Sistema inteligente para la gestión de conocimiento científico en grupos de investigación

Óscar Gil-Bernad, Sergio Jimenez-Frago, Hala Neji[0000−0003−3595−005X], and Javier Lacasta[0000−0002−6557−2494]

Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza, Aragón, España, EU,
{870919,sjimenezfr,hala.neji,jlacasta}@unizar.es

Resumen Este trabajo presenta una herramienta inteligente para la gestión del conocimiento científico. La propuesta integra la extracción de contenido de documentos PDF, la identificación de temas, líneas de investigación y la generación automática de informes usando modelos de lenguaje abiertos. El sistema ha sido aplicado en el Instituto de Investigación en Ingeniería de Aragón (I3A), permitiendo la generación automática de informes de la actividad investigadora de sus miembros para facilitar el análisis de la producción científica.

Keywords: Procesamiento de lenguaje · Modelos de lenguaje · Generación de conocimiento.

1. Introducción

La actividad científica genera grandes cantidades de información de investigadores de universidades, centros de investigación y grupos científicos, en forma de artículos, informes, comunicaciones y otros tipos de documentos. Conforme aumenta dicha producción, también crece la dificultad para analizar la evolución de sus líneas de trabajo, identificar tendencias, sintetizar contribuciones o elaborar informes que describan la actividad en un determinado periodo temporal.

Tradicionalmente, estas tareas han requerido un importante esfuerzo por parte de investigadores, responsables de grupos y gestores científicos. La revisión individual de publicaciones, la extracción de información relevante y la elaboración de resúmenes de la trayectoria investigadora consumen tiempo y dificultan tener visión global y actualizada del trabajo desarrollado. Esta situación resulta especialmente grave en grupos de investigación consolidados cuya producción científica abarca cientos de publicaciones escritas a lo largo de muchos años.

Este trabajo presenta una herramienta inteligente para la gestión del conocimiento científico de grupos de investigación. La propuesta integra un proceso automatizado de recopilación y extracción de contenido en PDF a un almacén estructurado de conocimiento, como base de un sistema de generación automática de informes basado en modelos de lenguaje abiertos. Está compuesto de un

componente de extracción de contenido de PDFs, un componente de identificación de temas y líneas de trabajo a lo largo del tiempo y un componente de generación de informes automáticos usando modelos de lenguaje abiertos.

Las aportaciones se centran en estas tres áreas. Relacionado con la extracción de contenido, se muestra una comparativa de herramientas de extracción de texto a partir de documentos PDF y se compara su rendimiento con artículos científicos. En la identificación de temas, líneas de trabajos y generación de resúmenes se propone el uso de un modelo de lenguaje (LLM) abierto en una máquina sin hardware especializado. Estos modelos han demostrado la capacidad de sintetizar contenidos complejos y generar textos coherentes. Sin embargo, debido a que la cantidad de información a procesar excede el tamaño de entrada soportado por los modelos de estas características, el problema del resumen deja de depender únicamente del prompt y pasa a depender también de la forma de orquestar el flujo completo de trabajo. Ya no basta con pedir al modelo que resuma una entrada, sino que es necesario decidir cómo fragmentar el contenido, cómo procesar cada fragmento, cómo combinar resultados parciales y qué equilibrio alcanzar entre coherencia global, escalabilidad, coste computacional y fidelidad semántica. Para solventar este problema, el sistema incorpora una estrategia jerárquica de resumen que permite sintetizar conjuntos de publicaciones que superan ampliamente la ventana de contexto disponible, manteniendo la trazabilidad de la información hasta los artículos originales. Esta integración de múltiples técnicas de extracción y resumen de datos en este contexto, proporciona una forma novedosa de identificar de forma homogénea el contexto de investigación a diferentes niveles de granularidad (instituto, grupo, investigador).

La solución desarrollada ha sido aplicada en el contexto del Instituto de Investigación en Ingeniería de Aragón (I3A), donde se utiliza para analizar la producción científica de investigadores y grupos de investigación, facilitando tanto la exploración de líneas temáticas como la generación automática de informes narrativos y estadísticas de actividad científica.

