

Trabajo de Grado

Prototipo de software de extracción de contexto de referencias nombradas en documentos científicos

Presentado por:

Daniel Otero Robles

Maestría con énfasis en Ingeniería de Sistemas y Computación

daniel.otero@correounivalle.edu.co

Director:

Raul Ernesto Gutiérrez de Piñerez Reyez, Ph.D.

FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN
UNIVERSIDAD DEL VALLE
2025

Resumen

Este trabajo presenta el diseño e implementación de un prototipo de software para el análisis automático del contexto semántico de referencias en textos científicos. El enfoque híbrido propuesto integra técnicas de procesamiento de lenguaje natural, sistemas de recuperación aumentada por generación (RAG) y agentes inteligentes basados en modelos de lenguaje grandes (LLMs).

La metodología desarrollada implementa un pipeline de cinco etapas: extracción estructurada de contenido PDF, chunking y vectorización semántica, construcción de bases vectoriales duales, sistema multi-agente especializado, y evaluación comparativa. El sistema integra cinco LLMs (GPT-4o Mini, Gemini-2.0 Flash, Deepseek R1 Distill Qwen 32B, Ministral 8B 2410, y Llama-3.3 70B Instruct) para analizar el contexto de citas bibliográficas.

Los resultados experimentales demuestran la efectividad del enfoque multi-agente, con consistencia temporal robusta (diferencia máxima de latencia de 0.22 segundos entre modelos) y especialización por tipo de consulta. Ciertos modelos mostraron mayor eficiencia en tareas específicas como localización de referencias y análisis de herramientas, versus análisis metodológico y comparativo.

Las contribuciones principales incluyen: un framework práctico para análisis semántico automatizado de citas, evaluación comparativa de LLMs en tareas de comprensión científica, y demostración de la viabilidad de sistemas RAG multi-agente para análisis de literatura académica.

Palabras clave: *análisis de citas, contexto semántico, RAG, agentes inteligentes, LLMs, cienciometría.*

Índice

1. Introducción	1
2. Justificación del problema	3
2.1. Pregunta de investigación	3
2.2. Objetivo general	4
2.3. Objetivos específicos	4
3. Estado del Arte	5
3.1. Métodos computacionales para el análisis del contenido semántico de referencias nombradas	6
3.2. Enfoque Sintáctico	6
3.3. Enfoque Semántico	10
3.4. Enfoque híbrido	17
3.4.1. Enfoque híbrido estadístico	17
3.4.2. Enfoque basado en Deep Learning	19
3.5. Enfoque contextual (LLMs & RAG)	28
4. Marco Conceptual	39
4.1. LLMs y prompt engineering	39
4.1.1. LLMs	39
4.1.2. Prompt engineering	39
4.2. Modelos LLM usados en este documento	40
4.3. Análisis Comparativo de Técnicas de Entrenamiento	43
4.3.1. Reinforcement Learning from Human Feedback (RLHF) . . .	43
4.3.2. Direct Preference Optimization (DPO)	43
4.3.3. Destilación de Conocimiento	43
4.4. Arquitectura RAG	44
4.5. Sistemas multi-agentes	45
5. Metodología	48
5.1. Descripción del Dataset	48
5.2. Extracción estructura PDF	50
5.3. Chunking y vectorización	53
5.4. Bases vectoriales	54
5.5. Desarrollo del sistema de agentes	54
5.6. Evaluación y validación	59
6. Interfaz de Usuario y Experiencia de Interacción	61
7. Resultados y discusión	66
7.1. Resultados	66
7.2. Discusión	71
7.2.1. Validación de la Propuesta Metodológica	71

7.2.2.	Consistencia y Robustez Multi-LLM	71
7.2.3.	Especialización por Tipo de Consulta	72
7.2.4.	Escalabilidad de la Dificultad	72
7.2.5.	Análisis Diferenciado del Desempeño por Agente	72
7.2.6.	Comparación del Sistema Propuesto con el Estado del Arte . .	74
7.3.	Contribuciones al Campo	77
7.3.1.	Metodológica	77
7.3.2.	Técnica	77
7.3.3.	Aplicada	78
7.4.	Limitaciones Identificadas	78
7.4.1.	Limitaciones Metodológicas	78
7.4.2.	Limitaciones Técnicas	79
7.4.3.	Limitaciones de Escalabilidad	79
8. Conclusiones y Trabajo Futuro		80
8.1.	Conclusiones	80
8.2.	Trabajo Futuro	81
8.2.1.	Extensiones Metodológicas Inmediatas	81
8.2.2.	Avances Arquitectónicos	81
8.2.3.	Aplicaciones Emergentes	82
8.2.4.	Investigación Fundamental	82
8.3.	Reflexión Final	82
9. Anexos		84
9.1.	Dataset evaluación	84
9.2.	Prompt agente detección dominio	86
9.3.	Sistema de evaluación.	87

Índice de figuras

1.	Relación entre enfoques metodológicos y tareas en el análisis de contexto y citas.	5
2.	Pipeline RAG. Tomado de: https://medium.com/@annoliveai/rag-and-the-future-41196993bc9e	45
3.	Pipeline metodología propuesta	48
4.	Arquitectura de agentes.	58
5.	Interfaz principal de PaperRAG mostrando las secciones de carga de documentos y consultas RAG.	61
6.	Interfaz de carga mostrando un documento PDF seleccionado y listo para procesar	62
7.	Sección de consultas RAG con selector de modelo LLM y campo de entrada de preguntas.	63
8.	Ejemplo de consulta sobre localización de referencias bibliográficas específicas.	63
9.	Análisis detallado del contexto de uso de una referencia específica.	64
10.	Respuesta del sistema a una consulta conceptual sobre el tema principal del documento.	64
11.	Latencia promedio por modelo de lenguaje evaluado.	66
12.	Análisis de outliers por LLM utilizando método IQR	67
13.	Latencia promedio por LLM y nivel de dificultad	68
14.	Ranking de performance por tipos de pregunta (menor latencia = mejor performance)	68
15.	Heatmap de latencia promedio por tipo de pregunta y LLM	69
16.	Distribución de longitud de respuestas por modelo	70
17.	Sistema de evaluación.	87

Índice de cuadros

1.	Clasificación de metodologías para el análisis de citas.	7
2.	Comparación de diferentes modelos de redes neuronales	20
3.	Resultados de p-CNN, NCN y NPM en métricas MAP y MRR (top-10) reportados en [1]	22
4.	Resultados de Du et al. (2018) en diferentes datasets	23
5.	Distribución de clases en los conjuntos de datos utilizados por Karim et al. [2].	24
6.	Resultados de precisión (accuracy) y F1-score obtenidos por diferentes modelos en el análisis de sentimiento de citas (basado en Karim et al. [2]).	24
7.	Resultados de [3]: variantes LSTM en recomendación contextual (top-10)	26
8.	Resultados de [3]: comparación personalizada vs. no personalizada	27
9.	Resultados de [3]: comparación con otros métodos (top-10)	27

10.	Resultados de extracción de contexto de cita en AAN (Dai et al., 2020)	28
11.	Resultados de extracción de contexto de cita en DBLP [4]	28
12.	Macro F1-score en clasificación de intención de cita en ACL-ARC y SciCite (Lahiri et al., 2023).	29
13.	Precisión y kappa de GPT-3.5-turbo en anotación de propósito y sentimiento de cita en [5].	31
14.	Macro F1-score en clasificación de intención de cita usando fine-tuning en ACL-ARC y SciCite (Motrichenko et al., 2021).	32
15.	Resultados de fine-tuning de modelos Transformer en el dataset FO-CAL (Gowda et al., 2024).	32
16.	Resumen de los datasets utilizados en Li et al. (2024).	33
17.	Comparación de desempeño (F1/Accuracy) entre context window largo y variantes RAG (Li et al., 2024).	34
18.	Descripción de los conjuntos de datos utilizados en [6].	34
19.	Desempeño de GRAG y métodos comparados en WebQSP (F1, Hit@1, Recall) y ExplaGraphs (Accuracy).	35
20.	Desempeño de PaperQA2, humanos y sistemas automáticos en el benchmark LitQA2.	38
21.	Estrategias de prompting	40
22.	Distribución del dataset por dominio académico	49
23.	Distribución de estilos de citación en texto	49
24.	Estadísticas estructurales del dataset	49
25.	Comparación Regex vs. LLMs en procesamiento científico	51
26.	Actualización de atributos por agente en el pipeline	57
27.	Métricas de desempeño por agente	73
28.	Patrones de fallo diferenciados por agente	74
29.	Comparación resumida del sistema propuesto con el estado del arte	76
30.	Plantillas de preguntas GPQA por nivel de dificultad	85

1. Introducción

En la última década, los modelos de lenguaje de gran escala (LLMs, por sus siglas en inglés) han transformado de manera profunda el panorama de la inteligencia artificial, ofreciendo capacidades sin precedentes en tareas de generación, comprensión y razonamiento sobre texto natural [7, 8]. La convergencia de avances en arquitecturas de *transformers* [9, 10], técnicas de entrenamiento a gran escala y acceso a recursos computacionales masivos ha permitido la emergencia de modelos cada vez más versátiles y especializados [11, 12].

