Procesamiento de Lenguaje Natural Avanzado

RAG: Paradigmas, tecnología y evaluación

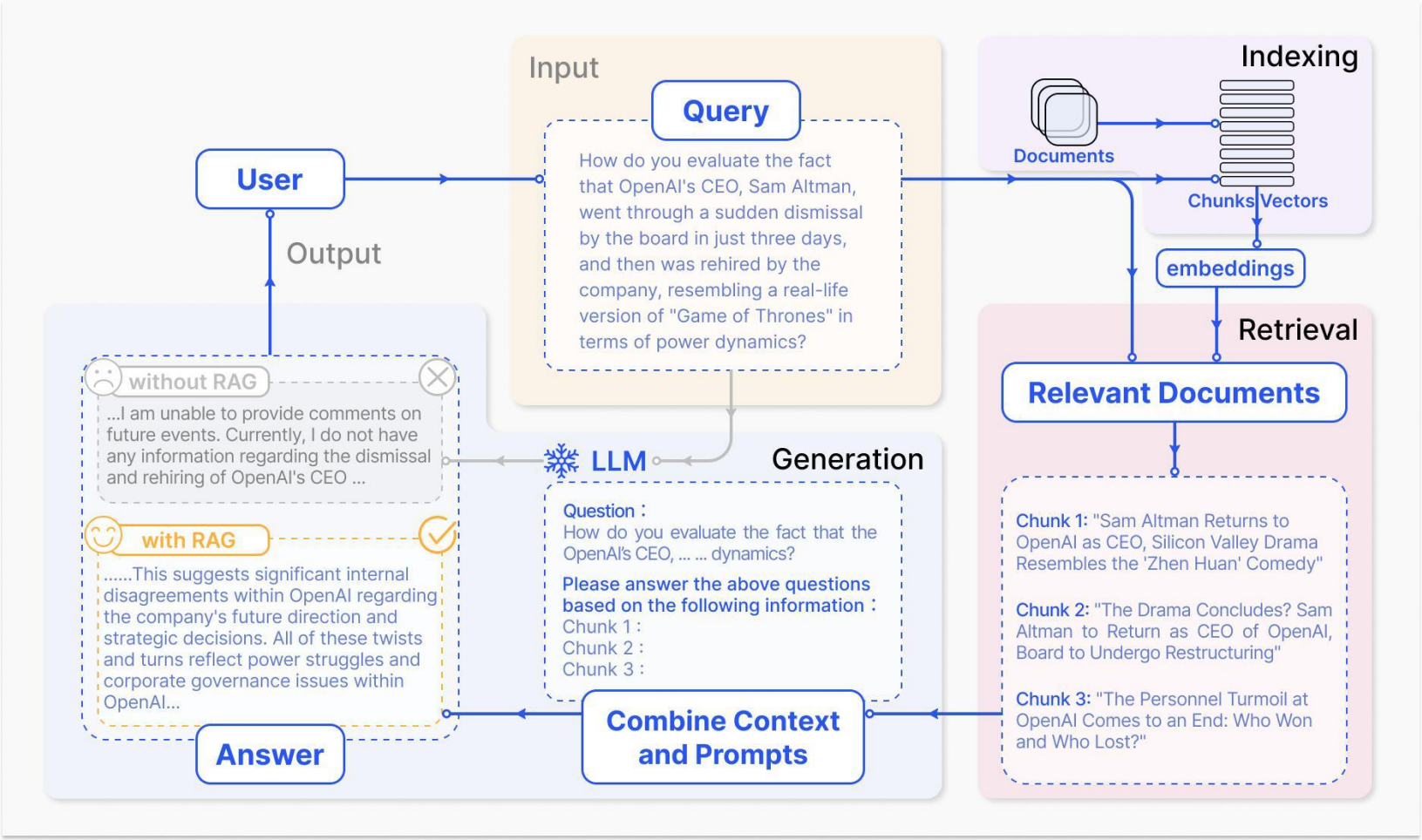
Dra. Helena Gómez Adorno
helena.gomez@iimas.unam.mx

Dr. Fazlourrahman Balouchzahi
fbalouc@iimas.unam.mx

Correo del curso:
pln.cienciadedatos@gmail.com

Generación aumentada por Recuperación (RAG)

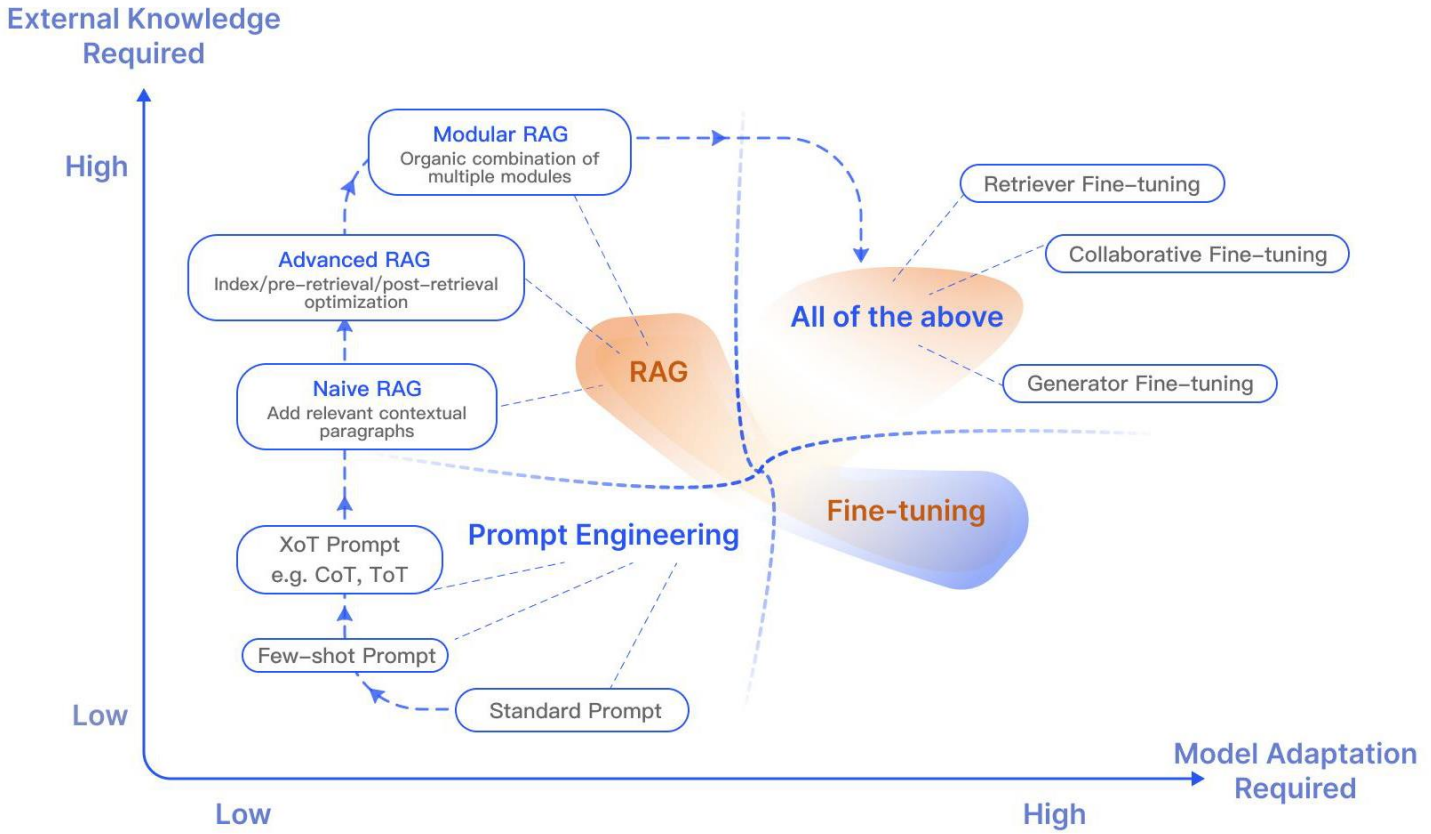
- Al responder preguntas o generar texto, primero **recupera información relevante** de un gran conjunto de documentos y luego los LLM generan respuestas basadas en esa información.
- Al incorporar una **base de conocimiento externa**, no es necesario reentrenar todo el modelo grande para cada tarea específica.
- El modelo RAG es especialmente adecuado para tareas **intensivas en conocimiento**.



Conocimiento simbólico o conocimiento paramétrico

Formas de optimizar LLMs:

- Ingeniería de promts
- Generación aumentada por recuperación
- Instruct / Fine-tuning



RAG vs Fine-tuning

Feature Comparison	RAG	Fine-Tuning
Knowledge Updates	Directly updating the retrieval knowledge base ensures that the information remains current without the need for frequent retraining, making it well-suited for dynamic data environments.	Stores static data, requiring retraining for knowledge and data updates.
External Knowledge	Proficient in leveraging external resources, particularly suitable for accessing documents or other structured/unstructured databases.	Can be utilized to align the externally acquired knowledge from pretraining with large language models, but may be less practical for frequently changing data sources.
Data Processing	Involves minimal data processing and handling.	Depends on the creation of high-quality datasets, and limited datasets may not result in significant performance improvements.
Model Customization	Focuses on information retrieval and integrating external knowledge but may not fully customize model behavior or writing style.	Allows adjustments of LLM behavior, writing style, or specific domain knowledge based on specific tones or terms.
Interpretability	Responses can be traced back to specific data sources, providing higher interpretability and traceability.	Similar to a black box, it is not always clear why the model reacts a certain way, resulting in relatively lower interpretability.
Computational Resources	Depends on computational resources to support retrieval strategies and technologies related to databases. Additionally, it requires the maintenance of external data source integration and updates.	The preparation and curation of high-quality training datasets, defining fine-tuning objectives, and providing corresponding computational resources are necessary.
Latency Requirements	Involves data retrieval, which may lead to higher latency.	LLM after fine-tuning can respond without retrieval, resulting in lower latency.
Reducing Hallucinations	Inherently less prone to hallucinations as each answer is grounded in retrieved evidence.	Can help reduce hallucinations by training the model based on specific domain data but may still exhibit hallucinations when faced with unfamiliar input.
Ethical and Privacy Issues	Ethical and privacy concerns arise from the storage and retrieval of text from external databases.	Ethical and privacy concerns may arise due to sensitive content in the training data.

Escenarios donde es aplicable RAG

- **Distribución de cola larga de los datos**: Cuando la información relevante está dispersa o es poco frecuente, y los modelos tradicionales tienen dificultad para acceder a ella.
- **Actualizaciones frecuentes de conocimiento**: Cuando el dominio de aplicación requiere información actualizada constantemente, sin necesidad de reentrenar el modelo base.
- **Respuestas que requieren verificación y trazabilidad**: Cuando es fundamental citar fuentes, auditar el origen de la información o justificar las respuestas generadas.
- **Conocimiento especializado de dominio:** En sectores como medicina, derecho o ingeniería, donde se necesita precisión técnica y acceso a bases de conocimiento específicas.
- **Preservación de la privacidad de los datos:** Cuando la información sensible debe mantenerse en entornos controlados y recuperarse de forma segura, sin exponerla durante el entrenamiento del modelo.