El resto del artículo se organiza de la siguiente forma. La Sección 2 revisa el estado del arte relacionado con la extracción de información de documentos científicos y la generación automática de resúmenes. La Sección 3 describe la arquitectura del sistema propuesto, incluyendo los procesos de extracción de contenido, construcción del almacén de conocimiento y generación de informes. La Sección 4 presenta la aplicación desarrollada para los usuarios finales. Finalmente, la Sección 5 recoge las conclusiones y las principales líneas de trabajo futuro.

2. Estado del arte

La aproximación clásica al análisis del trabajo de los investigadores se ha centrado principalmente en el cálculo de métricas científicas. En este contexto plataformas como Scopus [1], Web of Science [3], Google Scholar [10], o ResearchGate [5] funcionan como agregadores del trabajo científico. Estos sistemas usan métricas como número de citas, h-index (número de artículos con ese número de

citas), índice i10 (artículos con más de 10 citas), o g-index (promedio de citas de las publicaciones más citadas). Estos sistemas son útiles para entender el éxito de un investigador, pero no cuál es el trabajo o trabajos que han causado dicho éxito. La mayor parte de estos sistemas, simplemente proporcionan directamente listados de los artículos publicados para que sean manualmente revisados.

La aparición de los modelos de lenguaje ha ampliado enormemente la capacidad de extraer información descriptiva de documentos. Esto permite la generación de resúmenes de grandes volúmenes de información, aplicables a los artículos publicados por un investigador o grupo de investigación, con el objetivo de generar informes que faciliten la identificación de su campo de investigación.

En este contexto, ya existen aproximaciones generales centradas en la generación de resúmenes de textos largos. Pilault et al. [8] proponen un enfoque híbrido para la generación de textos largos consistente en un proceso de extracción de las partes más relevantes de un documento usando un modelo que opera frase a frase y, posteriormente, proporciona ese resumen a un transformer para que genere el resumen final en base a dichas frases. Zhang et al. [12] describen un sistema multi-etapa para la generación del resumen de un documento para situaciones en las que los textos son más grandes que la ventana de contexto de los modelos de lenguaje. Esto lo realiza partiendo el documento y generando resúmenes de los fragmentos, que se van refinando hasta obtener el resumen final. Sarthi et al. [9] presentan un sistema de indexación y recuperación orientado a enriquecer el contexto de los modelos de lenguaje. El enfoque se basa en una estructura en forma de árbol que organiza resúmenes a distintos niveles de abstracción, permitiendo capturar información tanto de alto como de bajo nivel del texto. Wang et al. [11] muestran un modelo de generación de resúmenes que se centra en tener en cuenta la estructura jerárquica de los documentos y del contexto local de cada sección. Para ello, emplea un transformer con un contexto mayor y un módulo de memoria a corto y largo plazo (LSTM) para la extracción de temas a nivel de segmento. Ou and Lapata [7] proponen una mejora en los métodos de resumen jerárquico aplicados a documentos largos. En vez de simplemente extraer fragmentos de documentos, resumirlos y combinarlos, enriquecen dichas etapas incluyendo en los resúmenes intermedios fragmentos del texto original mediante reemplazo, apoyo o alineación con citas. Cardoso and Pardo [2] proponen un sistema para producir resúmenes de colecciones de documentos, centrándose en la segmentación en subtemas como criterio explícito de selección, de forma que el resumen automático garantice la cobertura de cada subtema modelando relaciones intra e interdocumentales. Finalmente, DeYoung et al. [4] comparan múltiples modelos de resumen de colecciones de documentos para analizar su calidad. Para ello, utilizan una colección de reseñas de películas y resúmenes de experimentos médicos de multitudes creadores, y aplican un rango de modelos que va desde transformers ajustados hasta GPT-4.

Aunque el estado del arte muestra que hay numerosas propuestas que permiten la realización de resúmenes de documentos largos, estas se centran en documentos individuales o en tareas de recuperación de información con colecciones de documentos. Nuestra contribución presenta una herramienta que integra en

un informe el contenido de múltiples artículos pertenecientes a un investigador, a un grupo de investigación o al conjunto del instituto, y que puede desplegarse sobre infraestructura estándar sin necesidad de hardware especializado.

3. Sistema de gestión de conocimiento

Esta sección describe el sistema de gestión de conocimiento que se ha desarrollado. Está compuesto de un componente de extracción de contenido de PDFs, un componente de identificación de temas y secuencias de temas entre artículos y un componente de generación de informes jerárquico que resume por conjuntos de artículos la colección a procesar hasta obtener un resumen completo.

