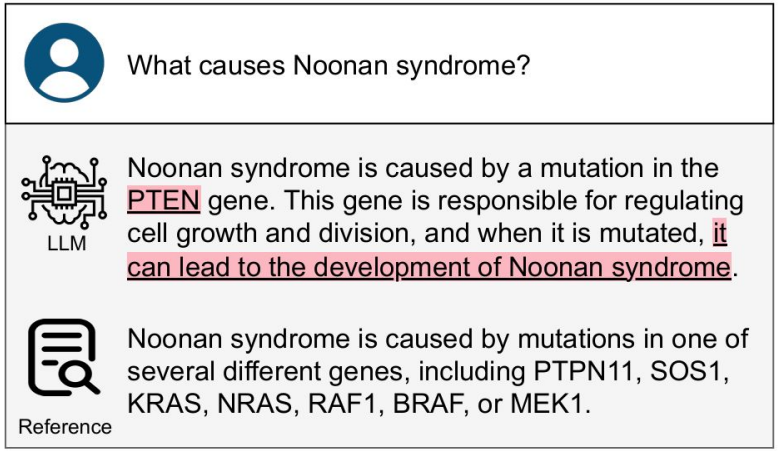
Retrieval-Augmented Generation

Fabián Villena

Introducción

Los grandes modelos de lenguaje, a pesar de su buen rendimiento en un gran número de tareas generativas, aún enfrentan el desafío de las alucinaciones, especialmente en dominios específicos y tareas que requieren un alto nivel de conocimiento.

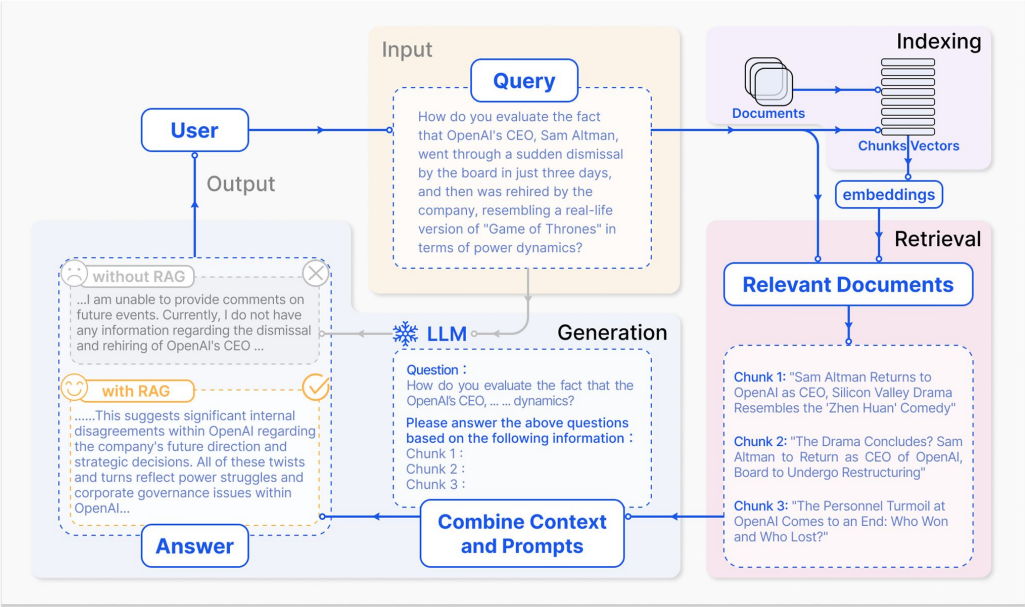
Las alucinaciones se refieren a la generación de información que parece plausible pero que es incorrecta o no tiene sentido.



Descripción general de la solución

La Retrieval-Augmented generation (RAG) es un método que mejora el rendimiento de los LLM al recuperar pedazos de información relevante desde conocimiento externo a través de cálculos de similaridad semántica.