Q&A
RETRO (Borgeaud et al2021)
REALM (Gu et al, 2020)
ATLAS (Izacard et al, 2023)

Fact Checking
RAG (Lewis et al, 2020)
ATLAS (Izacard et al, 2022)
Evi. Generator (Asai et al, 2022a)

Dialog
BlenderBot3 (Shuster et al.2022)
Internet-augmented generation
(Komeili et a., 2022)

Summary
FLARE (Jiang et al, 2023)

Machine Translation
kNN-MT (Khandelwal et al., 2020)TRIME-MT (Zhong et al., 2022)

Code Generation
DocPrompting (Zhou et al., 2023
Natural ProverWelleck et al., 2022)

Natural Language Inference
kNN-Prompt (Shi et al., 2022)
NPM (Min et al., 2023)

Sentiment analysis
kNN-Prompt (Shi et al., 2022)NPM (Min et al., 2023)

Commonsense reasoning
Raco (Yu et al, 2022)

RAG: Cambio de paradigma

Paso 1: Indexación

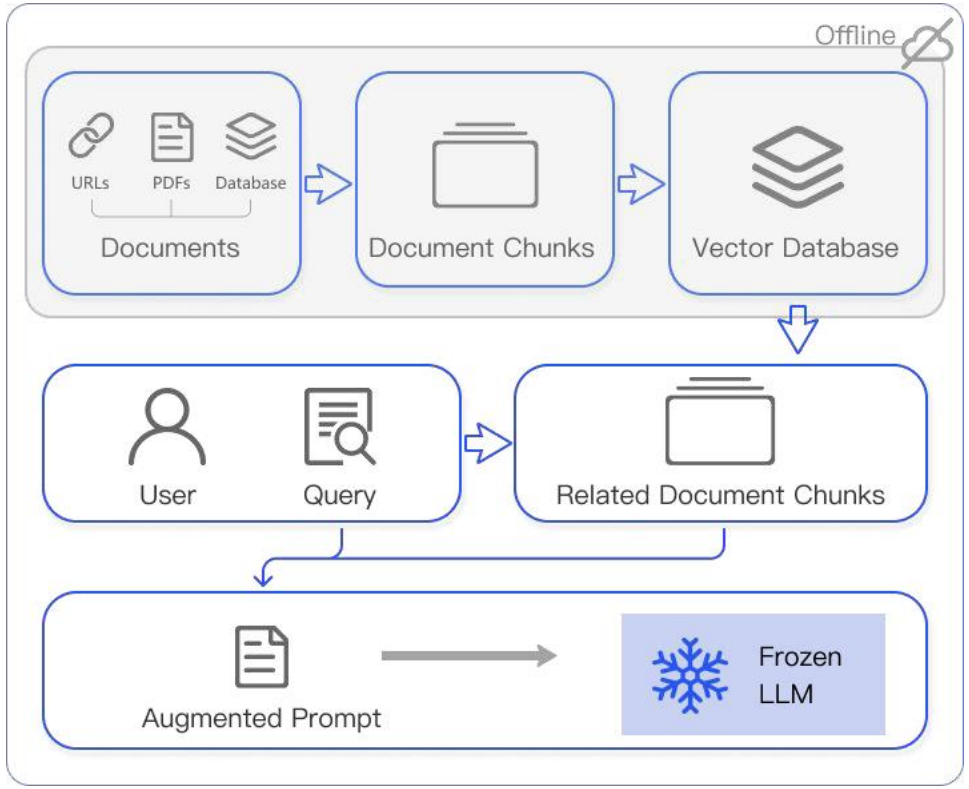
- Dividir el documento en fragmentos (chunks) uniformes, donde cada fragmento es una porción del texto original.
- Utilizar un modelo de codificación (encoding model) para generar un embedding (representación vectorial) para cada fragmento.
- Almacenar el embedding de cada bloque en una base de datos vectorial.

Paso 2: Recuperación

- Recuperar los k documentos más relevantes mediante búsqueda por similitud vectorial.

Paso 3: Generación

- Combinar la consulta original con los textos recuperados y alimentarlos en un Modelo de Lenguaje Grande (LLM) para obtener la respuesta final.



Naive RAG

Advanced RAG

Modular RAG

RAG: Cambio de paradigma

Optimización del índice → Proceso de pre-recuperación → Recuperación → Proceso de post-recuperación→ Generación

Optimización de la indexación de datos:

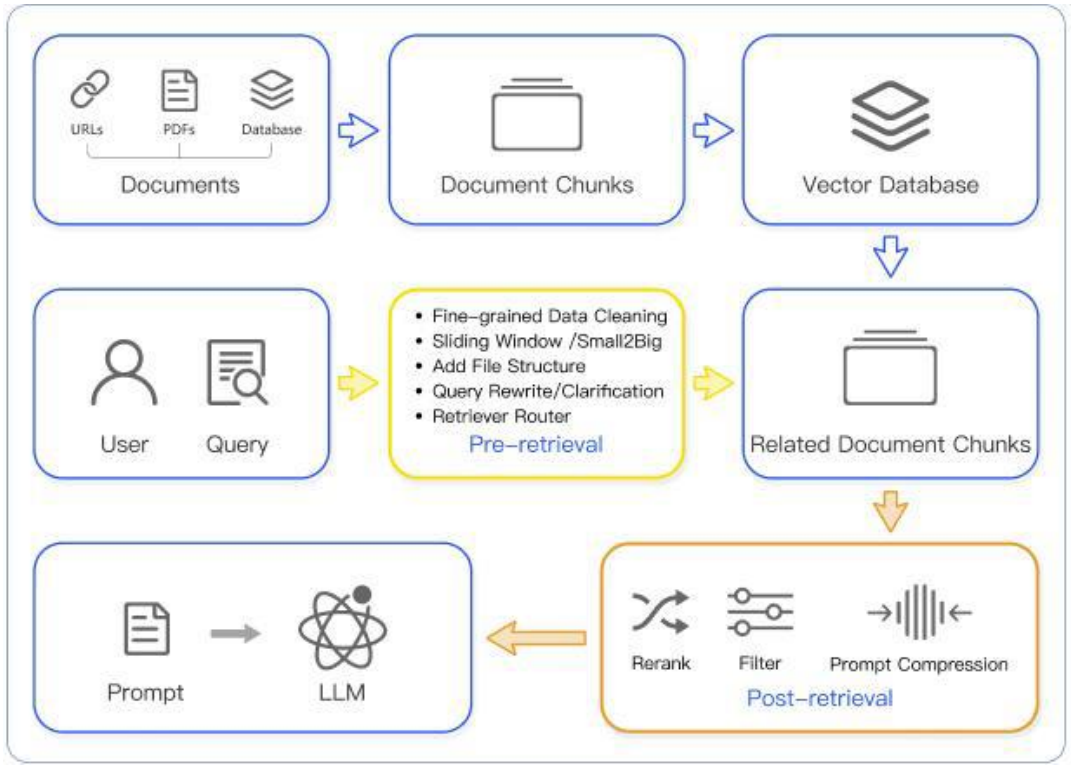
- Ventana deslizante (sliding window), segmentación de grano fino (fine-grained segmentation), adición de metadatos.

Proceso de pre-recuperación:

- Enrutamiento de consultas (retrieve routes), generación de resúmenes, reformulación de consultas (rewriting) y evaluación de confianza (confidence judgment).

Proceso de post-recuperación:

- Reordenamiento (re-ranking), filtrado y compresión del contenido recuperado.