3.1. Extracción de contenido

Actualmente, existen numerosas herramientas de extracción de texto de un fichero PDF (pueden extraer otros elementos, pero solo usamos el texto). Sin embargo, hay una gran variabilidad en términos de calidad para diferentes tipos de documentos. Al no encontrar una comparativa exhaustiva para su uso con artículos científicos, hemos realizado una con las principales herramientas encontradas durante la revisión del estado del arte y que se describen a continuación:

PyMuPDF[1] Biblioteca Python orientada a la extracción, análisis, conversión y manipulación de datos de documentos en formatos como PDF, EPUB y TXT entre otros. Su extensión PyMuPDF4LLM genera formatos estructurados optimizados para su procesamiento por modelos de lenguaje, detectando encabezados, párrafos, listas, tablas y áreas de figuras. Asimismo, omite cabeceras y pies de página repetitivos presentes en cada página del documento.

PDFPlumber[2] Herramienta de código abierto para la extracción de texto y tablas a partir de documentos PDF. Fundamenta su proceso de extracción en un análisis geométrico, de disposición espacial y estructural del documento.

Apache Tika[3] Biblioteca de análisis de contenido de Apache Software Foundation para la detección y extracción unificada de texto y metadatos en múltiples formatos de documentos. En el caso de los PDFs usa PDFBox, una librería que realiza un análisis estructural y geométrico del documento.

GROBID[4] Biblioteca de aprendizaje automático diseñada para la extracción de información estructurada de documentos académicos (publicaciones técnicas y científicas) en PDF. Proporciona una salida estructurada usando Conditional Random Fields o modelos de aprendizaje profundo, para interpretar información visual del PDF (posición, fuente, estilo, columnas), lo que permite un reconocimiento estructural más robusto.

CERMINE[5] Biblioteca Java y servicio web para la extracción de metadatos y contenido de artículos científicos. El sistema analiza el contenido de un archivo PDF y extrae información bibliográfica (título, autores, palabras clave, resumen, entre otros) y estructural (secciones, subsecciones, párrafos). Esto se realiza mediante modelos como Máquinas de Vectores de Soporte, K-medias para agrupaciones de contenido y Campos Aleatorios Condicionales.

Science Parse[6] Herramienta del Allen Institute for AI que analiza artículos científicos en PDF y los devuelve de forma estructurada en JSON. Los campos que extrae incluyen título, autores, resumen y secciones. El procesamiento del contenido está basado en el uso de Campos Aleatorios Condicionales para la identificación de los distintos elementos del documento PDF.

La comparación de las herramientas ha usado un corpus de 20 artículos científicos en PDF seleccionados aleatoriamente con las siguientes restricciones. Para que la evaluación comparativa fuese representativa, el corpus se ha construido con documentos heterogéneos, con maquetaciones diferentes en cuanto a estructura, tipografías, distribución en columnas, presencia de tablas, figuras y organización de los metadatos iniciales. El contenido de dichos artículos se ha extraído y etiquetado manualmente, indicando la sección a la que pertenece cada fragmento. Las secciones evaluadas son: título, autores, resumen (abstract), palabras clave (keywords), introducción, discusión, conclusiones y agradecimientos.

La metodología de evaluación sigue el marco propuesto por Meuschke et al. [6], que implica la conversión a minúsculas y normalización de espacios en blanco del texto de referencia y el extraído. La similitud entre ambos se calcula mediante la distancia de Levenshtein. Consideramos que dos tokens coinciden cuando su similitud supera el umbral $\tau = 0{,}7$ en la matriz de similitud. A partir de esta correspondencia se calculan distintas métricas: la precisión (fracción de tokens extraídos correctamente), exhaustividad (cobertura del conjunto de referencia por parte de los elementos extraídos), F1, exactitud (razón de Levenshtein sobre el texto completo) y tasa de detección de cada herramienta (porcentaje de secciones detectadas). La tabla 1 muestra los resultados obtenidos en dichas métricas, junto con el tiempo medio de procesamiento por PDF de la colección (T(s)) en un computador de oficina con 16 GB de RAM y procesador Intel i7-10700T.