En este contexto, el desarrollo de sistemas multi-agente basados en LLMs abre nuevas oportunidades para abordar problemas complejos que requieren la integración de múltiples perspectivas, especializaciones y flujos de razonamiento coordinados [13, 14]. A diferencia de los sistemas monolíticos, un enfoque multi-agente posibilita la asignación de tareas diferenciadas a modelos optimizados para distintos dominios, potenciando tanto la eficiencia como la calidad de las respuestas [15].

Sin embargo, la diversidad de arquitecturas, técnicas de alineación y estrategias de entrenamiento de los LLMs disponibles plantea el reto de seleccionar un conjunto representativo que maximice la complementariedad y minimice redundancias [16, 17]. La elección de los modelos adecuados no solo impacta el rendimiento final del sistema, sino también su viabilidad técnica y económica, especialmente en escenarios que demandan análisis de información científica a gran escala.

Esta tesis se propone diseñar y evaluar un sistema multi-agente para el análisis automatizado de contexto semántico en literatura científica, explorando el impacto de diferentes combinaciones de modelos de lenguaje en la calidad y eficiencia de la extracción de información. Para ello, fue seleccionado un conjunto de LLMs que representen la diversidad del panorama actual, considerando factores como: (i) arquitectura y filosofía de diseño, (ii) tamaño y eficiencia computacional, (iii) técnicas de entrenamiento empleadas, (iv) rendimiento en *benchmarks* científicos, y (v) disponibilidad y reproducibilidad [18, 19, 20].

La investigación se estructura de la siguiente forma: en el Capítulo 4 se presenta el marco teórico y el estado del arte de los LLMs y sistemas de agentes; en el Capítulo 5

se detalla la metodología de selección, integración y evaluación de modelos; en el Capítulo 7.1 se discuten los resultados obtenidos; y finalmente, en el Capítulo 8 se presentan las conclusiones y posibles líneas de trabajo futuro.

2. Justificación del problema

En la producción científica contemporánea, las citas cumplen un papel central en la construcción y validación del conocimiento. Sin embargo, más allá de señalar la fuente, cada cita aporta un contexto específico que permite comprender cómo un autor utiliza, discute o contrasta una referencia dentro de su argumento. Identificar y extraer ese contexto resulta fundamental para interpretar de manera precisa la función de la cita en el discurso académico.

A pesar de su importancia, los enfoques tradicionales para el análisis de referencias se han concentrado principalmente en el conteo de citas o en la simple vinculación con las bases de datos bibliográficas. Estas aproximaciones, aunque útiles, resultan limitadas porque no capturan la riqueza semántica y discursiva que acompaña a cada referencia. En consecuencia, se pierde información valiosa para el análisis bibliométrico, la construcción de mapas de conocimiento y la comprensión del impacto real de un trabajo científico.

El avance de los modelos de lenguaje y las técnicas de procesamiento de lenguaje natural ofrece una oportunidad para superar estas limitaciones. En particular, el diseño de sistemas capaces de extraer el contexto de las citas directamente de documentos científicos en formato PDF representa un aporte significativo tanto en el ámbito académico como en el tecnológico. Este tipo de soluciones permite no solo mejorar la trazabilidad y el análisis de la producción científica, sino también ofrecer herramientas más robustas para investigadores, instituciones y comunidades interesadas en comprender cómo circula y se transforma el conocimiento.

Por estas razones, el presente trabajo se orienta a abordar la necesidad de un sistema que automatice la extracción del contexto de las citas en documentos científicos, con el fin de contribuir al fortalecimiento de las prácticas de análisis y representación de la información académica.

2.1. Pregunta de investigación

¿Cómo diseñar un sistema que permita extraer de manera automática el contexto asociado a las citas en documentos científicos en formato PDF, garantizando precisión y relevancia en la información recuperada?

2.2. Objetivo general

Diseñar un sistema para la extracción del contenido semántico-contextual de las citas en documentos científicos, con el fin de mejorar el análisis y la representación de la información académica.

2.3. Objetivos específicos

- Implementar un mecanismo de procesamiento de documentos científicos en formato PDF para identificar y segmentar las citas presentes en el texto.
- Desarrollar un modelo para la extracción del contexto asociado a cada cita, mediante la recuperación de las oraciones o fragmentos relevantes.
- Integrar los componentes en un prototipo funcional que permita la exploración y análisis de citas con su respectivo contexto.
- Evaluar el desempeño del sistema en términos de precisión y relevancia de la información extraída.

3. Estado del Arte

El análisis computacional del contenido semántico de las referencias nombradas en la literatura científica ha transitado por diferentes paradigmas metodológicos, desde aproximaciones basadas en reglas sintácticas y análisis semántico tradicional, hasta modelos híbridos, estrategias de aprendizaje profundo y, recientemente, la incorporación de grandes modelos de lenguaje y técnicas de recuperación aumentada. Cada uno de estos enfoques ha aportado soluciones y retos particulares para abordar tareas fundamentales como la extracción automática de fragmentos de cita, el análisis de polaridad y propósito, la recomendación de referencias relevantes y la identificación avanzada de contexto.

La diversidad de métodos y tareas abordadas requiere de un marco de referencia integrador que permita visualizar de manera global la cobertura y limitaciones de cada paradigma. La Figura 1 sintetiza la relación entre los principales enfoques computacionales y las tareas centrales del campo, mostrando cómo la evolución tecnológica ha ampliado progresivamente el espectro de problemas abordables y la profundidad del análisis posible. Esta panorámica sirve como punto de partida para la revisión detallada de métodos, recursos y hallazgos recientes que se desarrollan en la sección siguiente.

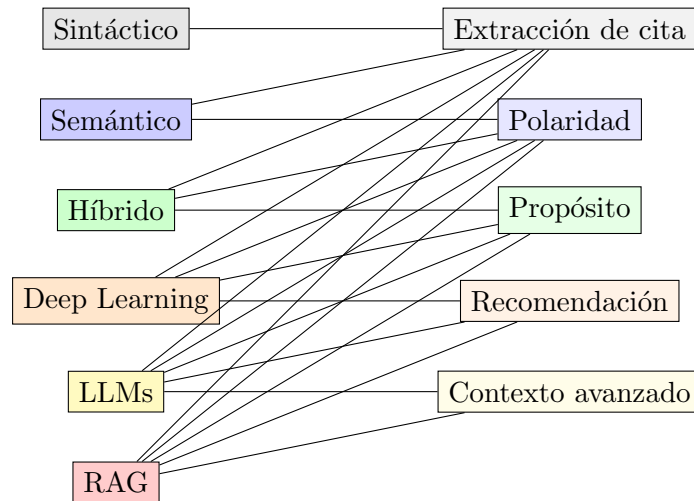


Figura 1: Relación entre enfoques metodológicos y tareas en el análisis de contexto y citas.

3.1. Métodos computacionales para el análisis del contenido semántico de referencias nombradas

El análisis de las citas en documento de texto se divide en cuatro categorías –o enfoques–: sintáctico, semántico, híbrido, y LLMs. Todos los trabajos presentados aquí se han clasificado dentro de estas categorías, pues no se encontraron documentos que realizaran análisis por fuera de estos enfoques.

En cada enfoque, sin embargo, se utilizan distintas metodologías para desarrollar el concepto subyacente a la categoría. El cuadro 1 agrupa cada tecnología por método y muestra cómo han sido aplicadas estas tecnologías.

En las siguientes secciones se va a desarrollar a más profundidad los enfoques presentados en el cuadro 1. El objetivo es poder explorar de forma exhaustiva las metodologías usadas, los resultados y hallazgos encontrados en cada artículo presentado.

