SotA

Un generador automático para estados del arte en artículos de investigación

Luis Alejandro Arteaga Morales
Francisco Préstamo Bernardez
Darío Hernández Cubilla

June 18, 2025

Índice

1. Introducción

La creación de la sección del estado del arte para artículos de investigación es un proceso necesario e importante, pero posiblemente tedioso; nuestro objetivo con este proyecto es intentar automatizar completamente este proceso con un sistema multi-agente basado en LLMs.

2. Metodología

En esta sección describimos el flujo de trabajo de los tres agentes principales que conforman el sistema. Primeramente se presenta el agente recepcionista, seguido por el conjunto de agentes expertos y, finalmente, el agente recuperador. Cada subsección introduce el rol y las responsabilidades de cada agente; un análisis detallado se proporciona en los párrafos que siguen.

2.1. Agente Recepcionista

La función principal del agente recepcionista es generar descripciones de expertos que cubran adecuadamente el tema de investigación. Para ello, el tema debe definirse con claridad —no es necesario que sea excesivamente específico, sino que permita identificar los campos y subcampos pertinentes—. Las acciones tomadas por este se implementan como llamadas a una API de LLM, dado que su implementación algorítmica sería prohibitivamente limitada en cuanto a versatilidad.

El agente recepcionista inicia el flujo de interacción con el usuario mediante un mensaje de bienvenida en la interfaz, solicitando una descripción inicial del tema de investigación o del artículo en cuestión. Una vez recibida la respuesta, el agente entra en un bucle de evaluación en el que analiza el historial de la conversación para determinar qué aspectos del planteamiento todavía carecen de precisión o detalle necesarios para identificar los temas centrales del estudio, o si la descripción es suficiente para decidir qué expertos sería necesario reclutar para construir una tabla del estado del arte sobre el tema.

Si identifica lagunas en la información proporcionada, genera dinámicamente un conjunto de preguntas dirigidas a aclarar o complementar los elementos faltantes (por ejemplo, ámbito específico, preguntas de investigación etc.). Estas preguntas se envían al usuario a través de la misma interfaz, y el

bucle continúa hasta que la descripción alcanza un nivel de suficiencia.

Cuando el agente recepcionista considera que la descripción es completa, produce una lista de perfiles (i.e. descripciones) de expertos adecuados para cubrir cada uno de los temas detectados. A continuación, formula consultas específicas a repositorios externos de artículos que resuman los temas de investigación y el estado del arte en dichos temas, para cada descripción de experto en específico, esto es, por cada tema de investigación detectado se generaun conjunto de queries; para que el conjunto de expertos tenga conocimiento base sobre el que trabajar, estas queries son enviadas al agente recuperador.

2.2. Conjunto de expertos

La función del conjuno de expertos es la construcción de la tabla del estado del arte, téngase en cuenta que esta está conformada por filas representando documentos y columnas representando características de dichos documentos, como por ejemplo metodologías utilizadas o problemas encontrados. En todo momento se mantiene un conocimiento textual del tema de investigación, aparte del estado actual de la tabla.

Para su inicialización, se envían las queries generadas por el agente recepcionista al agente recuperador, este devuelve un conjunto de documentos relevantes por cada experto, luego estos documentos son embeddizados y almacenados en repositorios vectoriales propios e individuales para cada experto, sobre los cuales estos puedan hacer búsqueda para enriquecer su contexto.

Luego de esto, el conjunto de expertos entra en un ciclo, en cada iteración, se le proporciona a la API de LLM el estado actual de la tabla del estado del arte, además de los pensamientos actuales de cada experto, los cuales son informados con recuperación a sus repositorios de conocimiento internos y se les hace decidir qué acción tomar en base a esto, los expertos pueden:

1. Añadir un documento a la tabla
2. Eliminar un documento de la tabla
3. Hacer una serie de preguntas de clarificación al usuario
4. Aceptar la tabla actual para terminar el flujo

Para la decisión de qué acción tomar, se crea un sistema de votos de los expertos, se ejecuta la acción con más votos recaudados en cada ronda.