Extractor	**Prec.**	**Exhaust.**	**F1 ($\mu \pm \sigma$)**	**Exact.**	**Detec.**	**T(s)**	**Sig.**
GROBID	0.9629	0.9444	0.93 ± 0.07	0.9174	96.79 %	40.50	—
ScienceParse	0.9448	0.8417	0.82 ± 0.08	0.8018	87.50 %	07.22	***
PyMuPDF	0.8341	0.8904	0.81 ± 0.09	0.7443	98.12 %	00.41	***
CERMINE	0.9393	0.8204	0.79 ± 0.12	0.7839	83.75 %	59.29	**
Tika	0.7580	0.9026	0.77 ± 0.08	0.7116	98.12 %	01.34	***
PyMuPDF4LLM	0.9087	0.6811	0.65 ± 0.08	0.6393	76.78 %	09.91	***
PDFPlumber	0.7157	0.6179	0.53 ± 0.16	0.5383	78.75 %	08.89	***

Tabla 1. Comparativa de extractores de texto (media ± desviación típica en F1; prueba de Friedman global $\chi^2(6) = 71{,}21$, $p < 0{,}001$, $N = 20$; **Sig.** indica el resultado del post-hoc de Wilcoxon con corrección de Bonferroni frente a GROBID: *** $p_{adj} < 0{,}001$, ** $p_{adj} < 0{,}01$, * $p_{adj} < 0{,}05$)

En la tabla se observa que GROBID presenta mejores resultados en la mayoría de las métricas, excepto en la tasa de detección, donde es superado por PyMuPDF y Tika, y en tiempo de procesamiento, donde es el segundo más lento.

Esta menor tasa de detección implica que, en algunas situaciones, no identifica secciones completas del documento que sí son detectadas por otras herramientas. Respecto al tiempo, aunque la diferencia es significativa, el procesamiento en segundo plano reduce su relevancia como criterio de decisión, más allá de ser conscientes del tiempo de espera requerido para procesar las colecciones. Teniendo en cuenta esto, hemos usado GROBID como herramienta de extracción principal, pero con un sistema de respaldo basado en ScienceParse. Hemos decidido usar este último ya que, aunque presenta peores resultados que otros, cubre mejor las situaciones en las que GROBID no consigue extraer contenido.

3.2. Preparación del almacén de datos

El flujo seguido para generar el almacén de información usado es descrito en la figura 1. A partir de un listado de metadatos de las publicaciones en sistema, el componente de descarga se adapta a la fuente de los documentos. En la figura aparece como ejemplo Zaguan[7], que es el repositorio institucional de documentos de la Universidad de Zaragoza, pero lectores específicos para múltiples editoriales pueden ser añadidos como plugins. El almacén con los PDFs descargados se procesa con GROBID + ScienceParse, almacenando el resultado en crudo. A partir de estos resultados de extracción, se aplica un paso de transformación para limpiar las secciones relevantes de cada documento, que facilita su uso posterior. Adicionalmente, se procesan los documentos para extraer sus temas, tanto los existentes en palabras claves como otros que se puedan obtener en secciones como conclusiones y trabajo futuro. El objetivo es tener pre-calculada esta información para que la herramienta desarrollada pueda mostrar rápidamente tanto estadísticas de campos de trabajo de los investigadores y grupos, como líneas de trabajo temáticas que se han mantenido a lo largo del tiempo. Este proceso de identificación de temas puede ser realizado por cualquier LLM al que se conecte, pero en este trabajo hemos comparado Llama 3.2 3B[8] y Qwen 3 14B[9], por ser ambos modelos abiertos capaces de ejecutarse en el hardware disponible. Siendo Llama un modelo que, a pesar de ser mucho más pequeño que Qwen, ha demostrado ser valido en muchos contextos de procesamiento del lenguaje.

3.3. Componente de generación de informes

El objetivo de este componente es resumir, bajo diferentes criterios, el trabajo de un investigador o grupo de investigación en base a las publicaciones integradas en el sistema. El proceso de generación de dichos informes es también realizado por un LLM (Llama 3.2 3B o Qwen 3 14B en nuestro contexto).