En primer lugar, se presentarán los métodos relacionados con el enfoque sintáctico, caracterizados por el uso de documentos de texto estructurados, etiquetados y con *tags* que facilitarán el procesamiento y extracción de las referencias nombradas en medio del documento. Luego, se presentarán las metodologías relacionadas con el enfoque semántico, donde abundan modelos de machine learning y técnicas de segmentación del texto; después, se tratarán los enfoques híbridos CCA (Citation Context Analysis), en donde se emplean métodos de los dos enfoques simultáneamente y de forma complementaria. Luego, aparece el enfoque Deep Learning, el cual se vale de datasets taguados para entrenar modelos de embeddings contextuales para inferir el propósito de las citas. Finalmente, se aborda el enfoque LLMs que abunda en más métodos que sus antecesores y usa métodos de Inteligencia Artificial para extraer el contexto de las citas.

3.2. Enfoque Sintáctico

Este enfoque se caracteriza por centrarse en la ubicación de las citas en medio del texto para extraer información relacionada con las citas. También se tiene en cuenta las re-citaciones dentro del documento y sus secciones. El objetivo principalmente es encontrar la forma de la distribución de las citas dentro del documento.

En [21] aplican algoritmos de minería de texto a un conjunto de datos relativamente

Enfoque	Objetivo del enfoque	Método	Descripción
Sintáctico	Se estudian las citas de acuerdo a su posición dentro de la estructura del texto. En general se usan técnicas de conteo o búsqueda de palabras asociadas a la cita para entender la relevancia de la posición de la cita en la estructura del texto.	Text Mining	Las metodologías de text mining se enfocan principalmente en dos métodos de conteo: CountX y CountOne. El primer método, CountX, cuenta el número de veces que una referencia es mencionada en cada sección del texto. El segundo, CountOne, cuenta cada referencia una vez por sección, sin importar cuantas veces es mencionada. Usualmente son estudios de distribución de las referencias a lo largo del texto.
		NLP	Se usan técnicas de NLP para explorar como se distribuyen las citas dentro de la estructura del texto, estilo de la referencia, número de menciones, etc.
		n-gramas	Estas metodologías tratan de caracterizar la distribución de referencias a lo largo del texto, añadiendo a la localización de las mismas, el hallazgo de n-gramas compuestos por, al menos, un verbo, que permita calificar las citas de acuerdo a la sección donde son mencionadas.
Semántico	Se estudian las citas para determinar la naturaleza de la contribución que realizan de acuerdo al texto cercano a la referencia nombrada. Se busca extraer el contexto de la citación para añadir valor cualitativo a la cita o determinar el propósito de la cita.	NLP	En el enfoque semántico con NLP, se categorizan las citas intentando mostrar el propósito de las mismas. Estas categorías buscan establecer el propósito de la cita.
Híbrido	Este enfoque combina el análisis semántico y sintáctico con modelos neuronales profundos para detectar patrones complejos en el texto que rodea a las citas; integra además representaciones semánticas y metadatos para clasificar la importancia o intención de las referencias.	Análisis de sentimientos	Se analiza el sentimiento del autor en relación a la referencia nombrada en el texto. Normalmente este sentimiento se clasifica en tres categorías: neutro, positivo o negativo.
		CCA	Examina el contexto sintáctico y semántico de las citas para obtener una mejor comprensión de las relaciones entre las obras citadas y citantes.
		Modelos neuronales	Uso de arquitecturas como redes convolucionales (CNN), unidades recurrentes con computuras (GRU) y embeddings contextuales como Word2Vec, FastText o BERT. Estas representaciones se combinan con metadatos (por ejemplo, sección del documento) para clasificar las citas como importantes o no, o inferir su propósito.
LLMs	Modelos de lenguaje a gran escala que permiten análisis sofisticado del contexto, intención y función de la cita. Se pueden usar directamente con instrucciones en lenguaje natural o ajustarse para tareas específicas. Permiten operar incluso con pocos o ningún dato etiquetado.	Prompt-based inference	Se diseñan instrucciones en lenguaje natural (prompts) para guiar modelos como GPT o T5 en tareas como clasificación de intención de citas. Puede aplicarse en modo <i>zero-shot</i> (sin ejemplos) o <i>few-shot</i> (con unos pocos ejemplos), lo que permite usar el modelo sin necesidad de reentrenarlo.
		Fine-tuning	Ajuste de modelos preentrenados como BERT, SciBERT o XLNet mediante datos etiquetados para tareas específicas como clasificación de intención o análisis de sentimiento en el contexto de citas.
		RAG	Retrieval-Augmented Generation combina recuperación de información y generación de texto con LLMs, permitiendo análisis contextualizados y actualizados sobre citas científicas.

Cuadro 1: Clasificación de metodologías para el análisis de citas.

grande (866 documentos de texto) para explorar las contribuciones hechas por los artículos –incluidos en la bibliografía– observando la distribución de las citas en las diferentes secciones de los documentos. Además, se comparan trabajos altamente citados haciendo uso de dos métodos de conteo: CountOne y CountX.

Los autores encuentran que la distribución de frecuencia de las citas sigue una ley de potencia para ambas metodologías de conteo. Además, encuentran que la mayoría de los trabajos re-citados (utilizando CountOne) en la muestra de datos tienden a aparecer en las secciones introducción, revisión de la literatura y metodología, mientras que al usar el método CountX, las 3 ubicaciones principales para los trabajos altamente citados son metodología, revisión de literatura e introducción. También concluyen que las clasificaciones de trabajos altamente citados se vieron influenciados por la metodología de conteo usada.

Señalan además, que un trabajo que se ha citado en la introducción o revisión de la literatura, y luego se menciona en las secciones de metodología o discusión, probablemente haya hecho una contribución más significativa al artículo de citación que uno que haya se ha mencionado una sola vez.

Bertin et al. [22] propone una metodología de procesamiento del lenguaje natural para identificar la función de las citas basados en un supuesto: el contexto de las referencias nombradas se asignan en base a la frecuencia de las co-ocurrencias de n -gramas ubicados cerca de las citas y en la ubicación de la cita en la estructura del texto.

De forma puntual, el objetivo de [22] es descubrir los patrones lingüísticos más frecuentes, basados en n -gramas [23]; los cuales se encuentran en el texto cercano a la referencia nombrada, es decir, la oración donde aparece la cita. Luego, los autores evalúan si estos patrones varían según su ubicación en la estructura retórica de los artículos.

Para analizar e identificar las citas en los documentos de texto, los autores siguieron dos pasos:

1. Identificar todos los tags en XML que representan citaciones.
2. Procesamiento de texto completo de todas las oraciones para identificar cualquier cita en el texto que se perdió durante el primer paso usando expresiones regulares. De hecho, algunas citas en el texto en el corpus están presentes en el

texto pero no se identifican como elementos XML.

Los autores eligieron n -gramas de tamaño 3-g. Es decir, secuencias de tres palabras, y tomaron en cuenta solo 3-g dentro de las oraciones que contienen citas.

Los resultados encontrados muestran que la estructura retórica juega un papel importante en la distribución de las clases de n -gramas en los textos. Además, este enfoque permite identificar oraciones que pueden ser anotadas potencialmente con actos de citación.

Atanassova & Bertin [24] estudian la posición de las citas en el texto y la relación entre la fecha de publicación de la referencia nombrada, la recurrencia de la cita¹ y su ubicación en el documento en la estructura IMRaD para buscar dar respuesta a una pregunta: ¿las diferentes secciones de un artículo muestran algún comportamiento específico con respecto a la edad de las referencias recurrentes en el texto que aparecen en ellas?

El interés en las referencias recurrentes, parte del supuesto de que las citas que se mencionan más de una vez en el texto son más importantes que aquellas que se mencionan solo una vez.

El objetivo es pues, examinar la estructura IMRaD –que se usa ampliamente en las ciencias experimentales– e identificar las ubicaciones donde es más probable que aparezcan las citas recurrentes en el texto y la relación de dicha recurrencia con la fecha de publicación del documento citado.

En los resultados encuentran que la recurrencia más común es de 2, además, la antigüedad de las referencias varía considerablemente en todos los documentos.

En [25] se ponderan las citas según la frecuencia (o el lugar) en que aparecen en un texto. Para esto, se presentan dos métodos de conteo: conteo de citas y conteo de autores. En este caso, se hace diferencia entre la recurrencia de citar un artículo o el mismo autor.

Los autores proponen 12 esquemas de conteo que son definidos en dos grupos: 6 de conteo para artículos y 6 de conteo para autores. Se comparan los rankings obtenidos para cada uno de los esquemas de conteo propuestos. El aporte más notable de [25] es la inclusión de la re-citación en el conteo y la diferenciación entre re-citar un artículo

¹Es decir, el número de veces que aparece en el texto

o re-citar un autor, lo que ayuda a crear nuevas categorías o formas de organizar la ciencia.

