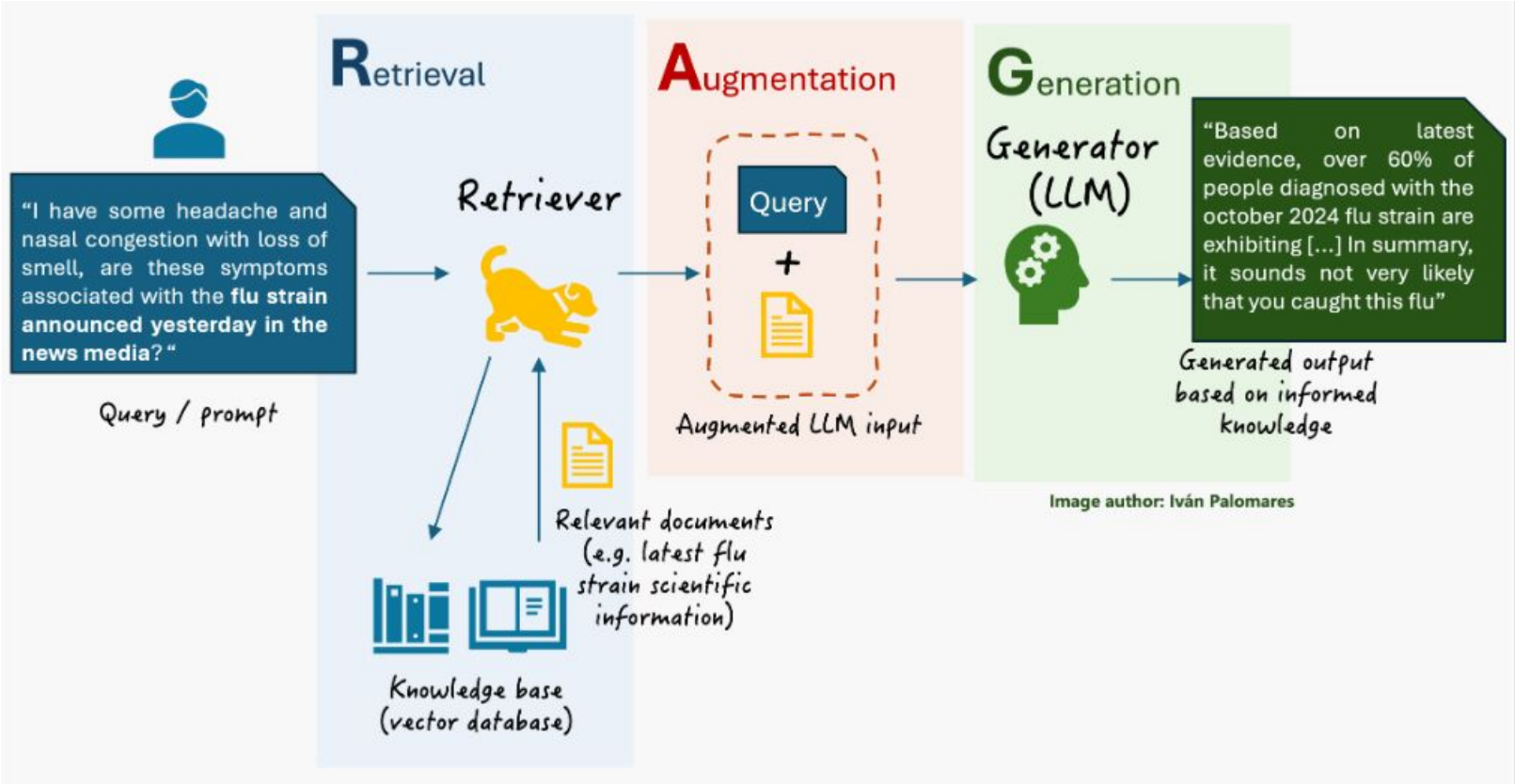
2. *Falta de datos actualizados*: Si se le pregunta algo fuera de su conocimiento o sobre eventos recientes, ChatGPT puede "inventar" respuestas que parecen creíbles.

3. *Adivinación probabilística:* GPT no "razona" como un humano ni verifica hechos en tiempo real. Selecciona las palabras más probables para formar una respuesta.