El proceso realizado está descrito en la figura 2. El prompt proporcionado a dicho modelo incluye como elementos principales la tarea a realizar, y la longitud y detalle del resultado a generar. Sin embargo, por limitaciones del modelo,

[7] https://zaguan.unizar.es/
[8] https://huggingface.co/meta-llama/Llama-3.2-3B
[9] https://huggingface.co/Qwen/Qwen3-14B

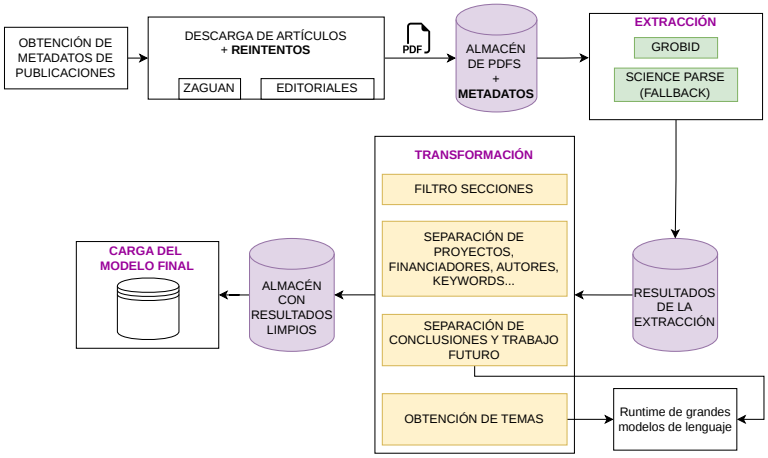


Figura 1. Proceso de generación del almacén de conocimiento

no es algo que se pueda hacer directamente. El principal desafío técnico de la generación de este resumen reside en que el volumen de información de las publicaciones de un investigador o grupo puede superar fácilmente la ventana de contexto del modelo de lenguaje empleado. Es decir, el límite máximo de texto que el modelo puede procesar en una petición, ya que un grupo de investigación puede haber producido centenares de publicaciones en el periodo de interés y, por cada artículo, el contenido relevante (resumen, introducción, discusión, conclusiones, temas) puede ocupar varias páginas. Lo cual excede con mucho la capacidad de contexto de cualquier LLM que requiera pocos GBs de RAM.

La solución trivial de enviar únicamente fragmentos cortos de cada artículo o de reducir el número de artículos considerados implicaría una pérdida significativa de información y comprometería la calidad del informe. Por tanto, en base a los trabajos indicados en el estado del arte, se ha optado por una estrategia de resúmenes jerárquicos en forma de árbol, que permite procesar conjuntos de artículos cuyo contenido agregado supera la ventana de contexto del modelo. Este enfoque divide la información en lotes manejables, generando resúmenes intermedios y consolidándolos progresivamente hasta obtener una narrativa final.

Antes de invocar al modelo, y para facilitar la sustitución del mismo, el sistema consulta la ventana de contexto disponible y calcula la cantidad de tokens que puede utilizar como entrada. Si todo el contenido cabe en una única petición, el informe se genera directamente mediante una sola llamada. En cambio, si el volumen de artículos supera dicha capacidad, el sistema agrupa los artículos por año y los divide en lotes ajustados al tamaño de contexto disponible.

Para cada lote se realiza una llamada independiente al modelo de lenguaje. La salida no se limita a texto libre, sino que adopta una estructura intermedia en formato JSON: fragmentos breves de resumen acompañados de los identifica-

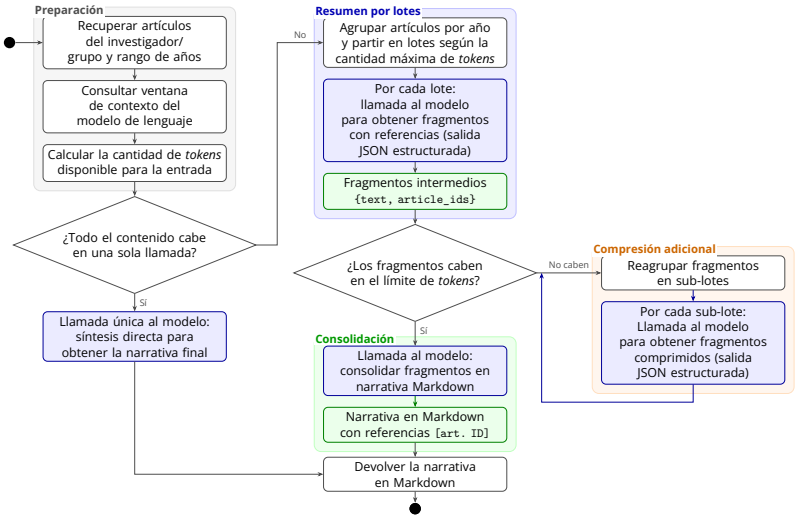


Figura 2. Proceso de generación de informes

dores de los artículos fuente. Esta representación permite conservar trazabilidad granular, de modo que cada bloque narrativo pueda vincularse explícitamente con los artículos de los que procede la información. La figura 3 muestra un resumen del prompt con las principales secciones indicadas.