3.3. Enfoque Semántico

Este enfoque busca dar un valor cualitativo a las referencias nombradas, haciendo uso del texto cercano a la citación. Normalmente, se busca categorizar el propósito de la cita para reducir la complejidad del procesamiento.

Jha et al. [26] utilizan técnicas de NLP para deducir la polaridad y el propósito de una citación por medio de anotación de texto. La polaridad de una cita es la “actitud” –es decir, el sentimiento– del artículo citante frente al artículo citado. Entonces, el agregado de todas las “actitudes” de todas las citas a un documento da una medida cuantitativa de la actitud de la comunidad científica hacia ese documento. El propósito de la cita está relacionado con la motivación y no con la “actitud”.

El análisis de sentimientos –o polaridad– se ejecuta desde el supuesto de que el autor citante tiene un sentimiento acerca de la referencia nombrada y que además, ese sentimiento es posible encontrarlo en el texto escrito cerca a la cita, es decir, las oraciones donde se encuentran la cita.

Para el estudio, [26] crea un conjunto de datos anotado. Seleccionan 30 artículos de AAN que tenían diferentes números de artículos citantes y que fueron citados constantemente desde que se publicaron.

El esquema de polaridad de las citas era positivo, negativo o neutral. Este esquema fue definido de una manera ligeramente diferente a su sentido habitual: una cita está marcada como positiva si indica explícitamente la fuerza del documento citado, o indica que el trabajo realizado en el documento citado ha sido utilizado por el autor citante o un tercero. También se etiqueta como positivo si se compara con otro artículo y se considera mejor de alguna manera.

El criterio de etiquetado como negativa una cita se basa en lo planteado por [27]. Una cita se marca como negativa si apunta explícitamente a una debilidad del documento citado. También se marca como negativa si se compara con otro documento y se considera peor de alguna manera. Una cita es neutral si es solo descriptiva.

Para anotar el propósito de la cita, los autores crean una taxonomía que consta de seis categorías basadas en taxonomías similares propuestas en trabajos anteriores [27, 28].

Seleccionaron las categorías que consideraban más importantes y útiles desde un punto de vista bibliométrico, y que pudiesen ser detectadas a través del análisis del texto de citas.

Las seis categorías propuestas por los autores fueron:

- **Crítico:** La crítica puede ser positiva o negativa. Una oración se clasificaría como crítica cuando menciona las debilidades o fortalezas del artículo citado; critica negativa o positivamente el artículo citado; y evalúa negativa o positivamente el artículo citado.
- **Comparado:** Una cita se clasifica como “comparada” cuando compara o contrasta el artículo citado con el trabajo del autor citante. Si se superpone a la primera categoría y la oración donde aparece la cita dice que un enfoque no es tan bueno como el otro, entonces, ese caso, se usará la primera categoría.
- **Usado:** Una cita se clasifica como “usada” cuando el artículo citante utiliza el método, la idea o la herramienta del artículo citado.
- **Justificado:** Una cita se clasifica como “justificada” cuando los resultados y las afirmaciones del artículo citante fundamentan y verifican el documento citado y se apoyan mutuamente.
- **Fundamental:** Una citación se clasifica como “fundamental” cuando el autor utiliza el trabajo citado como punto de partida –o motivación– y se extiende sobre el trabajo citado.
- **Neutral:** Una cita se clasifica como “neutral” cuando se trata de una descripción neutral del trabajo citado o si no se incluye en ninguna de las categorías anteriores.

El fin de usar dos métodos de anotación, polaridad y propósito, es para conocer la actitud de las citaciones para cada una de las categorías del propósito de las citas. Por ejemplo, cuando una cita es etiquetada como “crítica”, es más probable que sea negativa que positiva. Cuando es “comparada”, lo más común es que sea neutral; cuando es “usada”, normalmente presenta una actitud positiva.

En [29] se explora la técnica word2vec –basada en deep learning– para el análisis de sentimiento de citas. La idea de word2vec (word embeddings²) surgió del concepto de

²Para referirse a este término, se usará la traducción literal del inglés del mismo: incrustaciones de palabras

representación distribuida de palabras. El método común para derivar los vectores es usar un modelo neural probabilístico de lenguaje. Si es “justificada” y “fundamental” la actitud es positiva. Finalmente, si es neutral, la actitud es neutral.

El objetivo de Liu [29] es evaluar la técnica word2vec para el análisis de sentimientos aplicado a citas en artículos científicos, buscando responder a cuatro preguntas fundamentales:

1. ¿Qué tan bien funciona word2vec en la clasificación de citas positivas y negativas?
2. ¿Pueden las incrustaciones de palabras específicas del sentimiento mejorar el resultado de la clasificación?
3. ¿Qué tan bien funciona word2vec en la clasificación de citas implícitas?
4. ¿cómo funciona word2vec en la clasificación de citas positivas, negativas y objetivas en comparación con las características hechas a mano?

Para la extracción de las oraciones, el autor emplea un modelo de segmentación proporcionado por LingPipe llamado SentenceModel. El texto que utilizó estaba sin procesar y tenía ruido, por ejemplo, habían oraciones incompletas. Para resolver dicho problema, se decidió eliminar todas las oraciones con menos de 3 palabras.

Se construyeron oraciones incrustadas (sent2vec) basadas en palabras incrustadas. Para ello, se promediaron los vectores de las palabras en una oración para obtener oraciones incrustadas (sent2vec). Es vital en este paso usar la matriz sent2vec:

$$V_{sent2vec}(w) = \frac{1}{n} \sum W_w^{x_i} \quad (1)$$

donde W_w ($w = \langle w_1, x_2, \dots, w_n \rangle$) es la palabra incrustada para la palabra x_i . Los parámetros que fueron usados para entrenar el modelo son los mismos que se usaron en el trabajo de Sadeghian y Sharafat [30].

Para evaluar sent2vec en la detección de sentimientos se hicieron experimentos en tres bases de datos: *i.*) La primera base (basic dataset) se tomó originalmente de ACL Anthology y se encontraba anotada manualmente. Usé todas las oraciones etiquetadas (830 positivas, 280 negativas y 7626 objetivos) para las pruebas. *ii.*) El segundo conjunto de datos (conjunto de datos implícito) se utilizó para evaluar la clasificación de citas implícita, que contiene 200222 excluidas (x), 282 oraciones

anotadas positivas (p), 419 negativas (n) y 2880 objetivas (o). Cada frase que no contiene ninguna mención directa o indirecta de la cita se etiqueta como excluida (x) y *iii.*) dataset-pn, que es un subconjunto de dataset-basic, que contiene 828 citas positivas y 280 negativas.

Los resultados de la clasificación binaria mostraron que word2vec es una herramienta prometedora para distinguir citas positivas y negativas; y que la técnica word2vec tiene el potencial de capturar información de sentimientos en las citas, si embargo, las características hechas a mano tienen un mejor rendimiento.

En Wang & Koopman (2017) [31] se utiliza una supuesto crucial: semántica estadística –o como se conoce en lingüística, hipótesis distribucional–; donde se supone que si dos o más palabras son usadas en el mismo contexto, entonces, tienen significados similares.

El objetivo de [31] es formular una medida de similitud semántica entre artículos científicos para luego aplicar dos algoritmos de clustering enriquecidos con la similitud semántica entre los artículos. De forma puntual, los autores plantean dos preguntas de investigación: *i.*) si es posible reconstruir una representación semántica válida para los artículos de todas las entidades semánticas con las que están asociados e *ii.*) identificar clusters de artículos utilizando métodos estándar basados en dicha representación semántica.

Para ello se extienden las palabras a entidades (como autores, revistas, temas, citas) para que cada entidad sea indexada por un vector en un espacio semántico que refleje su contexto léxico, es decir, su coincidencia con ciertos términos.

Una vez que se generan los vectores para los artículos, el siguiente paso es identificar los clusters. Principalmente los autores experimentaron con K-Means el objetivo era comprobar si tales representaciones semánticas producen grupos sensibles.

También se intentó resolver el problema de agrupación desde el paradigma de redes complejas. Como un proceso adicional de tales representaciones semánticas, se transformó las similitudes calculadas en base a tales representaciones vectoriales a una red de similitud de artículos de los cuales se podrían detectar comunidades. Se aplicó el método de detección de comunidades de Louvain para comprobar si el método de Louvain podía encontrar comunidades basadas en similitudes semánticas de los artículos, en lugar de citas entre ellos.