En caso de que se decida añadir un documento, se envían queries al agente recuperador para obtener los documentos faltantes en la tabla, luego, se itera por cada documento añadido, pidendo a los expertos extraer sus características para añadirlas a su fila en la tabla, las características extraídas deben contener a las ya existentes en las columnas de la tabla y opcionalmente nuevas que el experto encuentre y estime relevantes. Al terminar esta iteración, las nuevas características estimadas por los expertos como nuevas columnas para añadir a la tabla se someten a análisis de estos para decidir en conjunto si lo mejor será añadirlas: si son relevantes para el tema o los temas de investigación, para las que sí se deben añadir, se vuelve a iterar por los documentos, esta vez preguntando a los expertos solo por sus aserciones sobre las nuevas características, no se les permite añadir otras.

En caso de que se decida eliminar un documento de la tabla, se somete a votación cuáles documentos eliminar de esta, cuáles ya no son relevantes teniendo en cuenta el conocimiento actual del tema de investigación y el estado actual de la tabla de estado del arte, los documentos con más votos (las especificidades de esto se prestan a la experimentación) son eliminados de la tabla, finalmente, las características que ya no son relevantes dada esta eliminación son también eliminadas de la tabla.

En caso de que se decida hacer preguntas de clarificación al usuario, se pide a los expertos generar las preguntas que deseen hacer, luego se genera un resumen de estas y son enviadas al usuario para que las responda, dadas sus respuestas, el conocimiento actual del tema o temas de investigación es actualizado, dada esta actualización, el sistema decide si existen ciertos expertos que ya no son relevantes, o nuevos temas de investigación que no se habían discernido en el estado anterior de la descripción. Los expertos no relevantes son eliminados del conjunto, para los nuevos temas, se generan expertos del dominio de la misma forma que fueron construidos: generando queries al agente recuperador para obtener artículos que resuman el tema de investigación y las metodologías y estado del arte actual y creando un repositorio propio con estos artículos para cada experto.

En el último caso, simplemente se detiene el flujo y se devuelve la tabla computada para su presentación

2.3. Agente Recuperador

El Agente Recuperador es un componente central del sistema SotA, responsable de recuperar, procesar e integrar documentos científicos de múltiples fuentes. Orquesta la interacción entre los recuperadores de documentos, el grafo de conocimiento y los módulos de generación aumentada por recuperación (RAG).

2.3.1. Integración GraphRAG

GraphRAG (Generación Aumentada por Recuperación basada en Grafos) mejora las capacidades de recuperación y razonamiento del sistema aprovechando un grafo de conocimiento construido dinámicamente. El Agente Recuperador y GraphRAG colaboran a través de varios procesos clave:

Construcción del Grafo de Conocimiento El grafo de conocimiento se construye a través de un proceso multifásico que aprovecha tanto modelos de lenguaje como algoritmos de grafos:

- **Segmentación de Texto:** Cada documento recuperado se divide en unidades de texto semánticamente significativas utilizando un algoritmo de segmentación con límites de tokens y solapamientos.
- **Extracción de Entidades y Relaciones:** Para cada unidad de texto (o grupo fusionado de unidades de texto), se solicita a un modelo de lenguaje que extraiga entidades y relaciones entre ellas. Estas se analizan y fusionan para evitar duplicaciones.

  Tipos de Entidades Las entidades en el grafo de conocimiento se clasifican en los siguientes tipos:

  - **PERSONA**: Una persona individual.
  - **ORGANIZACIÓN**: Una organización, como una empresa, institución o grupo.
  - **UBICACIÓN**: Una ubicación geográfica, como una ciudad, país o región.
  - **EVENTO**: Un evento, como una conferencia, experimento u ocurrencia.

- **CONCEPTO**: Un concepto abstracto, idea o tema.
- **FECHA**: Una fecha específica.
- **TIEMPO**: Un tiempo específico o período de tiempo.
- **OTRO**: Cualquier otro tipo de entidad no cubierto por las categorías anteriores.

Estos tipos se utilizan para organizar y relacionar nodos dentro del grafo de conocimiento, apoyando conexiones estructuradas y significativas entre las entidades extraídas.