Una vez generados los fragmentos intermedios, el sistema comprueba si caben en una nueva petición de consolidación. Si caben, se integran en una narrativa final en formato Markdown con referencias explícitas a los artículos. Si no caben, se reagrupan en nuevos sub-lotes y se resumen de nuevo, produciendo fragmentos más compactos. Este proceso de compresión puede repetirse de forma recursiva, garantizando que en todos los pasos cada entrada se ajuste a la ventana de contexto disponible. Este enfoque presenta las siguientes ventajas clave:

- Escalabilidad. El sistema puede trabajar con colecciones amplias de artículos sin depender de que todo el contenido quepa en una única llamada al modelo.
- Control del contexto. El tamaño de los lotes se determina en función de la ventana de contexto del modelo o del límite de tokens configurado.
- Trazabilidad. Los fragmentos intermedios mantienen los identificadores de los artículos fuente, lo que permite que el informe final conserve referencias explícitas a las publicaciones que respaldan cada afirmación.

4. Aplicación de usuario

Usando los componentes descritos anteriormente, se ha desarrollado una aplicación gráfica que permite a un usuario analizar la colección de artículos de forma sencilla. La figura 4 muestra los componentes de la aplicación desarrollada. El

```
# MAP: salida estructurada mediante JSON Schema
def get_map_json_schema( max_fragments: int) -> dict: return {
...
PROMPT_MAP_SYSTEM = You are an expert academic analyst. Synthesise
research articles into a concise thematic JSON summary
...
STRICT RULES:
- Write in English , IMPERSONAL style. Never use "we"/"our"/"I".
...
DIVERSITY  ( CRITICAL):
- Every fragment must describe a DIFFERENT research topic.
...
WHAT TO CAPTURE per fragment (pick the most relevant):
- The main research lines and topics covered.
...
Produce between { min_fragments} and { max_fragments} JSON fragments.
...
```

Figura 3. Resumen de las secciones del prompt

frontend es una aplicación web de tipo SPA (Single Page Application) que actúa como capa de presentación del sistema. Su función principal consiste en solicitar al backend la información a visualizar y renderizarla para el usuario. Además, recibe eventos en tiempo real mediante SSE (Server-Sent Events) para monitorizar procesos de larga duración, como la ejecución del flujo ETL y la generación de informes a partir de grandes colecciones de artículos. El backend realiza las tres principales tareas del sistema. La primera es la gestión del flujo ETL de extracción del contenido de los PDFs (orquestador de workflows). Esto incluye la ejecución periódica, el registro de errores y la posibilidad de corrección manual desde el frontend. La segunda es generación de estadísticas de publicaciones, temas, y evolución de las líneas de investigación. El último es la generación de los informes textuales de investigadores y grupos en los rangos de años seleccionados, así como su transformación a múltiples formatos tales como PDF, DOCX, o Markdown, entre otros. Dado que la generación de cada informe cuesta varios minutos, la interfaz proporciona información de progreso para poder dejarla en segundo plano, pero pudiendo saber en qué punto del proceso se encuentra en todo momento.