Los resultados de los experimentos muestran que la integración de la semántica de las entidades individuales refleja la semántica de los artículos y los resultados del agrupamiento son comparables con otras soluciones de agrupamiento que se basan principalmente en la información de citas.

En [32] se propone añadir información cualitativa –por medio de NLP– a la bibliométrica de los artículos científicos, analizando el texto que acompaña a la referencia nombrada en el documento.

Además, proponen métodos supervisados para detectar el texto circuncidante de la citación y determinar el propósito (intención del autor) y polaridad (sentimiento de la citación) de la citación. En el caso del propósito, los autores usan las mismas categorías de [26].

Para la polaridad usan un modelo de clasificación basado en 11 características para determinar el sentimiento y propósito asociado con la cita:

- **Conteo de referencias:** Número de referencias en el contexto de la cita.
- **Separado:** Si la citación aparece sola o acompañada de otras referencias.
- **Adverbio / adjetivo / verbo más cercano:** La forma lematizada del verbo / adjetivo / adverbio más cercano a la referencia objetivo o su representante o cualquier mención de este. La distancia es una medida basada en la ruta más corta en el árbol de dependencia.
- **Auto citación:** Si la cita es a otro artículo del mismo autor.
- **Pronombres en primera o tercera persona:** Si la citación contiene pronombres en primera o tercera persona.
- **Negación:** Si el contexto de la cita contiene una señal de negación.
- **Especulación:** Si el contexto de la cita contiene una señal de especulación.
- **Señal de subjetividad más cercana:** La señal de subjetividad más cercana a la referencia de destino o su representante o cualquier mención anafórica de ella.
- **Expresiones contrarias:** Si el contexto de la cita contiene una expresión contraria.
- **Sección:** Sección del documento de texto donde aparece la cita.

- **Relaciones de dependencia:** Todas las relaciones de dependencia que aparecen en el contexto de la cita.

Los experimentos demostraron clasificar el propósito y la polaridad de la cita con una buena precisión. También que el uso del contexto de citas mejora la precisión de la clasificación y aumenta el número de citas polarizadas detectadas.

En [33] se trata de averiguar si los términos de referencia de una cita se pueden representar como temas derivados de su contextos o no.

Para realizar el modelado de temas, los autores usaron la biblioteca de *gensim* en Python y usaron el algoritmo LDA. Como el algoritmo LDA –con los mismos parámetros y corpus– genera diferentes temas en cada ejecución, se fijó en un valor alto el número de iteraciones para estabilizar la generación de temas. Para proporcionar el número óptimo de temas, sin embargo, emplearon el método elbow con el algoritmo K-means. Los resultados muestran que los artículos muy citados son más precisos en la detección del tema que los artículos poco citados.

En [34] buscan aportar un nuevo valor a las publicaciones científicas mediante extracción automática y análisis semántico. Para ello, utilizan una categorización de marcadores lingüísticos de superficie organizados en ontologías lingüísticas. Las anotaciones semánticas se utilizan para enriquecer los metadatos del documento y para proporcionar nuevos tipos de visualizaciones en un contexto de recuperación de información.

Hassan y Haddaway [35] están interesados en analizar el flujo de conocimiento entre países usando la información contenida en las publicaciones científicas y las referencias citadas.

Usando los metadatos de Scopus toman un conjunto de campos de estudio y de artículos vinculados a esos campos y ordenan la localización geográfica de los campos por frecuencia, es decir, si en el campo *energía* hay más artículos de Estados Unidos que de otros países, se asume, entonces, que ese campo tiene su fuente de origen en Estados Unidos.

Luego, usan un modelo LDA para clasificar los artículos en temas y construyen una red de citas y analizan las citaciones del exterior hacia el país fuente de acuerdo a los temas clasificados.

En [36] también están interesados en medir el flujo de conocimiento entre países

usando publicaciones científicas y citas. Consideran, sin embargo, que no todas las citaciones son igual de importantes y, por lo tanto, no toda citación debe considerarse como parte del conocimiento que fluye de un país a otro.

Para clasificar las citas entre “importantes” y “no importantes” los autores usan dos modelos de machine learning: SVM y Random Forest; los cuales recibieron como input 14 variables para entrenar los modelos en un conjunto de datos anotado.

Los autores clasifican en 3 grupos las variables usadas:

- Características basadas en el contexto
 - Número total de citas para una referencia
 - Total de citas cuenta desde el documento actual hasta el documento citado.
 - Recuento total de citas en la sección de introducción
 - Recuento total de citas en la sección de revisión de literatura.
 - Recuento total de citas en la sección del método
 - Recuento total de citas en la sección de experimentos
 - Número total de citas en la sección de discusión.
 - Recuento total de citas en la sección de conclusión.
- Características basadas en palabras claves
 - Palabras de referencia para citas comparativas
 - Palabras de referencia para de trabajos relacionados citados
 - Palabras clave por utilizar el trabajo existente
 - Palabras clave para ampliar el trabajo existente
- Características basadas en texto
 - Cálculo de similitud entre el resumen del documento citado y el texto del artículo citado utilizando la medida de similitud de coseno.
 - Superposición del nombre del autor, si y solo si el documento citado y el documento que cita tienen al menos un autor común.

3.4. Enfoque híbrido

3.4.1. Enfoque híbrido estadístico

Este enfoque hace uso tanto del enfoque sintáctico como del semántico. La idea de usar ambos enfoques subyace en la creencia de que las referencias nombradas en documento de texto científico deben ser enriquecidas con toda la información disponible que nos pueda entregar el documento y no solo con una parte de ella.

En [37] se busca conocer las características de los artículos que son muy citados de acuerdo a la sección donde más se les menciona y a dos características del contexto: n -gramas y ventana de oraciones. En la ventana de oraciones se extrae la oración donde aparece la referencia nombrada más n oraciones antes y/o después de la referencia [38].

Posterior a ello, los autores realizan un análisis de sentimientos para ambas metodologías de extracción de contexto. Sus hallazgos, para $n = 5$ en la ventana de oraciones y n -gramas, muestran que un n grande no permite definir claramente el sentimiento asociado a la cita; además, que la definición del contexto es difusa al ser el n muy grande.

An et al. [39] parte de una idea clave: aprovechar la cita en sí misma como un elemento con semántica, que contiene un significado que codifica la perspectiva del autor que cita, y que proporciona un punto de vista parcial de la estructura intelectual.

Para ello, en [39] aplican técnicas avanzadas de minería de texto para calcular la similitud semántica y realizar el análisis de ubicación de citas, como el de contenido de citas.

Concluyen que los autores altamente citados se distribuyen de manera desigual y muestran una variación considerable en su rango de citas. También encuentran que las citas usualmente ocurren al comienzo de los textos, con el 37 % de las citas en la introducción. Sin embargo, la sección método es donde se cita con mayor frecuencia.

Además, para los autores más citados aparecen con mayor frecuencia en las secciones de método y evaluación. Además, la sección que se cita con más frecuencia varía a medida que consideramos un número mayor de autores citados en primer lugar.

En [40] se propone un *framework* para el análisis sintáctico y semántico del contenido de citas con la intención de que se pueda utilizar para analizar mejor el contexto

sociocultural del las comunidades científicas.

Bertin y Atanassova [41] proponen un método híbrido para la extracción y caracterización de citas en artículos científicos utilizando machine learning junto con enfoques basados en reglas. Su protocolo consiste en la extracción de metadatos, el análisis de bibliografía, el procesamiento de títulos de sección y la anotación semántica en el nivel de la oración de los textos.

En [42] tienen como objetivo proporcionar un nuevo método para calificar un trabajo de investigación basado en la calidad de sus citas. Para encontrar la calidad de la cita utilizan tres procesos semánticos diferentes, pero relacionados: *i.*) clasificación de la cita; *ii.*) análisis del sentimiento de la cita, *iii.*) relevancia del contenido.

Para extraer el puntaje de calidad de las citas se presenta como sigue:

- Hacer seguimiento de las actualizaciones en citación en tiempo real de los artículos de citas por medio de Google scholar.
- Extraer el contexto de las citas.
- Identificar la opinión del artículo que permite identificar si el autor dice cosas positivas o negativas sobre el artículo.
- Clasificar las citas en diferentes categorías para identificar la motivación del autor sobre el artículo citado.
- Identificar la relevancia de la cita por la similitud del coseno y detectar valores atípicos.
- Agregue el puntaje de citación que se obtuvo de los pasos anteriores en uno.