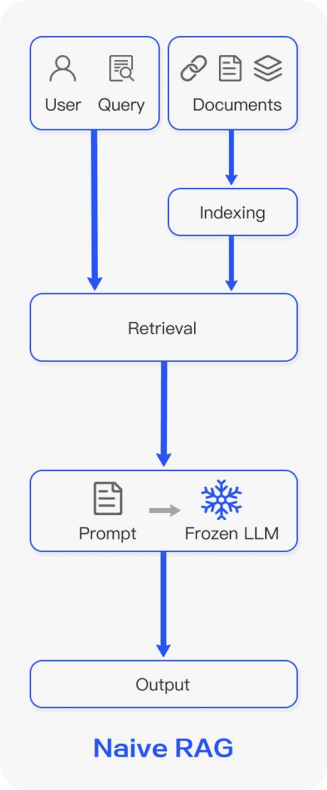
Al referenciar conocimiento externo, este método reduce el problema de las alucinaciones.



RAG Naïve

Esta implementación sigue un proceso tradicional simple que incluye indización, recuperación y generación.

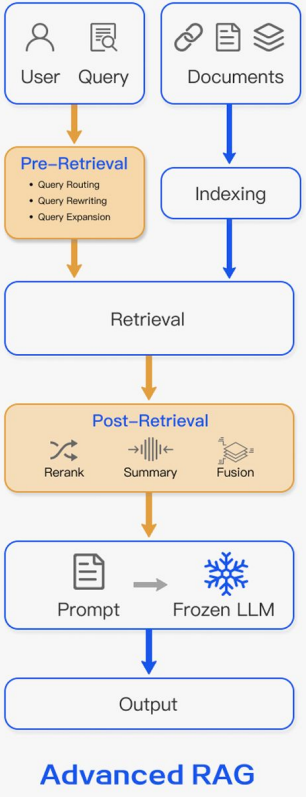
- La indización toma pedazos de documentos y los vectoriza para almacenar los embeddings en una base de datos vectorial.
- En el proceso de recuperación, la consulta del usuario se vectoriza y se extraen los K documentos más cercanos.
- En la generación, el modelo utiliza la consulta del usuario y los pedazos de documentos recuperados para contextualizar la generación.



RAG avanzada

En esta implementación se busca mejorar el paso de recuperación de información a través de dos procesos:

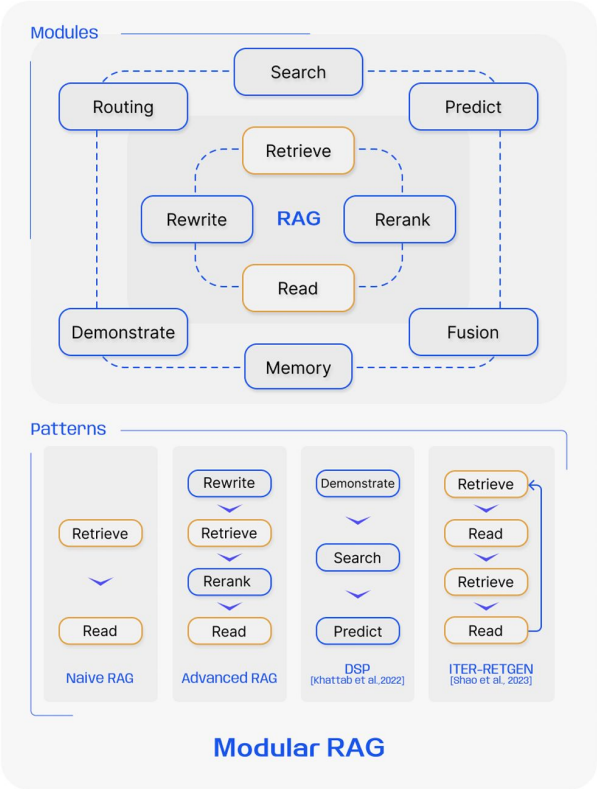
- Pre-recuperación: En este proceso se busca mejorar la calidad del contenido indizado y mejorar la consulta del usuario para hacerla más adecuada para el proceso de búsqueda.
- Pos-recuperación: Se busca incorporar de la mejor manera el contenido recuperado a través de los procesos de re-ranking y destilación del contenido recuperado.