Para comprobar la calidad de los informes y la influencia del LLM en dicha generación, se desarrolló una encuesta de 10 preguntas sobre los diferentes elementos en el informe. Dicha encuesta se proporcionó a investigadores del Instituto de Investigación en Ingeniería de Aragón (I3A). A cada investigador se le dio dos variantes de informes sobre su grupo, y dos sobre su trabajo personal, siendo cada variante generada con uno de los LLM comparados; Llama 3.2 3B como modelo más pequeño y simple y Qwen 3 14B como modelo más grande y mayor capacidad teórica de generar mejores respuestas. Se consiguieron 18 respuestas con un promedio de 7.45 sobre 10 en los informes generados con Llama y

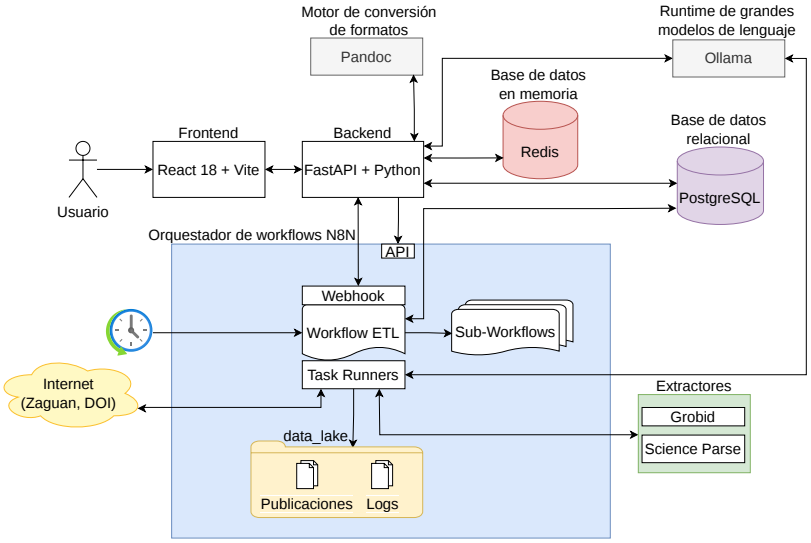


Figura 4. Arquitectura de componentes de la aplicación de usuario

8.67 en el caso de Qwen, mostrando la capacidad de estos sistemas para realizar el trabajo deseado y la mejora que produce el uso de un LLM de mayor tamaño.

La figura 5 muestra la interfaz del sistema de generación de informes. Desde ella es posible configurar las características del informe, que van desde el tipo de entidad a analizar (investigador, grupo o instituto), el intervalo temporal y los elementos a incluir. En la parte derecha se presenta el estado de la generación, el resultado obtenido y la opción de descarga. Al resto de funcionalidades se accede mediante las pestañas de la izquierda. Para usuarios básicos, el sistema ofrece la visualización de temas y la evolución de las líneas de investigación, información que puede incorporarse al informe marcando las opciones correspondientes. Para la administración del sistema, se proporciona un módulo de ejecución y monitorización del proceso ETL de adquisición de los artículos, un sistema de revisión manual de la información extraída, un dashboard de estadísticas y un conjunto de opciones de configuración de las fuentes de datos.

5. Conclusiones y Trabajo futuro

Este trabajo ha presentado una aplicación inteligente para la gestión del conocimiento científico en grupos de investigación, basada en tres tareas principales: extracción de contenido de documentos PDF, identificación de temas y líneas de investigación, y generación automática de informes mediante modelos de lenguaje abiertos. Este sistema se ha desarrollado completamente con librerías y modelos abiertos, permitiendo su uso sin necesidad de licencias y en entornos sin requisitos de hardware especializado.

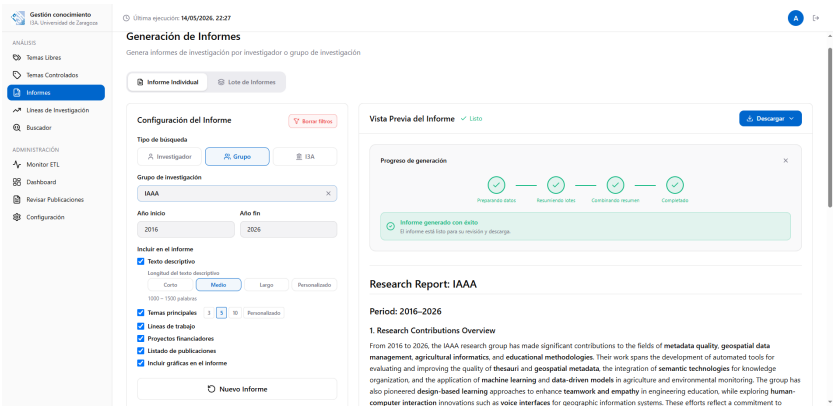


Figura 5. Interfaz de la aplicación

La comparativa realizada entre herramientas de extracción de contenido ha mostrado que GROBID ofrece el mejor rendimiento global para el procesamiento de artículos. Sin embargo, todas las herramientas analizadas presentan problemas significativos en unas u otras partes del proceso, lo que evidencia que el campo de procesamiento de PDFs sigue abierto y con margen para numerosos avances. Respecto a la estrategia de resumen jerárquico, esta ha demostrado ser una solución eficaz para superar las limitaciones de la ventana de contexto de los modelos de lenguaje de menor tamaño.