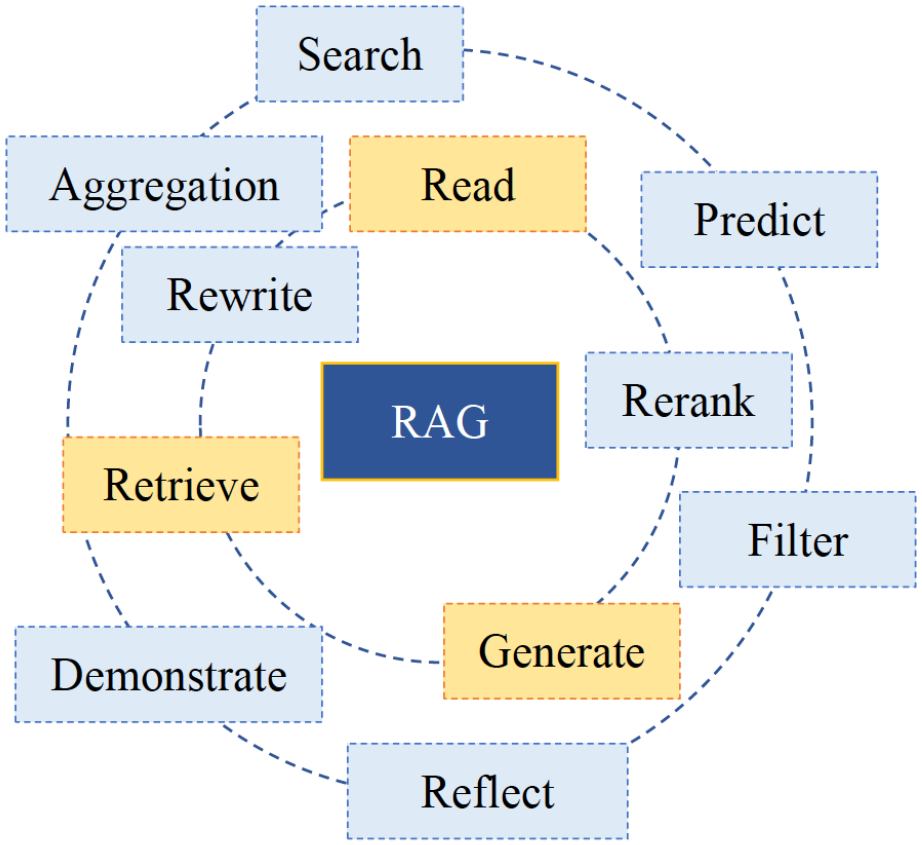


Naive RAG

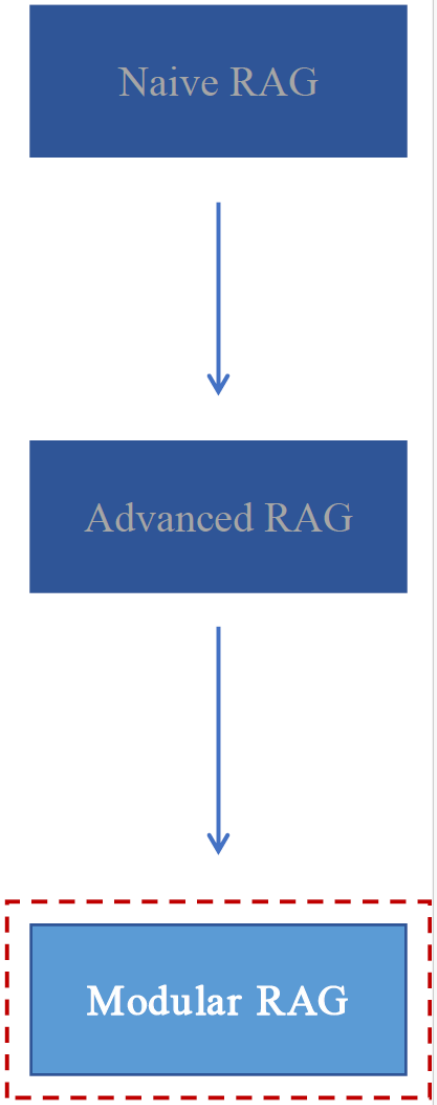
Advanced RAG

Modular RAG

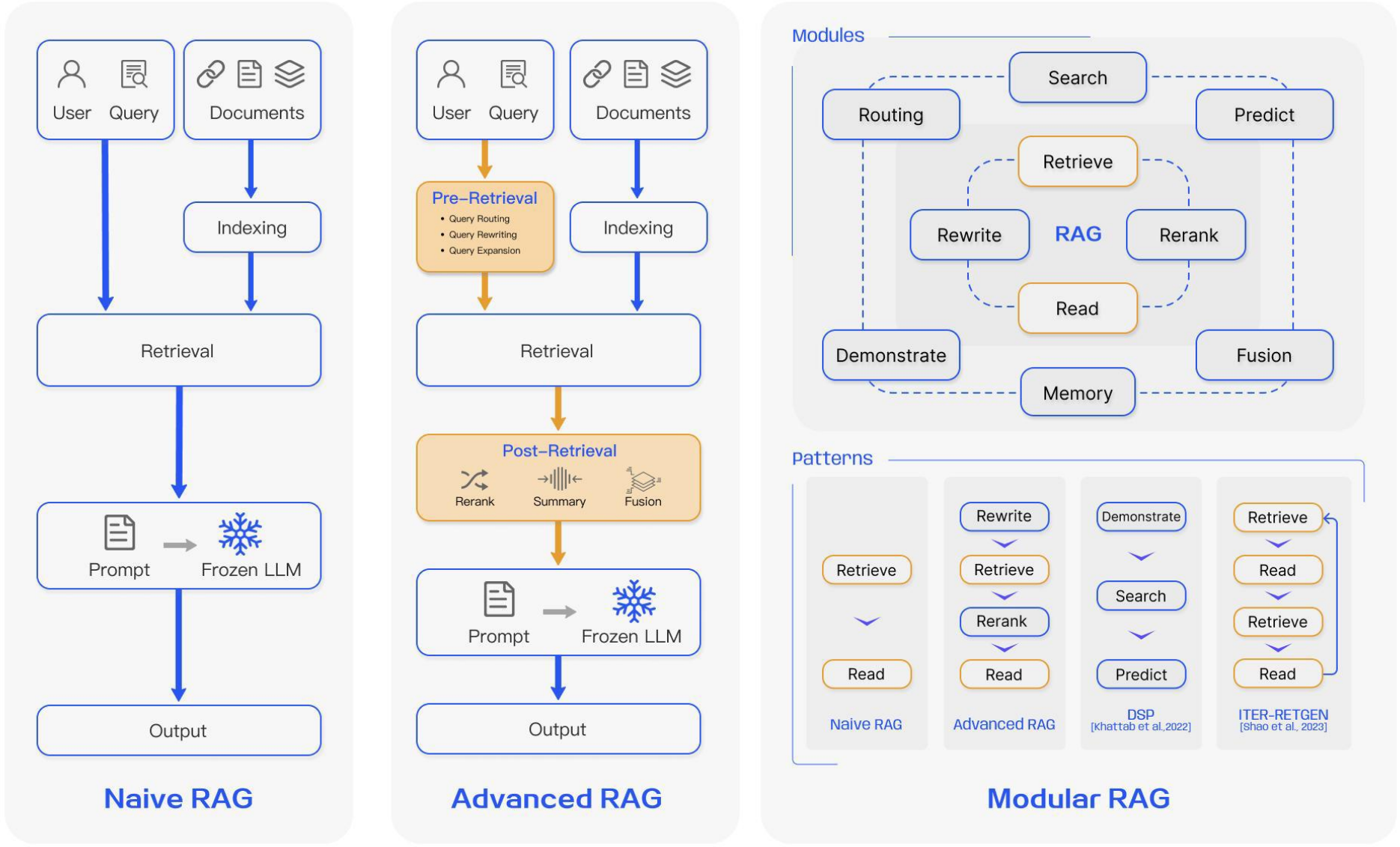
RAG: Cambio de paradigma



Pattern				
Naive RAG	Read	Retrieve	Generate	
DSP (2022)	Demonstrate	Search	Predict	Generate
Rewrite-Retrieve-Read (2023)	Rewrite	Retrieve	Read	
Retrieve-then-read (2023)	Retrieve	Read	Generate	



Comparación de paradigmas



Las tres preguntas clave de RAG

¿Qué recuperar?	¿Cómo recuperar?	¿Cómo generar?
• Token • Frase • Chunk • Parrafo • Entidad • Grafo de conocimiento	• Búsqueda simple • Cada token • Cada N tokens • Busqueda Adaptativa	• Capa de entrada / datos • Capa intermedia / del modelo • Capa de salida / predicción

Other Issues

Augmentation stage:
- Pre-training
- Fine-tuning
- Inference

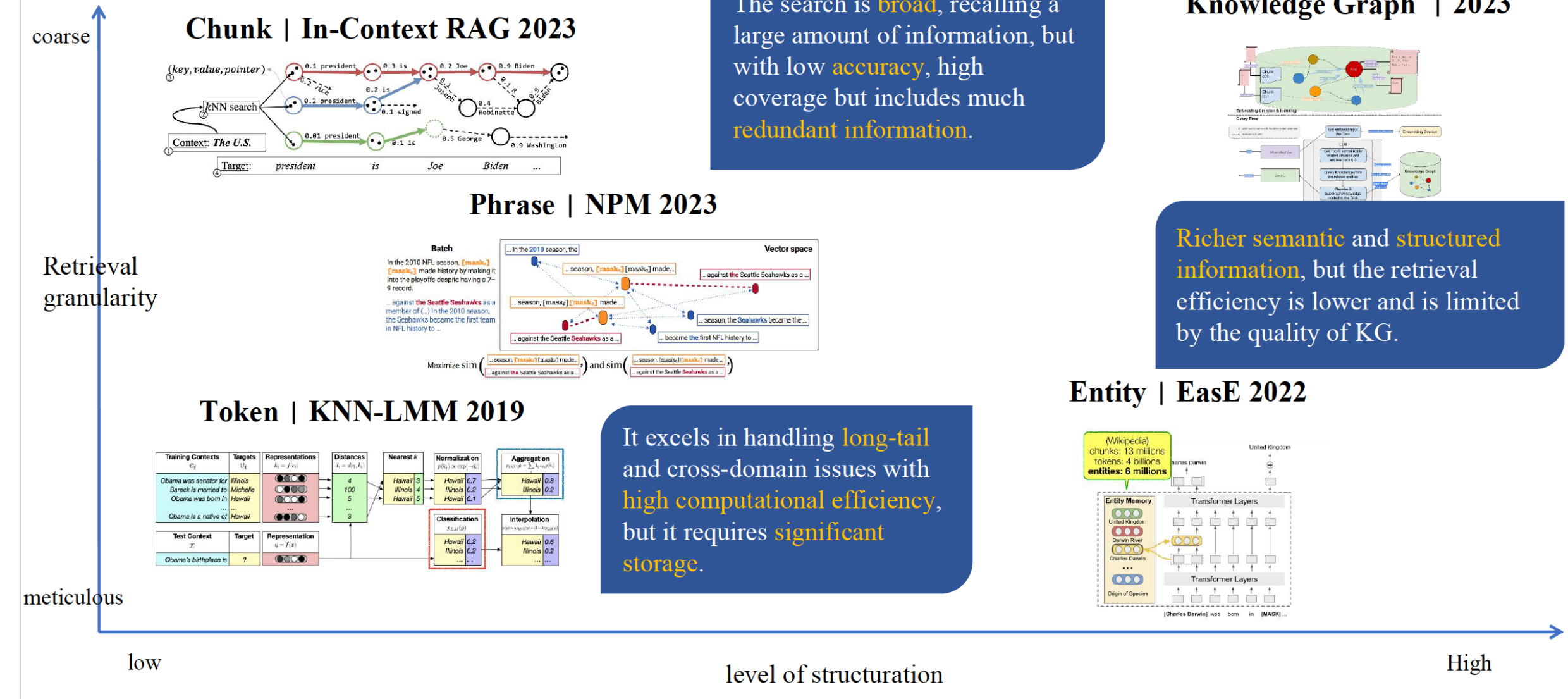
Retrieval choice:
- BERT
- Roberta
- BGE
- .....

Model Collaboration

Scale selectionz

Generation choice:
- GPT
- Llama
- T5
- .....

¿Qué recuperar?



Chunk | In-Context RAG 2023

The search is broad, recalling a large amount of information, but with low accuracy, high coverage but includes much redundant information.

Knowledge Graph | 2023

Richer semantic and structured information, but the retrieval efficiency is lower and is limited by the quality of KG.

Phrase | NPM 2023

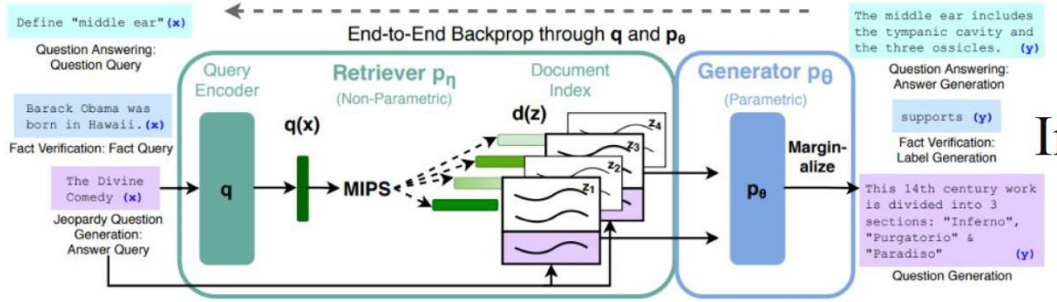
Token | KNN-LMM 2019

It excels in handling long-tail and cross-domain issues with high computational efficiency, but it requires significant storage.

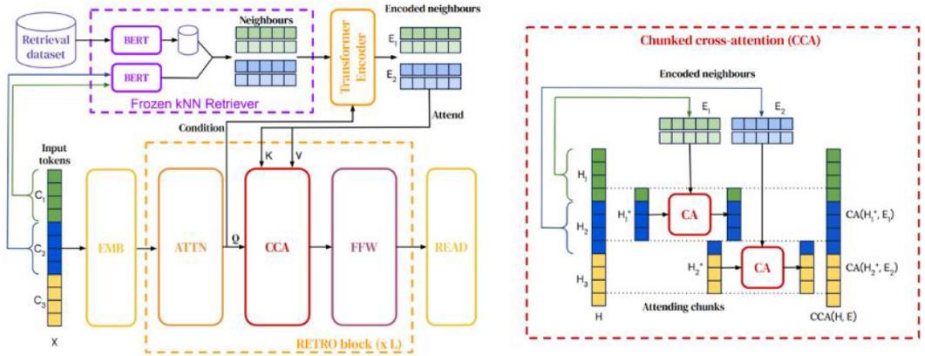
Entity | EasE 2022

¿Cómo usar el contexto recuperado?