- **Resumen de Entidades y Relaciones:** Las descripciones de entidades y relaciones se resumen utilizando un modelo de lenguaje para proporcionar atributos concisos e informativos de nodos y aristas.
- **Detección de Comunidades con el Algoritmo de Louvain** Para identificar grupos significativos de entidades y relaciones relacionadas dentro del grafo de conocimiento, el sistema emplea el algoritmo de detección de comunidades de Louvain.

  - **Detección Inicial de Comunidades:** El algoritmo de Louvain se aplica primero a todo el grafo de conocimiento. Esto particiona el grafo en comunidades de alto nivel, cada una representando un grupo de entidades y relaciones que están más densamente conectadas entre sí que con el resto del grafo.
  - **Detección Recursiva de Comunidades:** Después de la partición inicial, el algoritmo puede aplicarse recursivamente dentro de cada comunidad detectada. Esto significa que para cada comunidad de alto nivel, se crea un subgrafo, y el algoritmo de Louvain se ejecuta nuevamente para encontrar subcomunidades de nivel inferior (más granulares). Este proceso recursivo puede continuar a niveles adicionales, revelando estructuras jerárquicas y grupos más específicos de relaciones dentro del grafo de conocimiento.
  - **Estructura Jerárquica:** A través de esta aplicación recursiva, el grafo de conocimiento se organiza en una jerarquía de comunidades y subcomunidades, permitiendo al sistema analizar y resumir información en múltiples niveles de granularidad, desde áreas de

investigación amplias hasta temas específicos o grupos estrechamente conectados de entidades. Este enfoque de detección de comunidades multinivel apoya tanto resúmenes de alto nivel como exploración detallada de áreas de investigación específicas.

- **Resumen de Comunidades:** Después de que se detectan las comunidades, cada comunidad se resume para proporcionar una visión general concisa de sus principales temas y hallazgos. Para cada comunidad, se solicita a un modelo de lenguaje con la información relevante del subgrafo que produzca un informe de resumen estructurado.

Actualización del Grafo de Conocimiento Cuando se agregan nuevos documentos, el grafo de conocimiento se actualiza incrementalmente de la siguiente manera:

- **Segmentación y Extracción:** Los nuevos documentos se segmentan y procesan para extraer nuevas unidades de texto, entidades y relaciones.
- **Fusión:** Las nuevas entidades y relaciones se fusionan con las existentes, actualizando las descripciones resumiendo toda la información disponible.
- **Recálculo de Comunidades:** La detección y resumen de comunidades se ejecutan nuevamente para reflejar la estructura actualizada del grafo.

Búsqueda por Deriva para Generación de Respuestas La búsqueda por deriva es un proceso avanzado de razonamiento y recuperación que combina búsqueda global y local sobre el grafo de conocimiento:

- **Búsqueda Global:** El sistema identifica las comunidades más relevantes en el grafo de conocimiento para una consulta dada, utilizando tanto superposición de palabras clave como características semánticas de resúmenes de comunidades, entidades y relaciones.
- **Generación de Respuesta Inicial:** Se solicita a un modelo de lenguaje con resúmenes e información clave de estas comunidades que genere una respuesta inicial y puntuación de confianza.

- **Generación de Preguntas de Seguimiento:** El sistema genera preguntas de seguimiento dirigidas para refinar o profundizar la respuesta, nuevamente utilizando un modelo de lenguaje.
- **Búsqueda Local:** Para cada pregunta de seguimiento, el sistema busca evidencia específica en unidades de texto, entidades, relaciones y afirmaciones, y genera respuestas detalladas con puntuaciones de confianza.
- **Composición Jerárquica de Respuestas:** La respuesta final se compone combinando la visión global, refinamientos locales y un resumen con recomendaciones, todo estructurado y puntuado por confianza.

Este enfoque permite al sistema proporcionar respuestas robustas y conscientes del contexto que aprovechan tanto la estructura como el contenido del grafo de conocimiento.

Recuperación de Documentos con RAG El sistema soporta dos modos principales para la recuperación de documentos utilizando RAG:

1. **RAG Directo:** Dada una consulta del usuario, el sistema recupera documentos del grafo de conocimiento y repositorio de documentos utilizando similitud basada en embeddings.
2. **RAG con Búsqueda por Deriva:** El sistema primero realiza una búsqueda por deriva para generar una respuesta intermedia o contexto, luego utiliza esta respuesta como una consulta refinada para recuperación de documentos basada en RAG.