Para ello, aplican e implementan un sistema de machine learning supervisado basado en CRF para el análisis de citas y *sentiwordnet* para el análisis de sentimientos.

En [43] se presenta un método que genera de manera eficiente versiones JSON de documentos PDF que permiten un reconocimiento más fácil de las referencias de los documentos, referencias en el texto, contexto de citas, etc.

Para probar el método, Parinov analiza un conjunto extendido de datos de citas, incluyendo la ubicación de citas en la estructura del texto, la frecuencia de mención de las mismas, el estilo de referencia de referencia.

3.4.2. Enfoque basado en Deep Learning

El uso de modelos de aprendizaje profundo en el análisis de citas científicas ha emergido como una alternativa frente a los métodos tradicionales basados en reglas o conteo. Estos modelos, como las redes neuronales convolucionales (CNN), las redes neuronales recurrentes (RNN) y las redes con memoria a largo plazo (LSTM) permiten capturar relaciones complejas entre el texto que rodea a la cita y su función semántica o importancia relativa dentro del documento [44].

Iqbal et al. [44] realizan una revisión sistemática de los métodos y técnicas de Procesamiento de Lenguaje Natural (NLP) y Machine Learning (ML) aplicadas al análisis de contenido y contexto de citas en textos científicos. Su estudio revisa el desarrollo tecnológico y sus aplicaciones en cinco áreas principales:

1. Análisis de contenido y contexto de citas
2. Clasificación de citas
3. Análisis de sentimientos basado en citas
4. Sumarización basada en citas
5. Sistemas de recomendación de citas

Los autores encuentran las aplicaciones más comunes de los distintos modelos de Redes Neuronales en análisis de citas en documentos científicos, sus limitaciones, ventajas y aplicaciones (ver tabla 2).

Por un lado, entre las aplicaciones con ANN han servido como primer paso para sustituir métodos basados en TF-IDF. En [45] se propone NEAL, un enfoque “neurally enhanced”, el cual utiliza un perceptrón de capa única para vincular de forma automática citas y referencias en tareas de summarización de artículos.

Así, NEAL se sugiere como una alternativa para sustituir métodos basados en TF-IDF para vincular automáticamente citas con sus referencias en tareas de summarización de papers.

De forma puntual, Nomoto [45] trabaja con un subconjunto de artículos de la ACL Anthology Network. Entre sus resultados reporta una mejora en F1 ($\approx 0,76$) comparado con un TF-IDF puro ($\approx 0,62$) en la tarea de identificación de *spans*³

³En [45], un span se refiere a un fragmento contiguo de texto (una subsecuencia de tokens) que contiene la mención de la cita junto con su contexto inmediato. Es decir, para cada cita en un

Modelo	Descripción	Características clave	Aplicaciones destacadas	Limitaciones / Ventajas
ANN	Simula procesos interconectados inspirados en el cerebro.	- Single-layer para scoring - Sin memoria secuencial - Representación jerárquica limitada	- Extracción de spans - Clasificación de relevancia de citas	- Simple y rápido - No distingue bien targets, no retiene contexto
CNN	Redes convolucionales para patrones locales de imágenes o texto.	- Capas convolucionales + pooling - Aprende jerarquías - Detecta n-gramas	- NLP: análisis de sentimientos y polaridad en citas	- Buen rendimiento vs TF-IDF - No resume secuencias, requiere mucho dato
RNN	Feed-forward con bucles para retener contexto breve.	- Memoria a corto plazo - Procesa datos en secuencia variable	- Modelado de dependencias temporales - Recomendación de citas	- Captura bajo temporal - Gradiente desaparecimiento no paralelizable
LSTM	RNN extendido con celdas y tres puertas.	- Manejo eficaz de gradientes largos - Control selectivo de flujo	- Clasificación de citas críticas - Análisis de dependencias	- Retiene largo plazo: supera RNN - Coste computacional y de datos alto

Cuadro 2: Comparación de diferentes modelos de redes neuronales

relevantes, sin embargo, el autor no especifica precisión o recall por separado.

En Yin et al. [1] se busca mejorar la recomendación automática de citas incorporando información del autor en la capa fully-connected de una red neuronal convolucional (CNN).

Tradicionalmente, los sistemas de recomendación de citas basados en CNN trataban exclusivamente el contexto textual alrededor del marcador de cita (por ejemplo, las

artículo, Nomoto extrae el “span” de texto que rodea a esa cita (por ejemplo, la frase o conjunto de frases donde aparece el marcador de cita). Ese span se usa como unidad de entrada al modelo (perceptrón de capa única) para aprender a vincular automáticamente la mención de la cita con su referencia correspondiente.

palabras inmediatas que lo rodean). Los autores proponen entonces una *personalized CNN* (*p-CNN*), que agrega, en paralelo al vector de características extraído por las convoluciones, un vector de autoría para cada cita.

De esta manera, durante el entrenamiento discriminativo, la red puede ponderar no solo la similitud semántica del texto de cita sino también la afinidad del autor con las obras propuestas, mejorando la precisión de la recomendación.

Para entrenar y evaluar el modelo p-CNN, los autores usan tres corpus estándar:

- **AAN (ACL Anthology Network).** Con aproximadamente 10037 artículos y 105823 pares cita-referencia.
- **DBLP.** Con alrededor de 46874 artículos y 615305 pares cita-referencia.
- **RefSeer.** Cerca de 216980 artículos con 4214783 pares cita-referencia (limitado al título y resumen para escalabilidad).

El entrenamiento se realiza con particiones temporalmente separadas (pre-2012 para entrenamiento, post-2012 para prueba). Utilizan “loss margin” con función hinge y optimización SGD. La arquitectura básica consta de:

1. **Capa de embedding de palabras:** vectores de dimensión 150 preentrenados en Wikipedia.
2. **Capa convolucional:** múltiples filtros (3, 4 y 5 tokens) para extraer patrones de n-gramas, seguidos de pooling máximo.
3. **Vector de autoría:** embedding de autor (dimensión 50) para cada autor del artículo candidato.
4. **Capa fully connected:** concatena el vector resultante de la convolución con el vector de autoría y pasa por una capa oculta de 100 unidades antes de la capa de salida.

El cuadro 3 muestra los resultados reportados en [1], donde p-CNN es la CNN propuesta por los autores, NCN corresponde al baseline sin autor y NPM corresponde con un modelo probabilístico neuronal⁴.

En todos los casos, p-CNN supera consistentemente a los baselines: el incremento promedio ronda +5 % en MAP y +7 % en MRR respecto a NCN. Esto demuestra

⁴Neural Probabilistic Model

Conjunto	Modelo	MAP	MRR
AAN	p-CNN	0.187	0.279
	NCN	0.172	0.261
	NPM	0.179	0.268
DBLP	p-CNN	0.184	0.274
	NCN	0.168	0.255
	NPM	0.176	0.266
RefSeer	p-CNN	0.202	0.210
	NCN	0.185	0.194
	NPM	0.190	0.203

Cuadro 3: Resultados de p-CNN, NCN y NPM en métricas MAP y MRR (top-10) reportados en [1]

que añadir el embedding de autor ayuda a filtrar candidatos de cita que, aunque parezcan contextualmente similares, provienen de autores menos relacionados con el documento actual.

Por su parte, en [46] proponen POLAR, un modelo de red neuronal convolucional (CNN) basado en atención combinado con aprendizaje *one-shot* para la personalización en recomendación de artículos.

Dada una consulta, POLAR utiliza una CNN con mecanismo de atención para estimar la puntuación de relevancia entre la consulta y los artículos relacionados. Los autores evalúan el modelo en tres datasets distintos, diseñando experimentos específicos para cada caso:

■ **AMiner:**

- **Contenido:** 188 artículos académicos como consultas, con 10 candidatos cada uno.
- **Protocolo:** Etiquetado manual por expertos (relevancia binaria).
- **Entrenamiento:** Modelo preentrenado en predicción de citas usando el Citation Network Dataset de AMiner.

■ **Patent:**

- **Contenido:** 67 patentes de USPTO como consultas, con 20 candidatos cada una.

- **Reto:** Vocabulario técnico y no estandarizado en los documentos.
 - **Preprocesamiento:** Truncamiento de textos mayores a 96 términos para garantizar entradas de longitud uniforme.
- **RARD:**
- **Escala:** 63 923 consultas con feedback real de clicks (aprox. 9.1 artículos mostrados por consulta).
 - **Partición:** 800 consultas con más clicks reservadas para test, resto para entrenamiento.
 - **Adaptación:** Uso de títulos y abstracts (truncados a 64 términos) para construir las entradas.