Estrategias para reducir alucinaciones en LLMs

Estrategias para reducir alucinaciones en LLMs

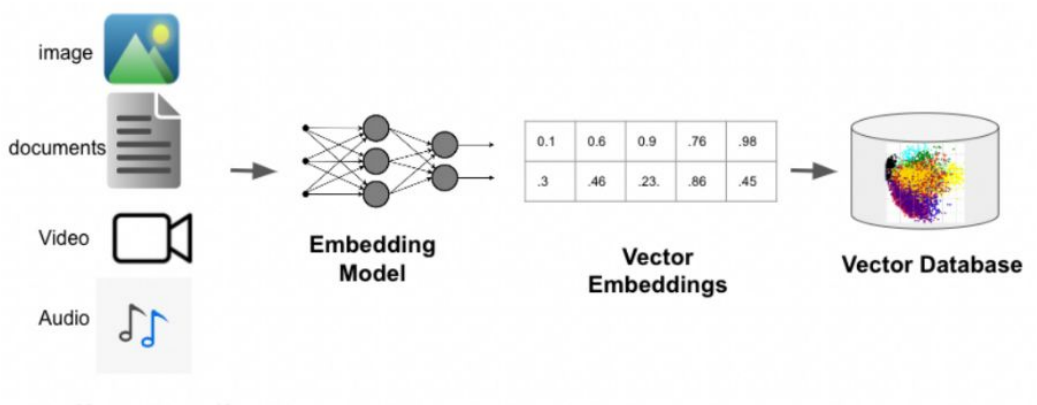
Que es una RAG?



Cómo construir una aplicación RAG?

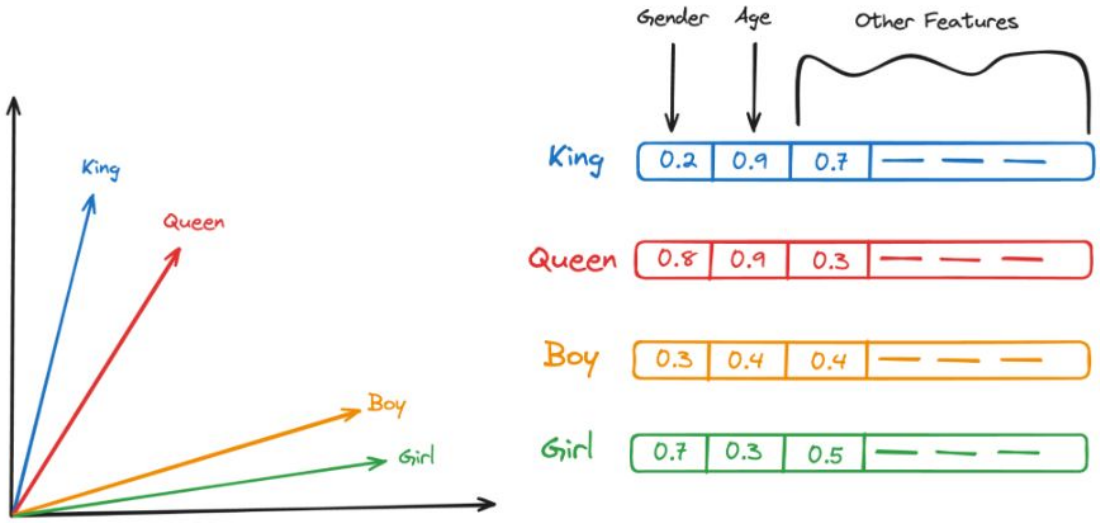
1️⃣ **Construcción de la base de conocimientos:** primero debemos procesar y almacenar la información que va a usar nuestra RAG. Para esto:

- Se recopilan documentos en textos planos, PDFs, bases de datos, etc.
- Se dividen en fragmentos pequeños (*chunks*) para mejorar la búsqueda.
- Se convierten en vectores numéricos usando un **modelo de embeddings** (Ej: OpenAI text-embedding-ada-002, Sentence-BERT).
- Se almacenan en una base de datos vectorial como **FAISS, Chroma o Pinecone**.



Que son los embeddings?

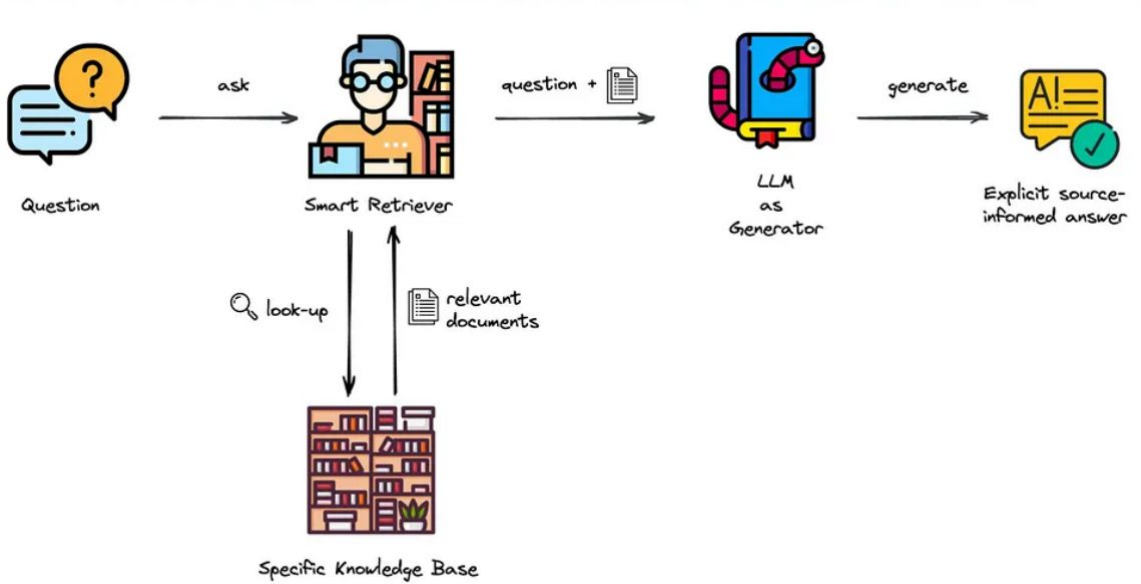
Un *embedding* es una representación matemática del texto en un espacio multidimensional. Convierte palabras o frases en vectores que capturan su significado y relaciones semánticas. En este espacio, términos con significados similares estarán más próximos entre sí. Los modelos de embeddings aprenden estas representaciones analizando el contexto en el que aparecen las palabras, permitiendo comprender mejor su relación con otras.