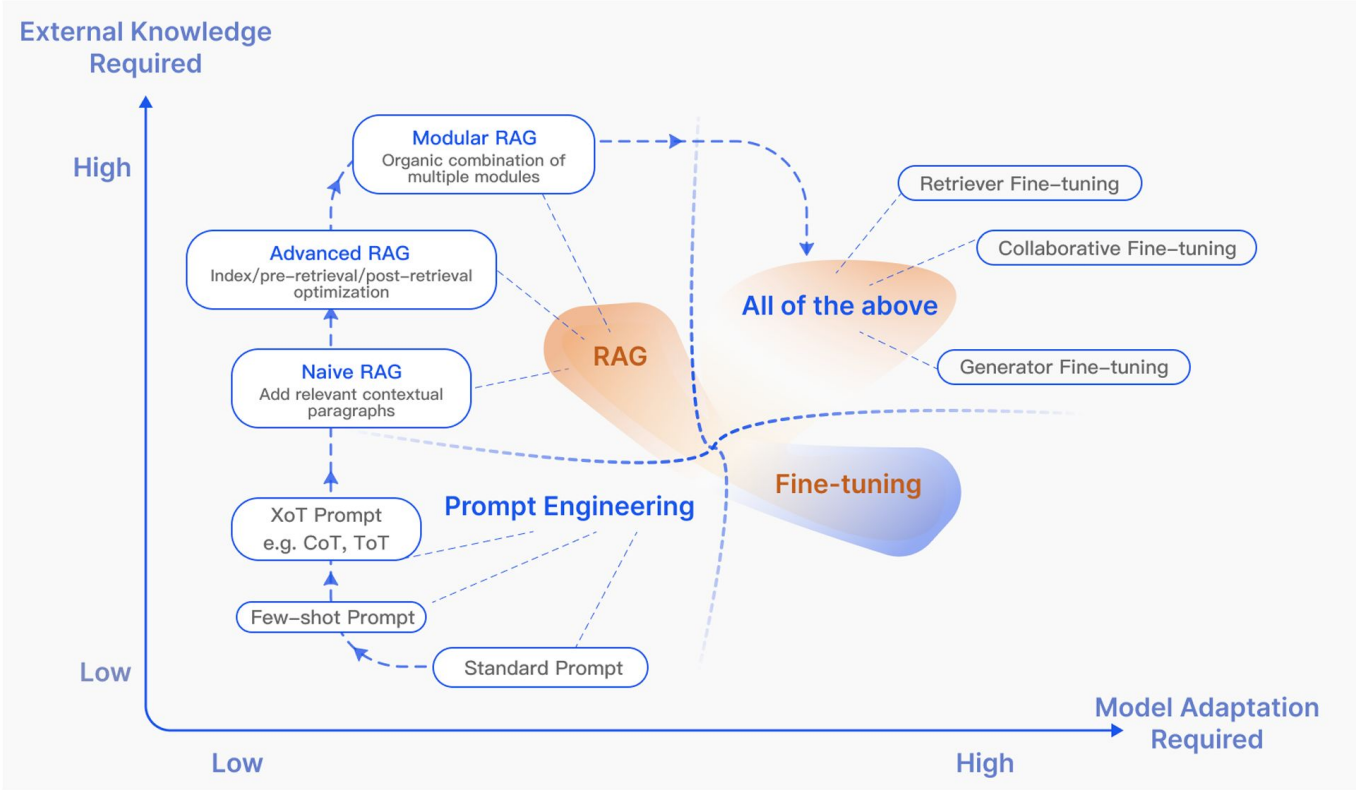
RAG modular

Esta implementación busca mejorar el rendimiento más allá de las implementaciones naïve y avanzadas.