La evaluación ha mostrado que el sistema, en general, genera resultados bien valorados. La narrativa inicial fue destacada como la sección más útil, por ofrecer una buena visión general de la actividad, seguida de las líneas de investigación y los gráficos de temas. Sin embargo, en casos sueltos el sistema no realizo alineamientos de temas correctos, mezclando contextos e introduciendo errores.

Como trabajo futuro, el primer paso consiste en corregir los problemas identificados en las evaluaciones. Otra tarea es ampliar el sistema de descarga de los artículos, permitiendo que, a partir del DOI, se acceda directamente a la revista donde está disponible el trabajo y se recupere el contenido cuando exista acceso disponible. También se quiere evaluar la perdida de información producida en los resúmenes, según el número de niveles aplicados, y como minimizarla. Finalmente, se desea añadir un sistema de recomendación de contactos, en el que agentes puedan solicitar al instituto información sobre expertos y grupos adecuados para colaborar en torno a un tema, propuesta o contrato, permitiendo así identificar y poner en contacto a las personas o entidades más apropiadas para cada caso, facilitando la conexión entre investigadores y organizaciones afines.

Agradecimientos

Este trabajo es parte del proyecto de I+D+i PID2024-155657OB-C21:AGRARD financiado por el Gobierno de España.

Referencias

1. Burnham, J.F.: Scopus database: a review. Biomedical digital libraries **3**(1), 1 (2006)
2. Cardoso, P.C., Pardo, T.A.: Multi-document summarization using semantic discourse models. Procesamiento del Lenguaje Natural **1**(56), 57–64 (2016)
3. Delgado, E., Repiso, R.: El impacto de las revistas de comunicación: comparando google scholar metrics, web of science y scopus. Comunicar **21**(41), 45–52 (2013)
4. DeYoung, J., Martinez, S.C., Marshall, I.J., Wallace, B.C.: Do multi-document summarization models synthesize? Transactions of the Association for Computational Linguistics **12**, 1043–1062 (2024)
5. Manca, S.: Researchgate and academia.edu as networked socio-technical systems for scholarly communication: a literature review. Research in Learning Technology **26**, 1–16 (2018)
6. Meuschke, N., Jagdale, A., Spinde, T., Mitrović, J., Gipp, B.: A Benchmark of PDF Information Extraction Tools Using a Multi-task and Multi-domain Evaluation Framework for Academic Documents. In: Information for a Better World: Normality, Virtuality, Physicality, Inclusivity, LNCS, vol. 13972, pp. 383–405, Springer Nature Switzerland (2023)
7. Ou, L., Lapata, M.: Context-aware hierarchical merging for long document summarization. In: Findings of the Association for Computational Linguistics: ACL 2025, pp. 5534–5561 (2025)
8. Pilault, J., Li, R., Subramanian, S., Pal, C.: On extractive and abstractive neural document summarization with transformer language models. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp. 9308–9319 (2020)
9. Sarthi, P., Abdullah, S., Tuli, A., Khanna, S., Goldie, A., Manning, C.: Raptor: Recursive abstractive processing for tree-organized retrieval. In: International Conference on Learning Representations, pp. 1–22 (2024)
10. Vine, R.: Google scholar. Journal of the Medical Library Association **94**(1), 97 (2006)
11. Wang, T., Yang, C., Zou, M., Liang, J., Xiang, D., Yang, W., Wang, H., Li, J.: A study of extractive summarization of long documents incorporating local topic and hierarchical information. Scientific Reports **14**(1), 10140 (2024)
12. Zhang, Y., Ni, A., Mao, Z., Wu, C.H., Zhu, C., Deb, B., Awadallah, A., Radev, D., Zhang, R.: Summn: A multi-stage summarization framework for long input dialogues and documents. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1592–1604 (2022)