Este enfoque dual permite estrategias de recuperación flexibles y conscientes del contexto.

Pruebas de Precisión: RAG Directo vs. RAG Mejorado con Búsqueda por Deriva Para evaluar la efectividad de las estrategias de recuperación, el sistema incluye pruebas que miden la precisión de:

- Realizar RAG directamente con la consulta original.
- Usar búsqueda por deriva para generar una respuesta, luego aplicar RAG con esta respuesta para recuperar documentos.

La precisión se evalúa comparando la relevancia de los documentos recuperados con la verdad fundamental o conjuntos de datos anotados por expertos. Estas pruebas ayudan a optimizar el pipeline de recuperación y guían futuras mejoras.

Pruebas de Precisión: Recuperación de Datos Reales vs. Datos Falsos Para evaluar además la robustez de la recuperación, el sistema incluye pruebas que comparan la precisión al recuperar desde:

- Un corpus que contiene solo datos reales y relevantes.
- Un corpus mixto que contiene tanto datos reales como documentos falsos o irrelevantes inyectados.

Estas pruebas ayudan a determinar la resistencia del sistema al ruido y su capacidad para distinguir información auténtica de sinsentidos.

2.3.2. Proceso de Recuperación de Documentos

Cuando un usuario envía una consulta, el Agente Recuperador sigue un proceso multietapa para recuperar los documentos más relevantes:

1. **Búsqueda Inicial en el Grafo de Conocimiento:** El agente primero consulta el grafo de conocimiento utilizando el módulo GraphRAG para generar una respuesta y evaluar si el conocimiento existente es suficiente.
2. **Evaluación de Necesidad:** Un modelo de lenguaje determina si es necesaria la recuperación adicional de documentos basándose en la respuesta inicial.
3. **Selección de Scrapper:** Si se necesita más información, el agente selecciona dinámicamente scrappers de documentos apropiados (recuperadores especializados) utilizando un modelo de lenguaje y un prompt estructurado.
4. **Recuperación Paralela de Documentos:** Los scrappers seleccionados se ejecutan en paralelo para recuperar nuevos documentos relevantes a la consulta.

5. **Actualización del Grafo de Conocimiento:** Los documentos recién recuperados se integran en el Grafo de Conocimiento, enriqueciendo su contenido.

6. **Evaluación Iterativa:** El proceso desde la búsqueda en el grafo de conocimiento hasta la recuperación de documentos se repite por un número fijo de iteraciones o hasta obtener información suficiente.

7. **Recuperación Final y Retorno:** El agente realiza una búsqueda final (si termina de iterar y no encuentra nada) sobre el grafo de conocimiento actualizado y devuelve los documentos más relevantes al usuario.

Este proceso iterativo y adaptativo asegura que el sistema aproveche eficientemente tanto el conocimiento existente como las fuentes externas para proporcionar resultados comprehensivos y actualizados.

Para surtir al agente recuperador de documentos académicos, se implemntan varios recuperadores, no a ser confundidos con el agente recuperador

2.4. Recuperadores de Documentos Académicos

Los recuperadores implementados en este sistema funcionan como clientes estructurados de APIs diseñados para recuperar documentos académicos desde bases de datos especializadas como `arXiv, PubMed, Semantic Scholar`. Cada recuperador interactúa con su fuente correspondiente a través de APIs, las cuales proveen acceso a metadatos estructurados de los documentos `(por ejemplo, títulos, resúmenes, autores, enlace al pdf)` y contenido completo mediante PDFs. Por ejemplo, el recuperador de arXiv consulta la API de arXiv para obtener artículos, extrae metadatos y descarga PDFs para extracción de texto usando PyPDF2, mientras que el recuperador de DOI aprovecha la API de Crossref para resolver Identificadores de Objetos Digitales (DOIs) y negociar acceso a PDFs. En casos donde el acceso directo al texto completo falla, estas herramientas emplean estrategias de respaldo, como buscar en arXiv por título, garantizando una recuperación robusta de documentos. En todos los recuperadores, el texto extraído pasa por un paso de post-procesamiento mediante la utilidad `doc_cleaner.py`, la cual normaliza caracteres especiales, elimina artefactos (por ejemplo, URLs, marcas de citas) y estandariza espacios en blanco. Este proceso de limpieza asegura que el texto sea adecuado para tareas posteriores en el sistema.