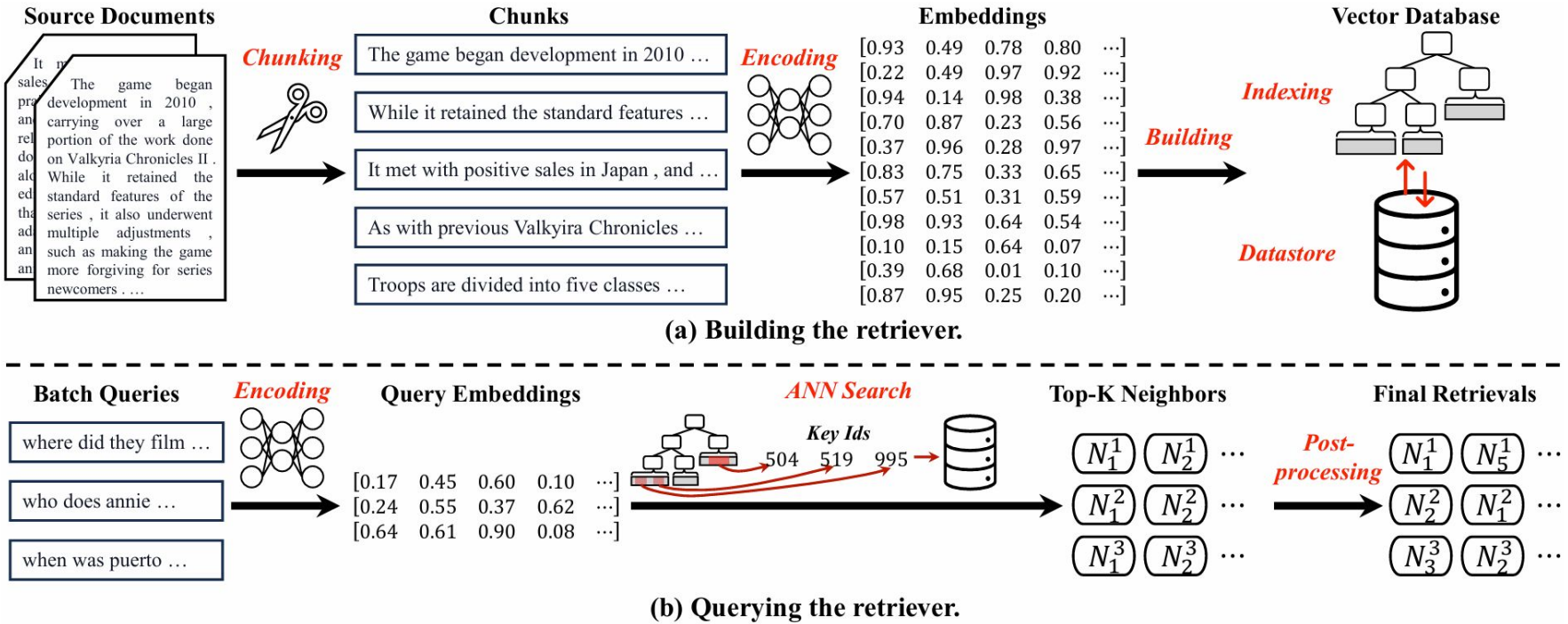
- La adición de módulos específicos como de Búsqueda, Memoria y Predicción permiten una generación más apegada a las necesidades de información precisa.
- El desarrollo de nuevos patrones que implementan distintas configuraciones de los módulos, permite que los modelos se ajusten a distintos tipos de tareas.



RAG vs. Fine-Tuning



Recuperador



Fragmentación

El proceso de fragmentación consiste en la segmentación de grandes piezas de texto en piezas más pequeñas. El objetivo de la fragmentación busca considerar el compromiso entre piezas pequeñas que entregan poca información y piezas largas que pueden ser muy difíciles de representar. Se destacan tres procesos de fragmentación:

- Fragmentación de largo fijo en donde se determina un largo constante para dividir el texto.
- Fragmentación semántica en donde se divide el texto por puntos o saltos de línea.
- Fragmentación basada en el contenido en donde se segmenta el texto según sus características específicas como secciones y subsecciones.

Codificación

En este proceso los fragmentos de los documentos se transforman en una representación numérica de su contenido.

Dentro de los métodos de codificación se encuentran:

- Codificación dispersa: Para este tipo de representaciones se utilizan métodos como bag-of-words o TF-IDF.
- Codificación densa: Este tipo de representaciones se genera principalmente a través de redes neuronales. Se destacan los métodos basados en BERT o LLMs.

Consulta

Para alinear la consulta con el espacio vectorial generado anteriormente, la consulta se representa utilizando el mismo modelo de representación utilizado y después se puede realizar el proceso de búsqueda.

Después de representar la consulta, se buscan los k vecinos más cercanos a la consulta para retornar los documentos asociados a esos vecinos.

Finalmente se puede realizar un proceso de reranking para reordenar los documentos en función de la tarea a resolver.