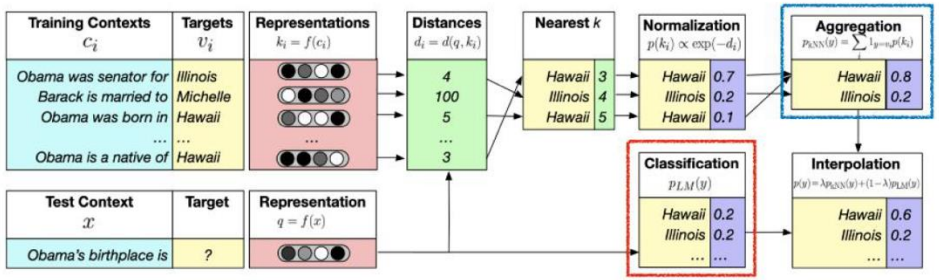
Integrar la información recuperada en distintas capas del modelo de generación durante el proceso de inferencia.



Input / Date layer



Model / Interlayer



Output /Prediction layer

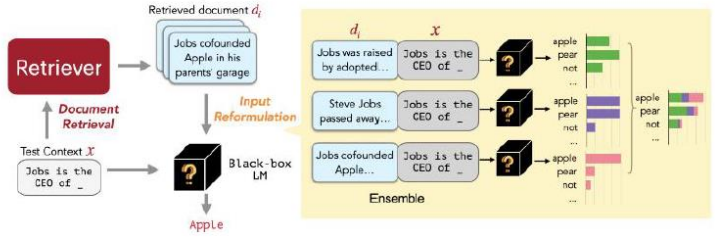
¿Cuándo recuperar?

Alta eficiencia, pero baja relevancia de los documentos recuperados.

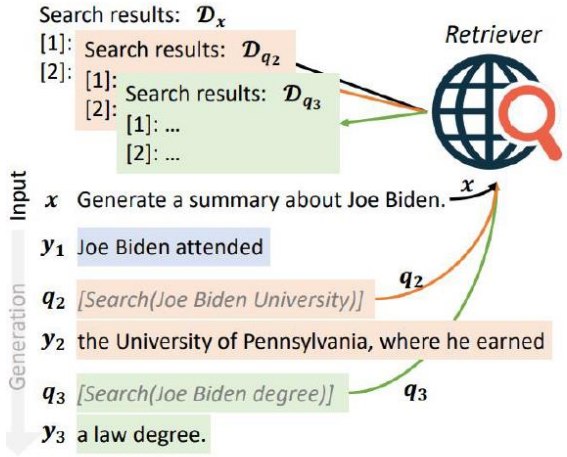
Equilibrar la eficiencia y la información podría no producir la solución óptima.

Una gran cantidad de información, con baja eficiencia y contenido redundante.

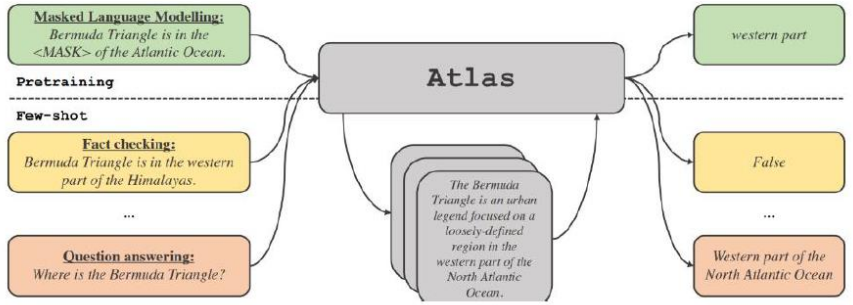
Once | Replug 2023



Adaptive | Flare 2023



Every N Tokens | Atlas 2023



Conducting once search during the reasoning process.

Adaptively conduct the search.

Retrieve once for every N tokens generated.

Low — **Retrieval frequency** — High

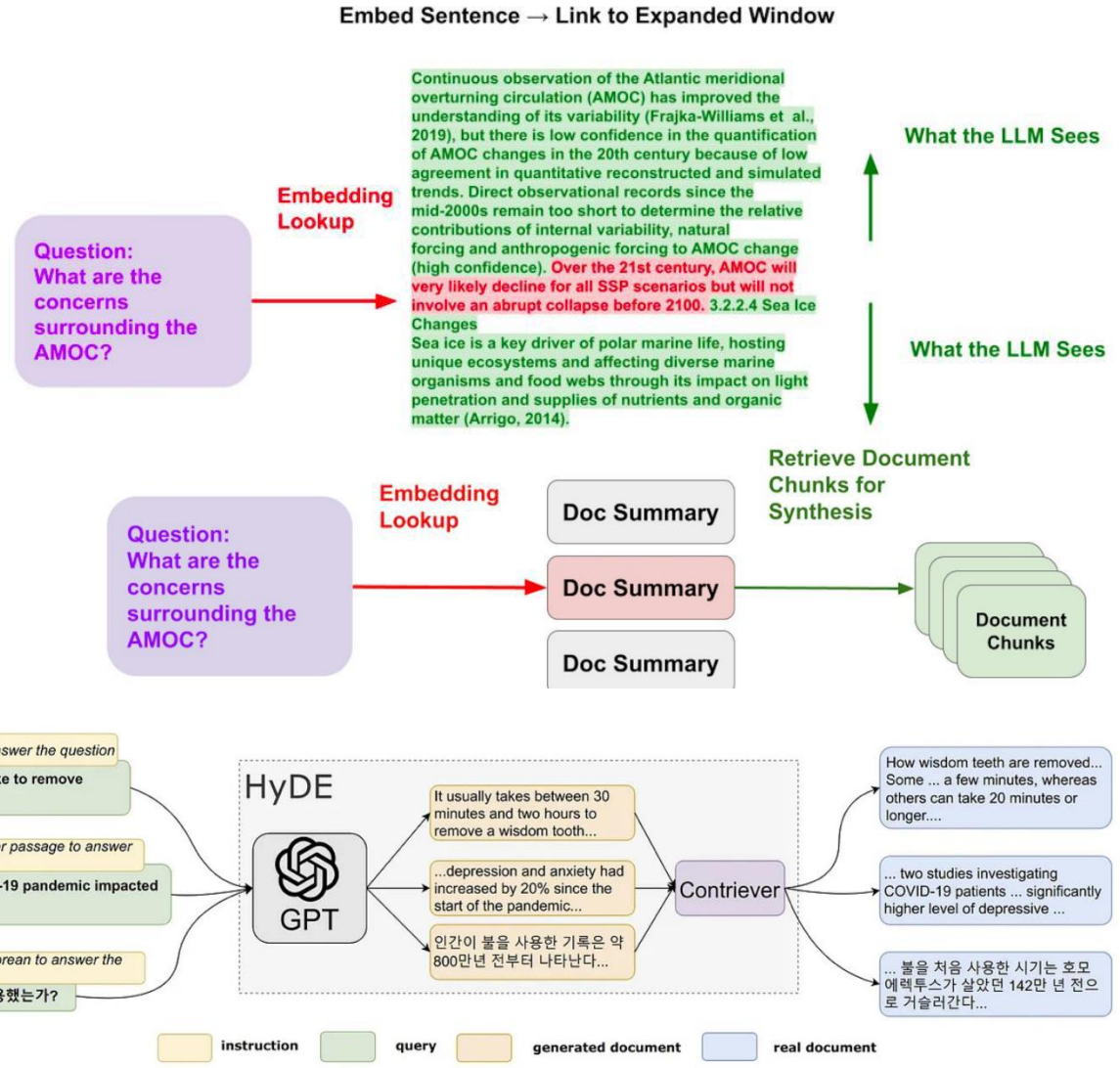
Técnicas para mejorar RAG — Optimización de la indexación de datos

Optimización de fragmentos (Chunk Optimization)

- Small-2-Big (De pequeño a grande)
- Sliding Window (Ventana deslizante)
- Summary (Recuperación mediante resúmenes)

Adding Metadata (Adición de metadatos)
Página, Documento, Título, Resultados, Autor

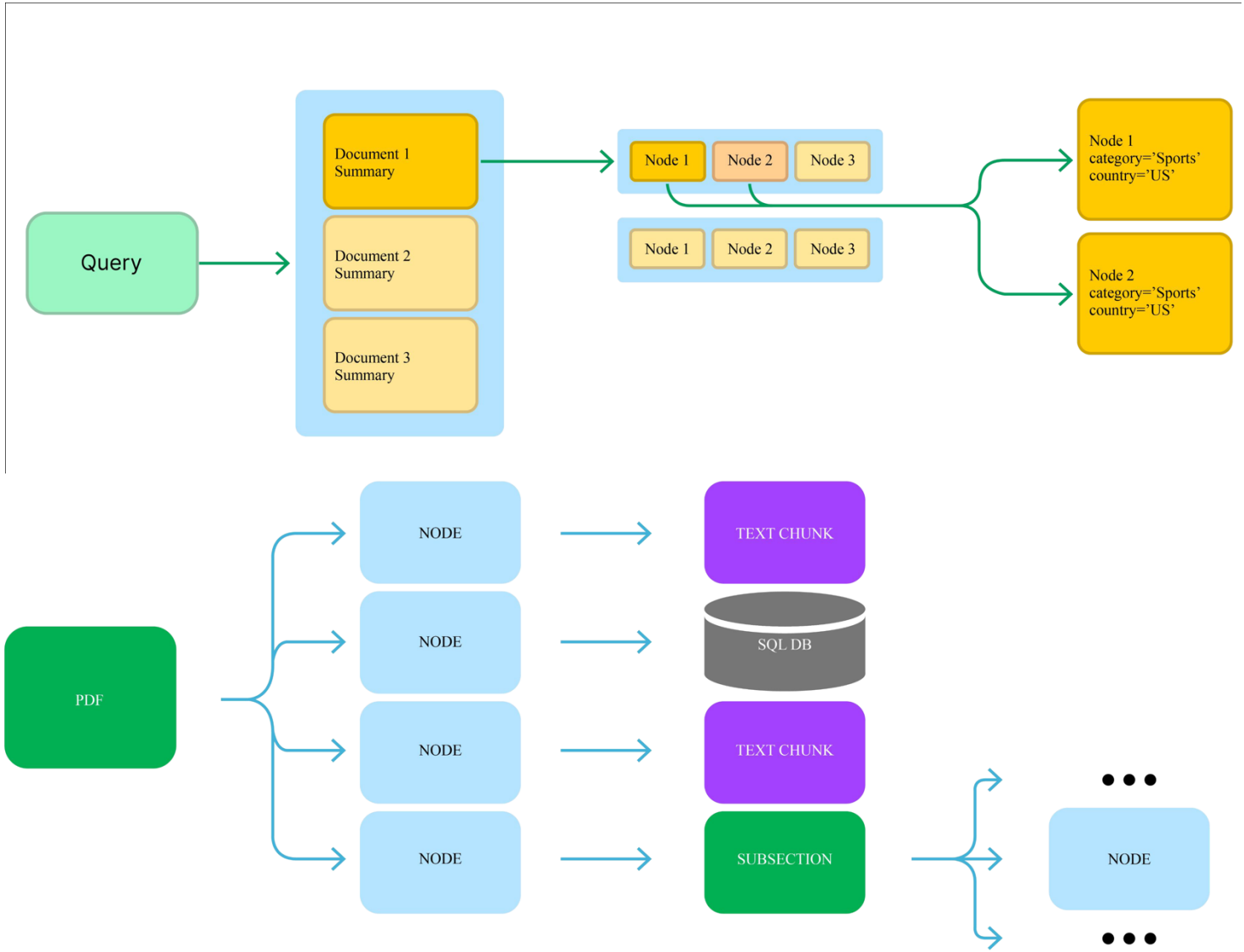
Metadata Filtering/Enrichment (Filtrado y enriquecimiento por metadatos)
- Pseudo Metadata Generation (Generación de pseudo-metadatos)
- Filtrado



Técnicas para mejorar RAG — Corpus estructurado

Resumen → Documento
Reemplazar la recuperación de documentos por recuperación de resúmenes, no solo recuperando los nodos más directamente relevantes, sino también explorando nodos adicionales asociados con dichos nodos.