2️⃣ Búsqueda eficiente de información (Retrieval)

🔍 Cuando un usuario hace una pregunta:

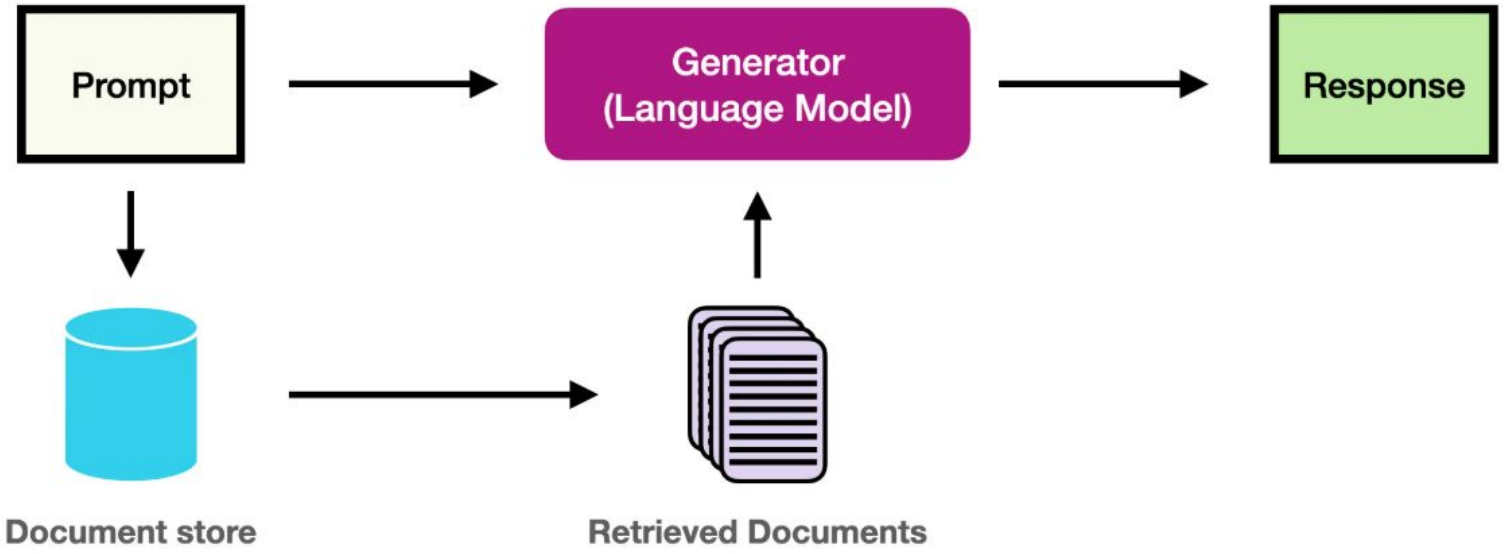
- Se convierte la pregunta en un vector numérico usando el mismo modelo de embeddings.
- Se busca en la base de datos vectorial los fragmentos más relevantes.
- Se obtiene un conjunto de documentos que contienen información útil.



3️⃣ Generación de respuesta usando LLM (Generation)

📝 Con los documentos recuperados:

- Se construye un **prompt** para un modelo de lenguaje (GPT-4, Mixtral o Llama).
- Se le pasa la pregunta junto con el contexto recuperado.
- El modelo genera una respuesta basada en la información proporcionada.



Conclusiones

🎯 **Beneficios de RAG**

✅ **Respuestas más precisas**: Se evita la "alucinación" de los modelos generativos.
✅ **Información actualizada**: Puedes agregar nuevos datos sin volver a entrenar un modelo desde cero.
✅ **Menos costos**: No necesitas un modelo grande con todo el conocimiento, solo una base de datos eficiente.

Una aplicación RAG combina lo mejor de dos mundos:

1. **Búsqueda eficiente** → Encuentra los mejores fragmentos de información.
2. **Generación de texto** → Usa un LLM para crear respuestas naturales y bien estructuradas.

Es una técnica fundamental en **aplicaciones de chatbots inteligentes, sistemas de recomendación y asistentes virtuales**.