De acuerdo a lo reportado por los autores, el rendimiento de POLAR (Tabla 4) superó a los baselines en los tres datasets. En AMiner, logra un +5.0 % en NDCG@3 gracias al mecanismo de atención, mientras que en RARD (con feedback escaso) el módulo de *one-shot learning* aporta un +2.7 % adicional.

Cuadro 4: Resultados de Du et al. (2018) en diferentes datasets

Dataset	Métrica	CNN base	CNN + Atención	POLAR
AMiner	NDCG@3	0.412	0.433	0.445
Patent	MAP@10	0.156	—	0.161
Patent	Recall@10	0.212	—	0.223
RARD	Recall@5	0.098	—	0.102
RARD	MAP@5	0.072	—	0.075

Finalmente, de acuerdo al reporte de los autores, POLAR demostró una gran flexibilidad operando de manera eficiente tanto en conjuntos de datos pequeños, como el de patentes (Patent), como en escenarios a gran escala, como RARD. Además, presenta una notable eficacia en situaciones de “cold-start”: con apenas 1.6 interacciones promedio por usuario en RARD, logra una mejora del 7 % en NDCG@1. Finalmente, su despliegue real en AMiner confirma su aplicabilidad práctica al validar la utilidad del modelo en un entorno de producción.

Karim et al. [47] proponen una metodología para el análisis de sentimiento en citas científicas, combinando técnicas de word embedding (fastText, fastText subword y GloVe) junto con un modelo personalizado de red neuronal convolucional (CNN).

El objetivo de su estudio es sobrepasar las limitaciones de los esquemas tradicionales que solo consideran el conteo de citas, incorporando el análisis cualitativo de la polaridad de las citas (positiva, negativa, neutral) como métrica de impacto más refinada.

Los autores emplearon dos conjuntos de datos:

- **Citation Sentiment Corpus:** 8,736 citas in-text de 194 artículos, anotadas manualmente con los valores “p” (positivo), “n” (negativo) y “o” (objetivo/-neutro).
- **Clinical Trials Dataset:** 4,182 citas de 285 artículos de ensayos clínicos, también anotadas manualmente por polaridad.

El cuadro 5 muestra la distribución de clases en cada dataset, donde se evidencia una mayor proporción de anotaciones neutrales.

Cuadro 5: Distribución de clases en los conjuntos de datos utilizados por Karim et al. [2].

Dataset	Total	Pos.	Neg.	Neutro
Citation Corpus	8,736	829	280	7,627
Clinical Trials	4,182	702	308	3,172

Para la tarea de clasificación, los autores compararon modelos de *machine learning* y *deep learning* utilizando diferentes técnicas de embedding por separado y combinadas. Los principales resultados sobre el *Citation Sentiment Corpus* se muestran en el cuadro 6.

Cuadro 6: Resultados de precisión (accuracy) y F1-score obtenidos por diferentes modelos en el análisis de sentimiento de citas (basado en Karim et al. [2]).

Modelo	Embedding	Accuracy	F1
CNN	fastText	0.89	0.86
CNN	fastText subword	0.87	0.83
CNN	GloVe	0.86	0.88
CNN	Combinado	0.93	0.95
LSTM	Combinado	0.89	0.92
RF	Combinado	0.88	0.90
Log. Reg.	Combinado	0.91	0.89

Según los resultados reportados por Karim et al. [2], la arquitectura CNN combinada con diferentes técnicas de embedding supera a los modelos tradicionales y otros

enfoques de *deep learning* en la tarea de análisis de sentimiento de citas científicas, tanto en precisión como en F1-score, utilizando los datos disponibles y métricas estándar de evaluación.

Además de los experimentos realizados en el *Citation Sentiment Corpus*, Karim et al. [2] validaron su enfoque en un segundo conjunto de datos de ensayos clínicos, donde el modelo CNN con embeddings combinados obtuvo una precisión de 0.92.

Los autores también analizaron la distribución de sentimientos en artículos altamente citados, encontrando que la mayoría de las citas en ambos conjuntos de datos son de tipo neutral, seguidas por un menor número de citas positivas y, en menor proporción, negativas.

Finalmente, los autores afirman que el conteo de la cantidad de citas no refleja necesariamente la valoración de la comunidad científica de un artículo, debido a que muchas de las citas no expresan una opinión explícita sobre el trabajo citado.

Yang et al. [3] abordan el problema de la recomendación de citas contextuales personalizadas, proponiendo un modelo basado en LSTM que integra no solo la similitud semántica entre el contexto de citación y los artículos científicos, sino también la información de autor y *venue*⁵. El objetivo es mejorar la precisión de la recomendación de referencias relevantes considerando factores tradicionalmente ignorados, como la autoría y el lugar de publicación, además del contenido textual.

Para la evaluación experimental los autores utilizaron dos datasets de referencia:

- **ACL Anthology Network (AAN)**: 12,555 artículos tras depuración; papers publicados antes de 2013 como entrenamiento (11,197) y posteriores como test (1,358).
- **DBLP**: 57,441 artículos en total; papers antes de 2014 para entrenamiento (50,273) y 2014-2015 para test (7,168). Incluye subáreas de recuperación de información, machine learning, visión computacional, seguridad informática y redes/comunicación.

El modelo propuesto por los autores utiliza una arquitectura donde:

- CNN y CBOW generan embeddings de sentencias para artículos y contextos de cita.

⁵En el contexto de [3], *venue* se refiere al lugar de publicación de un artículo científico, es decir, la conferencia o revista en la que el trabajo fue publicado.

- Un encoder LSTM integra información de autores y venue en la representación distribuida de cada paper.
- Un encoder Bi-LSTM representa el contexto de cita (izquierda y derecha del placeholder) y, para la versión personalizada, incluye el historial de publicaciones del autor del contexto.

Ellos comparan cuatro variantes del modelo LSTM para la recomendación:

- **LSTM-C**: solo información de contenido.
- **LSTM-CA**: contenido y autores.
- **LSTM-CV**: contenido y venue.
- **LSTM-CAV**: contenido, autores y venue.

Además, se reportan resultados para los enfoques personalizados y no personalizados, así como la comparación con otros métodos (NPP, TM y NN) presentes en la literatura.

Los resultados obtenidos para la tarea de recomendación se presentan en las siguientes tres tablas. En primer lugar, el cuadro 7 muestra el desempeño de las distintas variantes del modelo LSTM evaluados en los conjuntos de datos AAN y DBLP. El cuadro 8 presenta una comparación entre el enfoque personalizado y el no personalizado en ambos datasets. Por último, el cuadro 9 muestra los resultados obtenidos por los métodos NPP, TM, NN y LSTM-CAV en las métricas Recall@10, MAP y MRR, evaluados en los conjuntos de datos AAN y DBLP.

Cuadro 7: Resultados de [3]: variantes LSTM en recomendación contextual (top-10)

Modelo	AAN			DBLP		
	Recall@10	MAP	MRR	Recall@10	MAP	MRR
LSTM-C	0.2081	0.1328	0.2057	0.1482	0.0842	0.1373
LSTM-CA	0.2138	0.1362	0.2106	0.1532	0.0874	0.1420
LSTM-CV	0.2117	0.1346	0.2090	0.1503	0.0857	0.1399
LSTM-CAV	0.2183	0.1401	0.2154	0.1575	0.0908	0.1471

De acuerdo a los valores reportados por los autores en el cuadro 7, la variante LSTM-CAV –que incluye información de contenido, autores y venue– logra valores más altos de Recall@10, MAP y MRR en ambos datasets. Por otro lado, la versión personalizada obtiene valores superiores en todas las métricas evaluadas en ambos

Cuadro 8: Resultados de [3]: comparación personalizada vs. no personalizada

Método	AAN			DBLP		
	Recall@10	MAP	MRR	Recall@10	MAP	MRR
No personalizada	0.2183	0.1401	0.2154	0.1575	0.0908	0.1471
Personalizada	0.2295	0.1444	0.2216	0.1689	0.0949	0.1547

Cuadro 9: Resultados de [3]: comparación con otros métodos (top-10)

Modelo	AAN			DBLP		
	Recall@10	MAP	MRR	Recall@10	MAP	MRR
NPP	0.1769	0.1074	0.1765	0.1163	0.0616	0.1084
TM	0.1864	0.1147	0.1830	0.1283	0.0693	0.1215
NN	0.2075	0.1314	0.2041	0.1486	0.0837	0.1358
LSTM-CAV	0.2183	0.1401	0.2154	0.1575	0.0908	0.1471

conjuntos de datos (ver cuadro 8). Finalmente, al comparar el método propuesto por los autores contra otros métodos (ver cuadro 9), el método LSTM-CAV presentó valores más altos en todas las métricas y para ambos conjuntos de datos.