Documento → Objetos embebidos
Los documentos contienen objetos embebidos (como tablas y gráficos); primero se recuperan los objetos de referencia de entidades y luego se consultan los objetos subyacentes, como bloques de documentos, bases de datos y subnodos.



Técnicas para mejorar RAG — Grafo de conocimiento como fuente de recuperación

GraphRAG

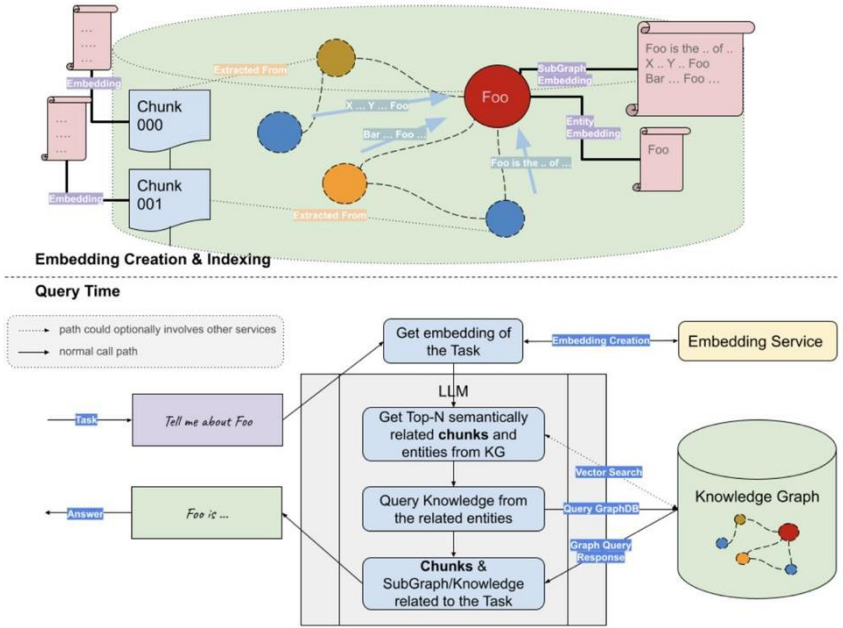
➤ Extraer entidades de la consulta del usuario, construir un subgrafo para conformar el contexto y, finalmente, introducirlo en el modelo grande para la generación.

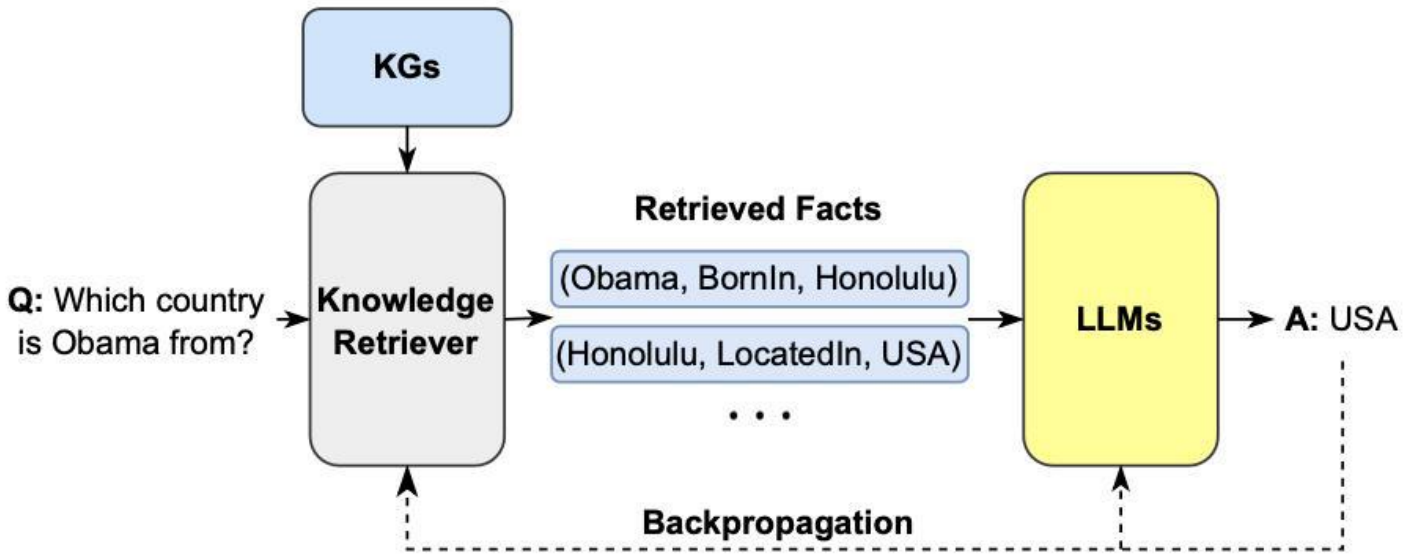
Implementación

➤ Utilizar un LLM (u otros modelos) para extraer las entidades clave de la pregunta.

➤ Recuperar subgrafos en función de dichas entidades, explorando hasta una profundidad determinada (por ejemplo, 2 saltos/hops o más).

➤ Utilizar el contexto obtenido para generar respuestas mediante el LLM.

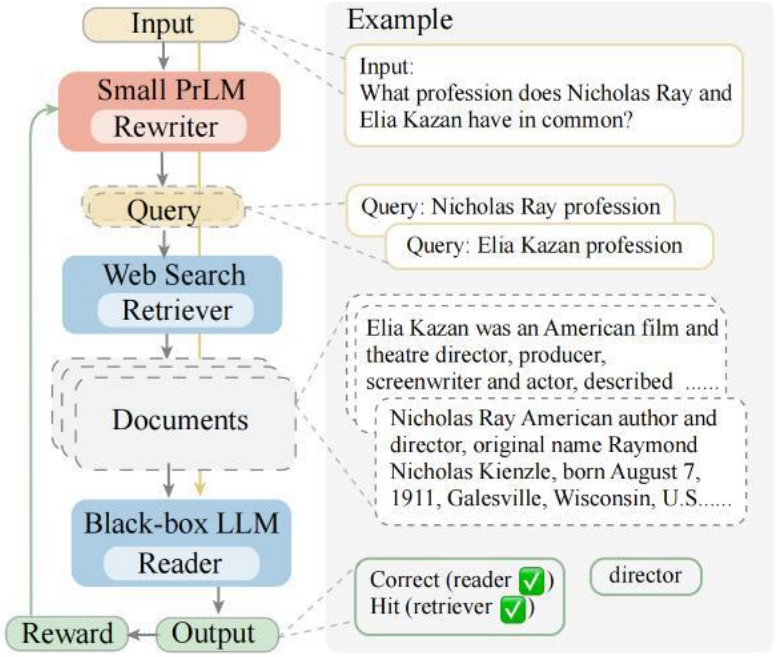




Técnicas para mejorar RAG — Optimización de consultas

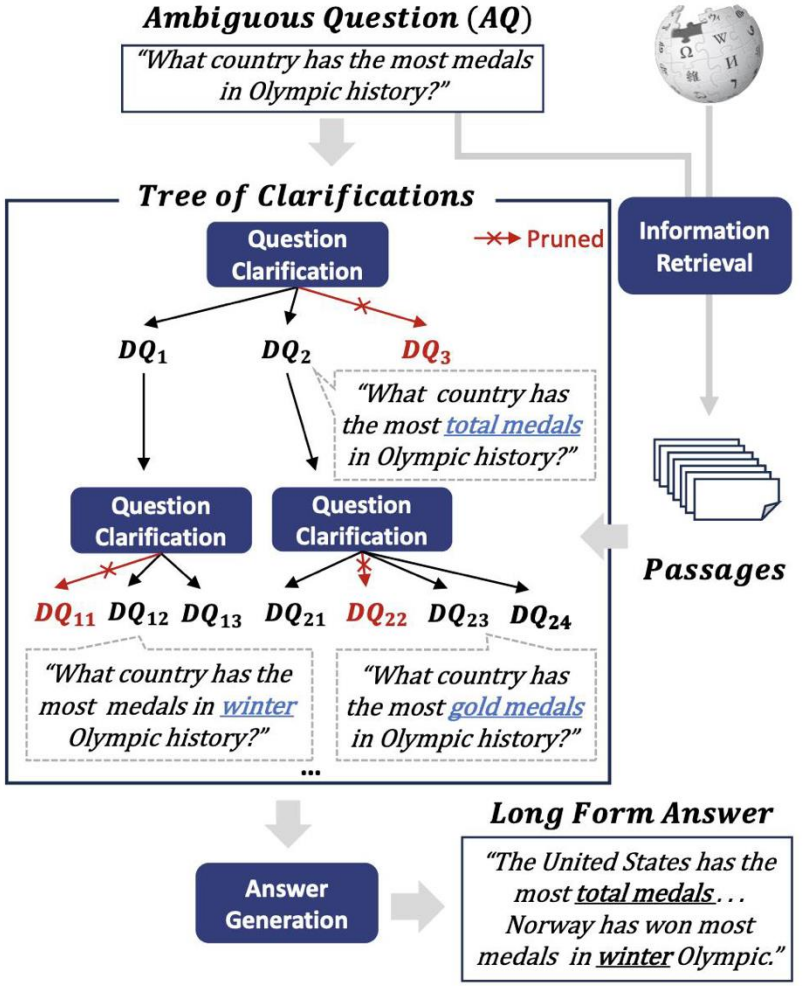
Las preguntas y las respuestas no siempre poseen alta similitud semántica; ajustar la consulta puede producir mejores resultados de recuperación.