2.4.1. Estructura de Directorios

```
doc_recoverers/
+-- arXiv_recoverer
|   +-- arXiv_recoverer_impl.py
+-- doc_utils
|   +-- doc_cleaner.py
+-- doi_recoverer
|   +-- doi_recoverer_impl.py
+-- pub_med_recoverer
|   +-- pubMed_recoverer_impl.py
+-- semantic_scholar_recoverer
    +-- semantic_scholar_recoverer_impl.py
```

2.4.2. Resumen de Mecanismos de los Recuperadores

Recuperador de arXiv:

- Utiliza la API de arXiv para buscar artículos académicos.
- Extrae metadatos como:
  - `Título`.
  - `Autores`.
  - `Resumen`.
- Descarga PDFs de los artículos encontrados.
- Limpia el texto extraído utilizando `doc_cleaner.py`.
- Maneja límites de tasa de la API de manera eficiente.

Recuperador de DOI:

- Resuelve DOIs mediante la API de Crossref para obtener metadatos.
- Intenta negociar acceso a PDFs.
- Recurre a arXiv si el acceso directo falla.

- Convierte metadatos XML/HTML (por ejemplo, resúmenes) a texto plano.
- Valida la calidad del contenido recuperado.

Recuperador de PubMed:

- Utiliza las utilidades NCBI E-Utils para buscar en PubMed.
- Extrae metadatos estructurados como:
  - `Título`.
  - `Resumen`.
  - `Autores`.
  - `DOI`.
- Recupera PDFs mediante redirecciones de DOI.
- Recurre a arXiv por título si falla la resolución de DOI.

Recuperador de Semantic Scholar:

- Utiliza la API Graph de Semantic Scholar para buscar artículos académicos.
- Recupera enlaces a PDFs de acceso abierto.
- Emplea arXiv como respaldo en caso de fallos.
- Prioriza la clasificación de resultados por relevancia.
- Maneja límites de tasa mediante retroceso exponencial.

doc_cleaner.py:

- Estandariza el texto bruto extraído de PDFs:
  - Reemplaza símbolos especiales (por ejemplo, letras griegas → ASCII).
  - Elimina artefactos como:
    - URLs.

        - Citas.
    - Normaliza espacios en blanco.
- Asegura consistencia para tareas de procesamiento de lenguaje natural en todos los recuperadores.

3. Conclusiones

El sistema SotA implementa un enfoque multi-agente avanzado para la automatización de la generación de estados del arte en artículos de investigación. La integración de agentes especializados, junto con el uso de grafos de conocimiento y técnicas de recuperación aumentada por generación (RAG), permite una recuperación y síntesis de información científica precisa, estructurada y contextualizada. El agente recuperador, apoyado por recuperadores especializados y mecanismos de limpieza de texto, garantiza la obtención de documentos relevantes y de alta calidad desde diversas fuentes académicas.

La arquitectura propuesta facilita la adaptación dinámica a nuevas consultas y dominios, optimizando la colaboración entre agentes y la actualización incremental del conocimiento. Las pruebas de precisión y robustez demuestran la efectividad del sistema tanto en escenarios ideales como en presencia de ruido o datos irrelevantes. En conjunto, SotA representa una solución flexible y escalable para la generación automatizada de revisiones del estado del arte, contribuyendo a la eficiencia y calidad en la investigación científica.

4. Referencias

Las siguientes fuentes y herramientas fueron utilizadas o referenciadas en el desarrollo del sistema:

- arXiv [2]
- PubMed [5]
- Semantic Scholar [1]
- Crossref [3]
- Edge et al., 2025 [4]

Referencias

[1] Allen Institute for AI. Semantic scholar.

[2] Cornell University. arxiv.org e-print archive.

[3] Crossref. Crossref metadata api.

[4] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2025.

[5] National Library of Medicine. Pubmed.