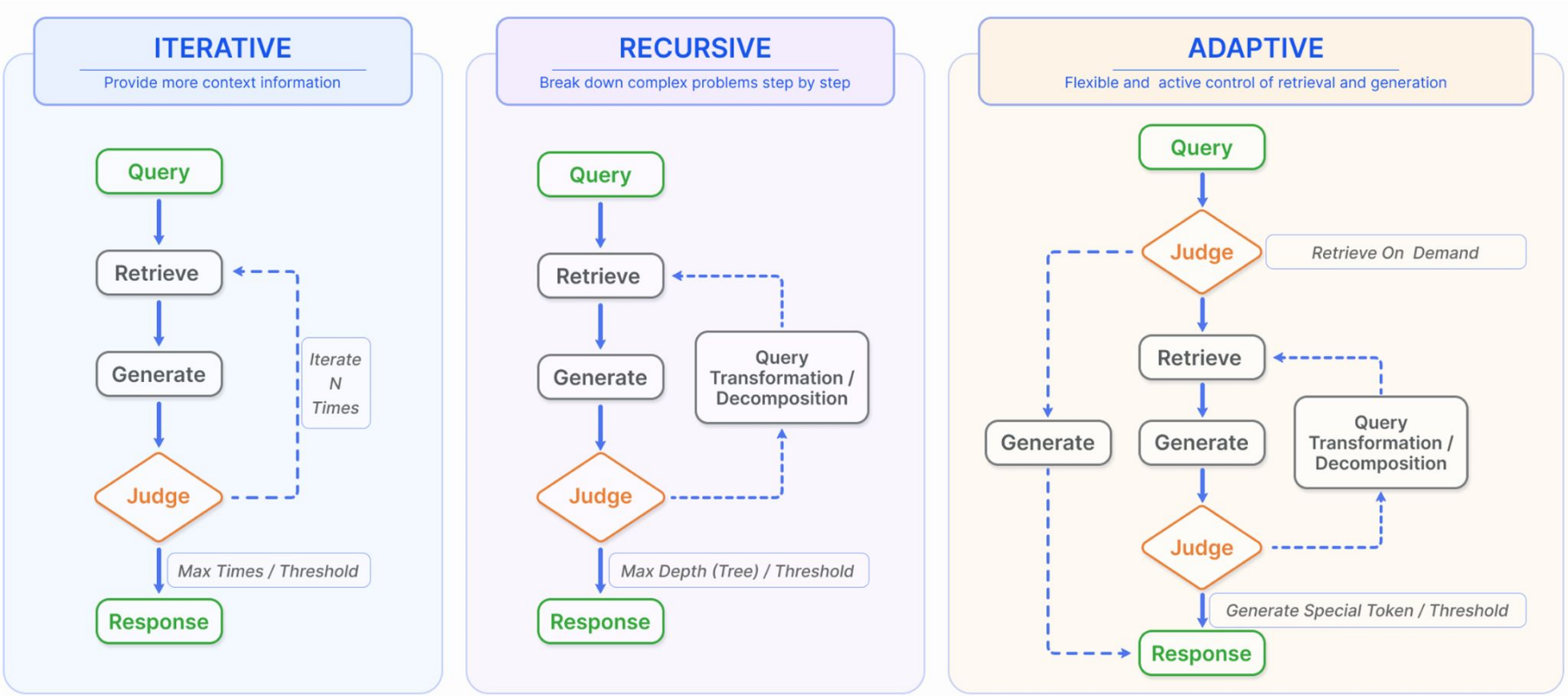
Generación

En el proceso de generación de la respuesta, los documentos recuperados se deben incluir como contexto para que el modelo pueda generar contenido.

Normalmente los documentos no se entregan de manera directa, sino que se modifica su contenido:

- Curación de contenido: La información redundante puede degradar a generación, por lo que el contenido recuperado se modifica.
  - Reranking
  - Compresión del contenido
- Afinamiento de modelos: Los modelos se pueden ajustar para adaptarse a las necesidades de los usuarios y los recuperadores.

Aumento


ITERATIVE
Provide more context information
Query
Retrieve
Generate
Judge
Iterate N Times
Max Times / Threshold
Response
RECURSIVE
Break down complex problems step by step
Query
Retrieve
Generate
Judge
Query Transformation / Decomposition
Max Depth (Tree) / Threshold
Response
ADAPTIVE
Flexible and active control of retrieval and generation
Query
Judge
Retrieve On Demand
Retrieve
Generate
Generate
Query Transformation / Decomposition
Judge
Generate Special Token / Threshold
Response