Re-escritura de consulta



Rewrite-Retrieve-Read [Ma et al., 2023]

Clarificación de consulta



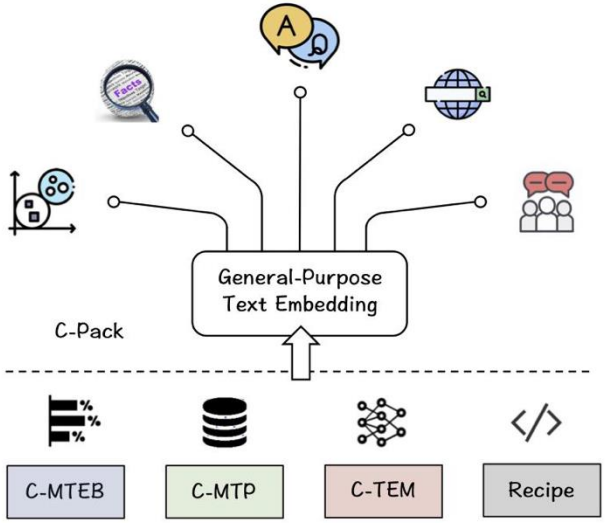
Tree of Clarifications（TOC）[Kim et al.,2023]

Técnicas para mejorar RAG — Optimización de embeddings

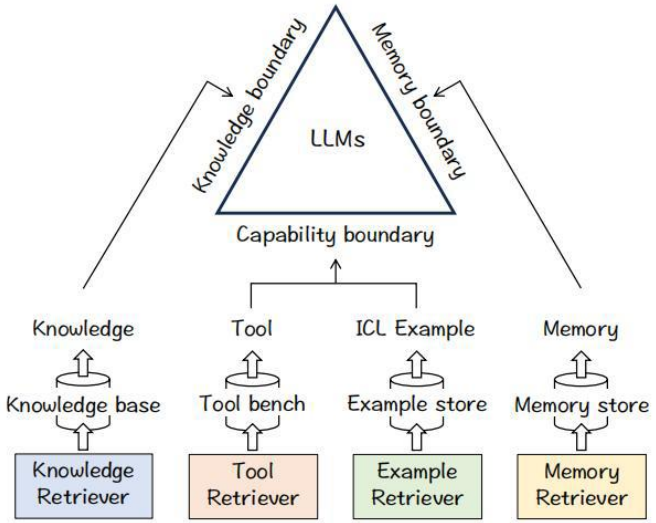
Seleccionar un proveedor de embeddings más adecuado

cohere · M3E · VOYAGE AI · OpenAI · llmrails · BAAI 智源研究院

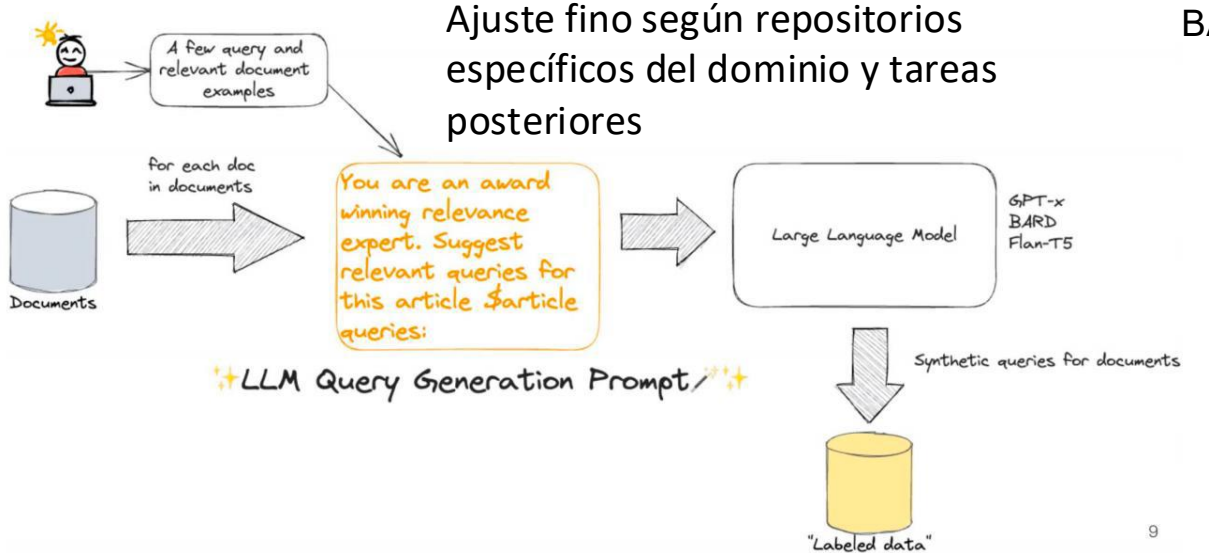
Ajuste fino del modelo de embeddings



BAAI-General-Embedding (BGE)

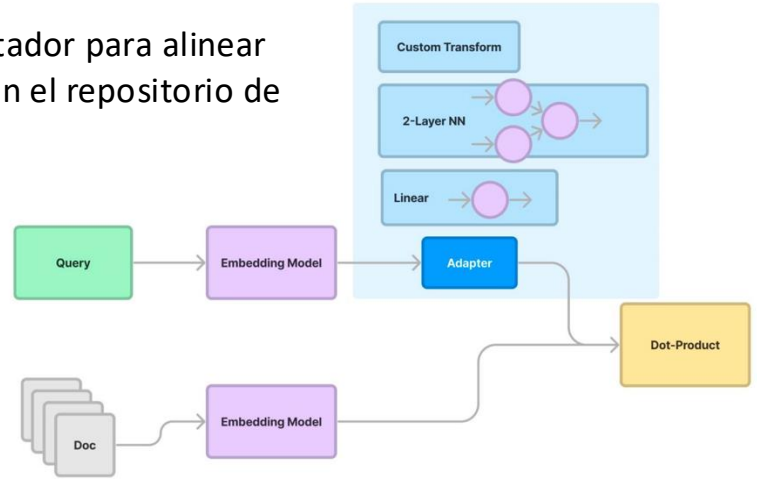


LLM-Embedder(BGE2) [Aksitov et al.,2023]



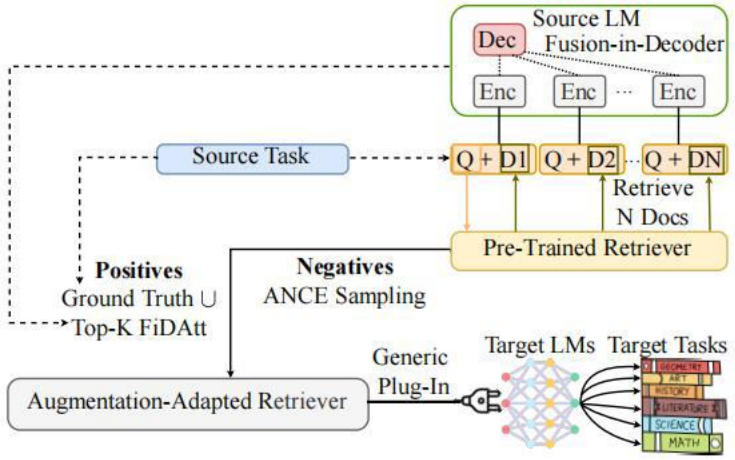
Ajuste fino según repositorios específicos del dominio y tareas posteriores

Ajuste fino del módulo adaptador para alinear el modelo de embeddings con el repositorio de recuperación



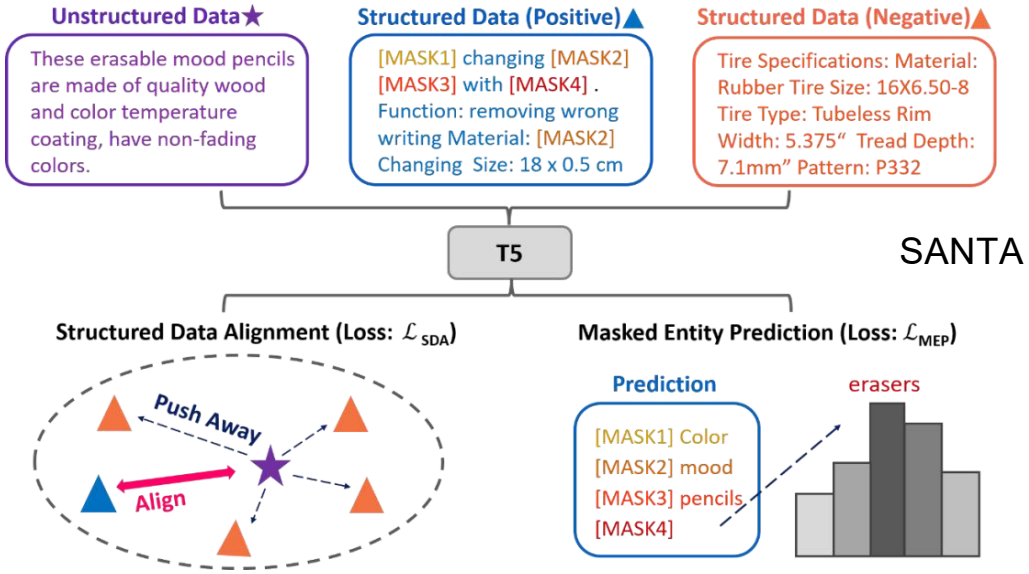
Técnicas para mejorar RAG — Hibrido (RAG + Fine-tuning)

Ajuste fino del Recuperador



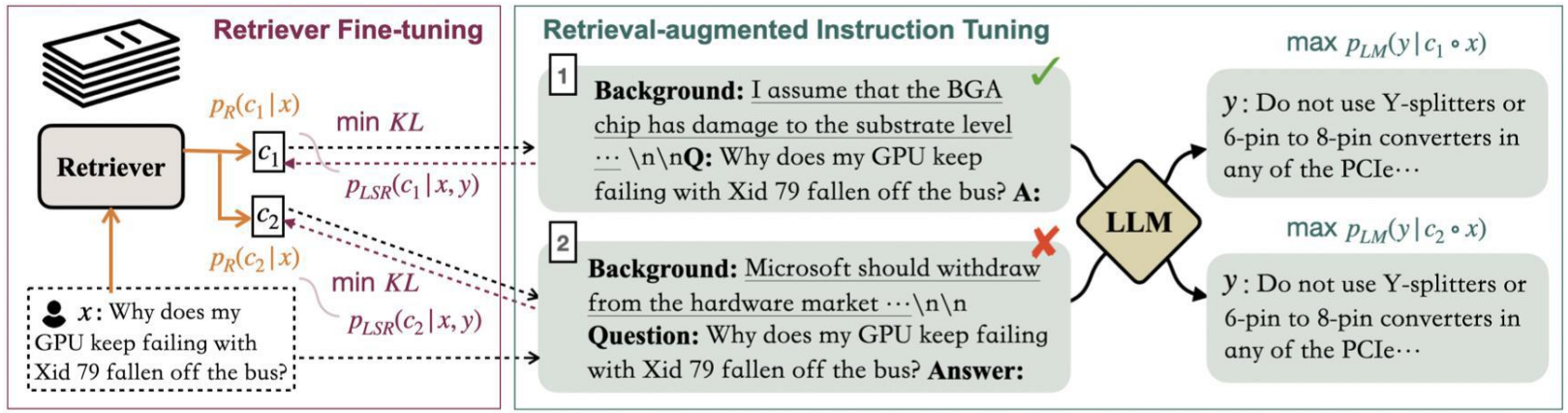
AAR [Yu et al., 2023]

Ajuste fino del generador



SANTA [Li et al., 2023]

Ajuste fino colaborativo / conjunto



- R-FT

Minimizar la divergencia KL entre la distribución del recuperador y las preferencias del LLM

- LM-FT

Maximizar la verosimilitud de la respuesta correcta dada instrucciones aumentadas con recuperación

Evaluación de la efectividad del RAG

Métodos de evaluación

- **Evaluación Independiente**
  - **Recuperador (Retriever):** Evaluar la calidad de los bloques de texto recuperados por la consulta
    Métricas: MRP, Tasa de Aciertos (Hit Rate), NDCG
  - **Generación / Síntesis:** Calidad del contexto enriquecido con documentos recuperados: Evaluación
    Métrica: Relevancia del Contexto (Context Relevance)
- **Evaluación end-to-end:** Evaluar el contenido finalmente generado por el modelo.
- **Según el contenido generado:**
  - Con etiquetas: EM (Exact Match), Accuracy (Precisión)
  - Sin etiquetas: Fidelidad, Relevancia, Inocuidad
- **Según el método de evaluación:**
  - Evaluación humana
  - Evaluación automática (LLM como juez)

Capacidades

- **Robustez al ruido**

¿Puede el modelo extraer información útil de documentos ruidosos?

- **Rechazo de negativos**

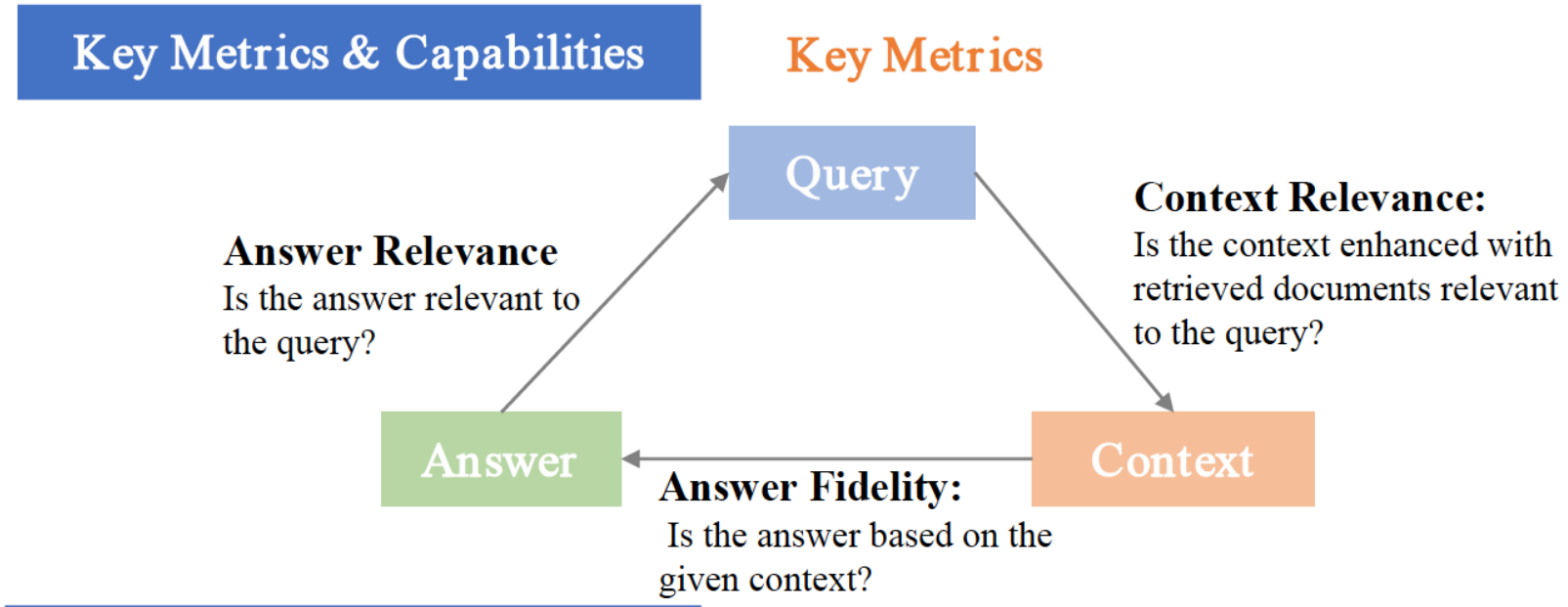
Cuando el conocimiento requerido no existe en los documentos recuperados, la respuesta debería ser rechazada.

- **Integración de información**

¿Puede el modelo responder preguntas complejas que requieren integrar información de múltiples documentos?

- **Robustez contrafactual**

¿Puede el modelo reconocer el riesgo de errores factuales conocidos en los documentos recuperados?



Métricas de evaluación del recuperador

Precision@k

- Mide la proporción de chunks correctos dentro del top-k recuperado.

$$P@k = \frac{\#\ de\ chunks\ relevantes\ en\ top-k}{k}$$

- Interpretación:
  - alto P@k = poco ruido en el contexto
  - bajo P@k = muchos chunks irrelevantes

Precision@k

Ejemplo:

- C = { c12, c13 }
- top-3 = [c7, c13, c22]
- Chunks correctos en top-3
  - c13 ✅
  - c7, c22 ❌

$$P@3 = \frac{1}{3} = 0.33$$

Interpretación: 1 de los 3 chunks recuperados era relevante

Recall@k

- Mide qué fracción de todos los chunks de referencia fue recuperada en el top-k.

$$R@k = \frac{\#\ de\ chunks\ de\ referencia\ recuperados\ en\ top-k}{\#\ de\ chunks\ de\ referencia\ totales}$$

- Interpretación:
  - alto R@k = el sistema no deja afuera evidencia importante
  - bajo R@k = el sistema deja afuera evidencia importante

Recall@k

Ejemplo:

- C = { c12, c13 }
- top-5 = [c7, c13, c22, c12, c3]
- Chunks correctos en top-5
  - c13, c12 ✅

$$R@5 = \frac{2}{2} = 1$$

Interpretación: en top-5 recuperó todos chunks de referencia

MRR (Mean Reciprocal Rank)

- Mide qué tan pronto aparece el primer chunk relevante correcto.

$$RR_i = \frac{1}{posición\ del\ primer\ relevante}$$

$$MRR = \frac{1}{N}\sum_{i=1}^{N} RR_i$$

- donde:
  - RR_i representa el Reciprocal Rank de una simulación i.
- Interpretación:
  - más alto = primer chunk correcto aparece más arriba

MRR (Mean Reciprocal Rank)

Ejemplo (N = 2):

- C_1 = { c12, c13 }
- X_1 = [c7, c13, c22, c12, c3]
- Primer relevante:
- c13 en posición 2

$$RR_1 = \frac{1}{2} = 0.5$$

- C_2 = { c12, c13 }
- X_2 = [c13, c7, c22, c3, c12]
- Primer relevante:
- c13 en posición 1

$$RR_2 = \frac{1}{1} = 1$$

$$MRR = \frac{1}{2} \times (0.5 + 1) = 0.75$$

MAP (Mean Average Precision)

- Mide la calidad global del ranking cuando puede haber varios chunks correctos.

$$AP_i = \frac{\sum_{k=1}^{K} P@k \cdot rel(k)}{|C|}$$

- donde:
  - AP_i representa el Average Precision de una simulación i.
- donde:
  - $rel(k) = 1$, si el chunk en posición k es relevante.
  - $|C|$ = cantidad de chunks de referencia

$$MAP = \frac{1}{N} \sum_{i=1}^{N} AP_i$$

MAP (Mean Average Precision)

Ejemplo (N = 2):

- C_1 = { c12, c13 }
- X_1 = [c7, **c13**, c22, **c12**, c3]
- Relevantes en posiciones 2 y 4.

$$P@2 = \frac{1}{2}, \qquad P@4 = \frac{2}{4} = \frac{1}{2}$$

$$AP_1 = \frac{P@2 + P@4}{2} = \frac{0.5 + 0.5}{2} = 0.5$$

- C_2 = { c12, c13 }
- X_2 = [**c13**, c7, c22, c3, **c12**]
- Relevantes en posiciones 1 y 5.

$$P@1 = 1, \qquad P@5 = \frac{2}{5}$$

$$AP_2 = \frac{P@1 + P@5}{2} = \frac{1 + 0.4}{2} = 0.7$$

$$MAP = \frac{1}{2} \times (0.5 + 0.45) = 0.475$$

nDCG@k (Normalized Discounted Cumulative Gain)

- Mide la calidad del ranking dando más peso a los aciertos que aparecen arriba.

$$DCG@k = \sum_{i=1}^{k} \frac{rel_i}{log_2(i+1)}$$

$$nDCG@k = \frac{DCG@k}{IDCG@k}$$

- donde:
  - $rel(k) = 1$, si el chunk en posición i es relevante.
  - $IDCG@k$ = DCG ideal, con el mejor orden posible

nDCG@k

Ejemplo:

- C = { c12, c13 }
- X = [c7, **c13**, c22, **c12**, c3]

$$DCG@5 = \frac{0}{log_2(2)} + \frac{1}{log_2(3)} + \frac{0}{log_2(4)} + \frac{1}{log_2(5)} + \frac{0}{log_2(6)} \approx 1.062$$

$$IDCG@5 = \frac{1}{log_2(2)} + \frac{1}{log_2(3)} + \frac{0}{log_2(4)} + \frac{0}{log_2(5)} + \frac{0}{log_2(6)} \approx 1.631$$

$$nDCG@5 = \frac{1.062}{1.631} \approx 0.651$$

Métricas de evaluación del generador

Token F1

- Mide el solapamiento de tokens entre la respuesta generada y la de referencia.

$$Precision = \frac{\#tokens\ compartidos}{\#tokens\ predichos},$$

$$Recall = \frac{\#tokens\ compartidos}{\#tokens\ de\ referencia}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

Token F1

Ejemplo:

- Respuesta de referencia:
  - “La autoridad competente es el juez”
- Predicción:
  - “La autoridad competente es el defensor”
- Tokens compartidos:
  - la, autoridad, competente, es, el

$$Precision = \frac{5}{6}, \qquad Recall = \frac{5}{6}$$

$$F1 = \frac{2 \cdot \frac{5}{6} \cdot \frac{5}{6}}{\frac{5}{6} + \frac{5}{6}} = \frac{5}{6} \approx 0.83$$

Observación: No detecta que juez y defensor son equivalentes.

ROUGE-L

- Mide la similitud basada en la Subsecuencia Común Más Larga (LCS).
- La LCS es la subsecuencia más larga compartida por 2 textos, respetando el orden.

$$Precision_{LCS} = \frac{LCS}{|Y|}, \qquad Recall_{LCS} = \frac{LCS}{|R^*|}$$

$$F_{LCS} = \frac{2 \cdot Precision_{LCS} \cdot Recall_{LCS}}{Precision_{LCS} + Recall_{LCS}}$$

ROUGE-L

Ejemplo:

- Respuesta de referencia:
  - “La autoridad competente es el juez”
- Predicción:
  - “La autoridad competente es el defensor”
- LCS:
  - “La autoridad competente es el”

$$P_{LCS} = \frac{5}{6}, \qquad R_{LCS} = \frac{5}{6}$$

$$F_{LCS} = \frac{2 \cdot \frac{5}{6} \cdot \frac{5}{6}}{\frac{5}{6} + \frac{5}{6}} = \frac{5}{6} \approx 0.83$$

Observación: No detecta que juez y defensor son equivalentes.

BERTScore

- Mide la similitud semántica entre la respuesta generada y de referencia usando embeddings contextuales.

- Pasos generales para calcular BERTScore:
  1. Tokenizar ambos textos.
  2. Obtener embedding por token con un enconder fijo.
  3. Calcular similaridad de coseno entre tokens.
  4. Para cada token, quedarse con su mejor match
  5. Promediar para formar Precision, Recall y F1.

BERTScore - Ejemplo

- Supuesto de simplificación:
  - Para hacer visible el cálculo, se muestran sólo los tokens con mayor carga semántica. En la implementación real se usan todos los tokens del tokenizer.
- Respuesta de referencia:
  - “El menor tiene derecho a ser oído”
- Predicción:
  - “El niño tiene derecho a ser escuchado en el proceso”
- Tokens considerados:
  - Referencia: [menor, derecho, oído]
  - Predicción: [niño, derecho, escuchado, proceso]
- Matriz de similitud:

$$S = \begin{bmatrix} 0.89 & 0.10 & 0.12 \\ 0.08 & 1.00 & 0.06 \\ 0.11 & 0.09 & 0.86 \\ 0.18 & 0.22 & 0.15 \end{bmatrix}$$

Filas: tokens de la predicción
Columnas: tokens de la referencia

BERTScore - Ejemplo

- Selección para Precisión
  - niño → max(0.89, 0.10, 0.12) = 0.89
  - derecho → max(0.08, 1.00, 0.06) = 1.00
  - escuchado → max(0.11, 0.09, 0.86) = 0.86
  - proceso → max(0.18, 0.22, 0.15) = 0.22

$$P_{BERT} = \frac{0.89 + 1.00 + 0.86 + 0.22}{4} = \frac{2.97}{4} = 0.7425$$

BERTScore - Ejemplo

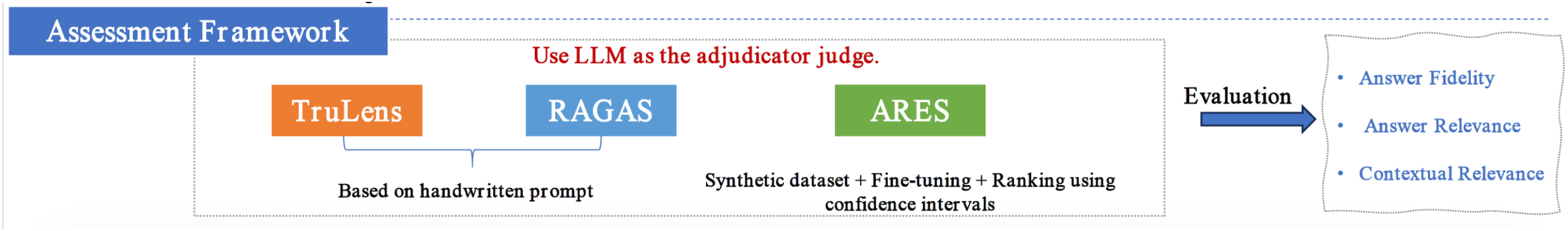
- Selección para Recall
  - menor → max(0.89, 0.08, 0.11, 0.18) = 0.89
  - derecho → max(0.10, 1.00, 0.09, 0.22) = 1.00
  - oído → max(0.12, 0.06, 0.86, 0.15) = 0.86

$$R_{BERT} = \frac{0.89 + 1.00 + 0.86}{3} = \frac{2.75}{3} \approx 0.917$$

$$F1 = \frac{2 \cdot 0.7425 \cdot 0.917}{0.7425 + 0.917} \approx 0.821$$

- **Interpretación:** la predicción cubre correctamente la mayor parte del significado de la referencia, pero agrega información extra (“proceso”) poco alineada con ella. Por eso el Recall es alto, la Precision baja un poco y el F1 queda en un valor intermedio-alto.

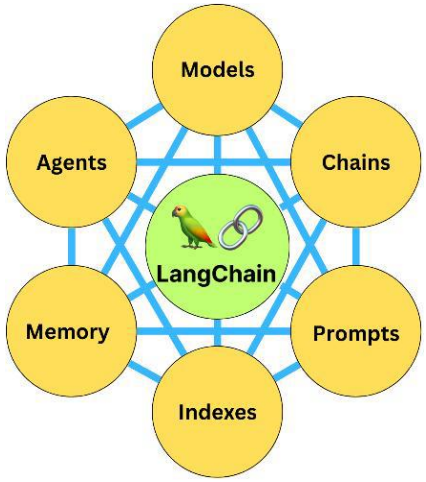
Marcos de evaluación (frameworks)



Stack tecnológico existente para RAG

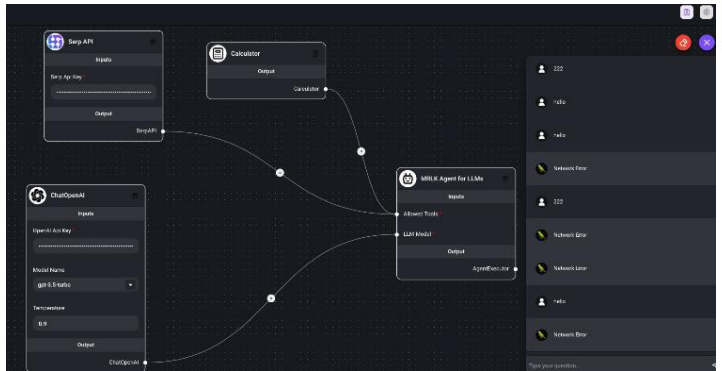
Nombre	Pros	Contras
LangChain	Modular, con funcionalidades completas	Comportamiento inconsistente, la API oculta detalles, alta complejidad y baja flexibilidad
LlamaIndex	Enfocado en RAG	Requiere uso combinado con otras herramientas, baja personalización
FlowiseAI	Fácil de comenzar, flujos de trabajo visuales	No soporta escenarios complejos
AutoGen	Se adapta a escenarios multi-agente	Baja eficiencia, requiere múltiples rondas de diálogo

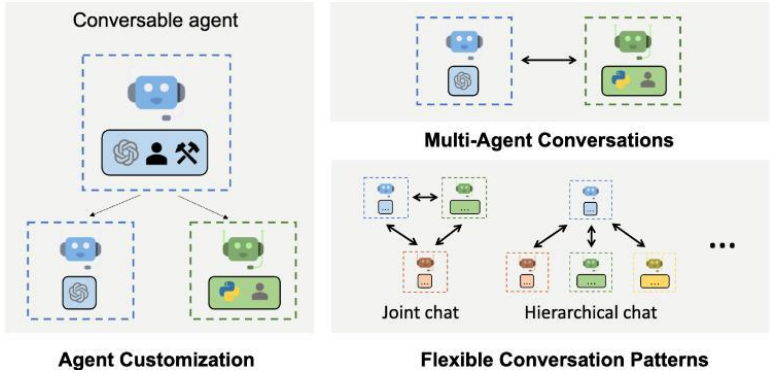
LangChain



LlamaIndex

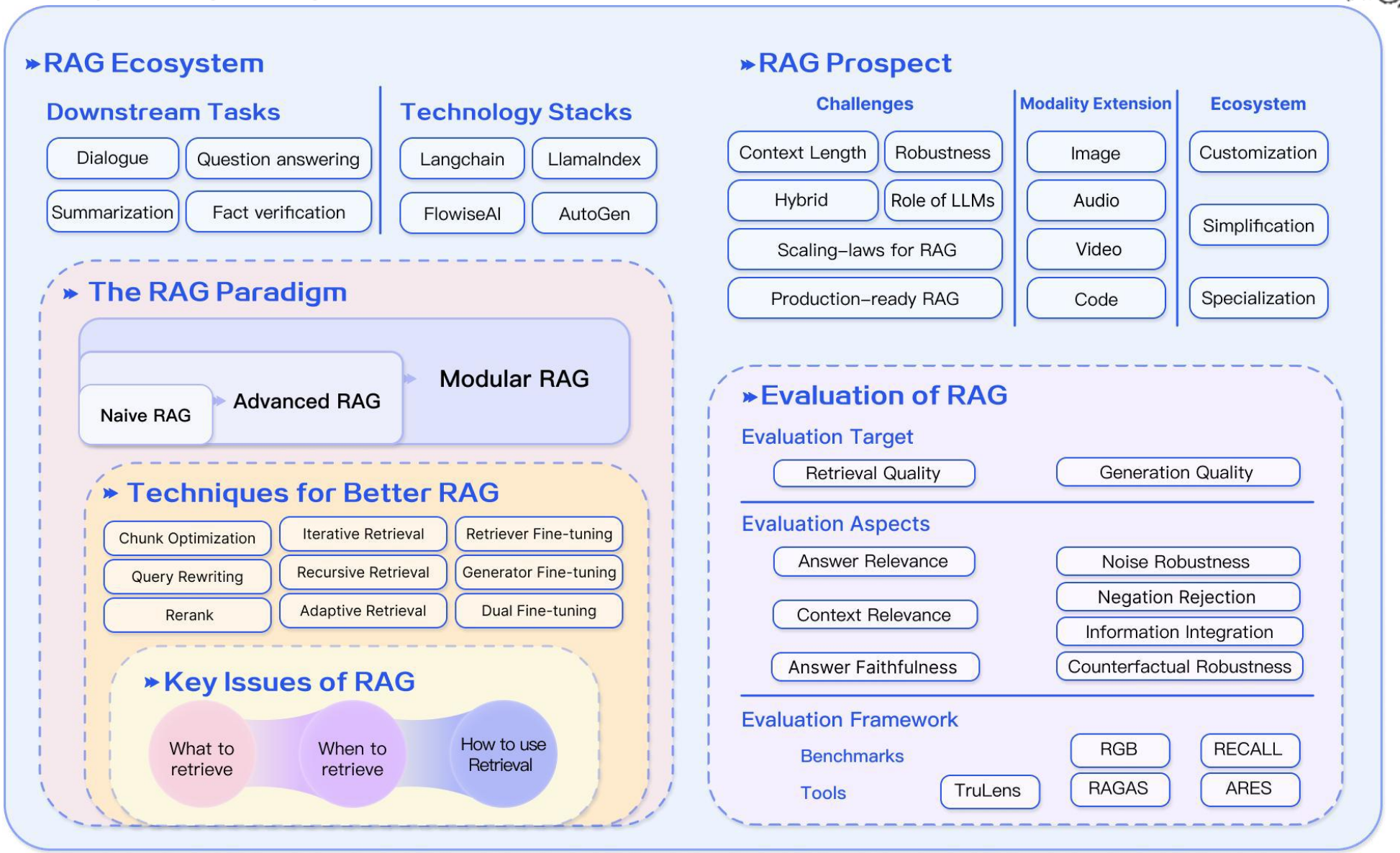
FlowiseAI





AutoGen

RAG Framework



Referencias

Paper : https://arxiv.org/abs/2312.10997
GitHub: https://github.com/Tongji-KGLLM/RAG-Survey

Repositorios relacionados:

- https://github.com/Danielskry/Awesome-RAG
- https://github.com/NirDiamant/rag_techniques