En [4] proponen el modelo ASL⁶ para recomendación de citas científicas y extracción automática del fragmento de contexto relevante en textos largos. El modelo integra mecanismos de atención global y local para representar tanto el documento citado como el contexto de citación.

Los autores utilizaron dos conjuntos de datos para testear su modelo:

- **AAN (ACL Anthology Network)**: contiene artículos de procesamiento de lenguaje natural, incluye 10,161 artículos y 107,931 pares contexto-citación, con textos extraídos de la ACL Anthology y enlaces de citas mapeados automáticamente. El contexto de cita se define como el fragmento que rodea el marcador de cita en el texto del artículo.
- **DBLP**: abarca literatura en ciencias de la computación, incluyendo artículos de diversas áreas. El subconjunto usado para extracción de contexto incluye 47,963 artículos y 628,253 pares contexto-citación, con contextos construidos a partir de ventanas de texto alrededor de las citas.

La evaluación de la extracción de contexto utilizó las métricas Max Coverage Precision

⁶Attentive Stacked Denoising Autoencoder with Bi-LSTM

(MCP), Max Coverage Recall (MCR) y Positive/Negative ratio (PN) sobre el top 1 de fragmentos predichos, comparando los métodos Tri-LM, TM, CNN, RNN, CRNN y ASL. Los resultados para ambos datasets se muestran a continuación.

Cuadro 10: Resultados de extracción de contexto de cita en AAN (Dai et al., 2020)

Método	MCP@1	MCR@1	PN@1
Tri-LM	0.5051	0.4502	0.5903
TM	0.5710	0.5355	0.7060
CNN	0.5982	0.5471	0.7392
RNN	0.6163	0.5610	0.7630
CRNN	0.6250	0.5748	0.7797
ASL	0.6969	0.6233	0.8689

Cuadro 11: Resultados de extracción de contexto de cita en DBLP [4]

Método	MCP@1	MCR@1	PN@1
Tri-LM	0.4995	0.4237	0.5827
TM	0.5640	0.5148	0.6974
CNN	0.5901	0.5387	0.7284
RNN	0.6047	0.5509	0.7497
CRNN	0.6147	0.5613	0.7657
ASL	0.6739	0.6048	0.8370

Según los resultados reportados, ASL alcanza los valores más altos en las métricas evaluadas para extracción de contexto en ambos conjuntos de datos. Los autores describen que la definición del contexto en cada par contexto-citación corresponde a fragmentos de texto seleccionados automáticamente alrededor de cada cita en los artículos fuente, y que el rendimiento se evalúa en función de la coincidencia entre el fragmento extraído por el modelo y el contexto de referencia proporcionado en los datasets.

3.5. Enfoque contextual (LLMs & RAG)

El reciente auge de los grandes modelos de lenguaje⁷ ha transformado la forma en que se abordan tareas complejas de procesamiento de lenguaje natural, incluido el análisis de citas científicas. Estos modelos permiten inferencias sofisticadas a partir del contexto textual, con o sin entrenamiento adicional, gracias a su capacidad de representar conocimiento lingüístico general.

En esta sección presentaremos algunos trabajos recientes enfocados a 3 métodos: ingeniería de prompts, fine-tuning y arquitecturas RAG.

⁷LLMs, por sus siglas en inglés.

Lahiri et al. [48] proponen el enfoque CitePrompt para evaluar la utilidad de la ingeniería de prompts para la clasificación de intención de cita en artículos científicos. Los autores buscan analizar si los LLMs pueden identificar la intención de cita utilizando únicamente prompts en escenarios zero-shot, one-shot y few-shot, y comparar este desempeño con modelos pre-entrenados y fine-tuned.

El diseño experimental incluye prompts manuales, prompts automáticos y análisis de sensibilidad al diseño del prompt sobre GPT-3 (text-davinci-003), SciBERT, BERT, RoBERTa, BioBERT y DeBERTa en dos datasets:

- **ACL-ARC:** Corpus de 1,941 artículos científicos de NLP (ACL Anthology Reference Corpus) donde cada cita se encuentra anotada manualmente con intención: background, method, result, objective, extend, etc.
- **SciCite:** Enfocado en dominios de informática y biomédico. con más de 11,000 citas anotadas, extraídas de 31,000 artículos científicos de diferentes disciplinas. Cada cita anotada como background, method o result.

El cuadro 12 muestra los resultados reportados por Lahiri et al. [48]. Por un lado, los modelos basados en prompting, tanto en GPT-3 como en SciBERT, lograron macro F1-scores de entre 0.48 y 0.49 en ACL-ARC y entre 0.69 y 0.71 en SciCite, bajo settings zero-shot y one-shot. Por otro lado, los modelos fine-tuned superaron estos valores, alcanzando 0.65 (SciBERT) y 0.61 (BERT) en ACL-ARC, y 0.81 (SciBERT) y 0.77 (BERT) en SciCite.

Cuadro 12: Macro F1-score en clasificación de intención de cita en ACL-ARC y SciCite (Lahiri et al., 2023).

Modelo / Setting	ACL-ARC	SciCite
GPT-3 Zero-shot	0.48	0.69
GPT-3 One-shot	0.49	0.71
SciBERT Zero-shot prompt	0.49	0.70
SciBERT Fine-tuned	0.65	0.81
BERT Fine-tuned	0.61	0.77

Los autores reportan que la diferencia de desempeño entre los datasets se debe en parte a la simplicidad relativa de las clases en SciCite frente a la mayor diversidad y ambigüedad en ACL-ARC.

Además, en [48] señalan que el rendimiento de los modelos basados en prompts depende en gran medida del diseño y la redacción del prompt, así como del número de ejemplos proporcionados (zero-shot, one-shot, few-shot).

Aunque el desempeño del prompting es comparable al de modelos supervisados en el dataset SciCite, es inferior al obtenido por modelos fine-tuned en el dataset ACL-ARC, que presenta mayor complejidad y diversidad de clases.

Por último, los autores resaltan que los resultados pueden no generalizarse fácilmente a otros dominios científicos, y la variabilidad entre datasets limita la robustez del enfoque.

Por otra parte, en [5] usan GPT-3.5-turbo con variantes de prompts (simple, basic, precise, full) para explorar si los LLMs pueden reemplazar a humanos en la anotación de propósito y polaridad de citas, usando para la evaluación un gold standard humano vs accuracy y kappa.

Para este propósito, los autores utilizaron un conjunto de datos compuesto por artículos científicos relacionados con los Objetivos de Desarrollo Sostenible (ODS), específicamente SDG7 (energía asequible y no contaminante) y SDG13 (acción por el clima), que incluía 181 pares de citas anotadas manualmente. Cada cita fue clasificada según:

Propósito de la cita

- Fundamentación (background)
- Comparación (comparison)
- Crítica (criticize)
- Evidencia (evidence)
- Uso metodológico (use)

Sentimiento de la cita

- Positivo
- Negativo
- Neutral

En el cuadro 13 se resumen los principales resultados reportados por [5]. Por un lado, GPT-3.5-turbo obtuvo una precisión de 61.3% y un kappa de 0.77 para la tarea de anotación de propósito de cita, y 64.6% de precisión y 0.72 de kappa para la anotación de sentimiento de cita, utilizando el prompt más detallado (“full”).

El análisis mostró alta consistencia interna en el modelo, pero su precisión general, sin embargo, es inferior en comparación con la referencia humana. Además, la variación en los resultados debida al tipo de prompt fue limitada, lo que sugiere que un mayor detalle en las instrucciones no garantiza una mejora significativa.

Los autores concluyen que, si bien los LLMs pueden ser útiles como herramienta complementaria en la anotación de citas, su desempeño actual no justifica su uso exclusivo en la automatización de esta tarea.

Cuadro 13: Precisión y kappa de GPT-3.5-turbo en anotación de propósito y sentimiento de cita en [5].

Tarea	Accuracy (%)	Kappa
Propósito de cita	61.3	0.77
Sentimiento de cita	64.6	0.72

Por otro lado, en [49] analizan y sistematizan diversas estrategias de entrenamiento para mejorar el rendimiento en la clasificación de intención de citas, utilizando modelos basados en Transformer, con especial énfasis en SciBERT.

Para ello, los autores implementan y comparan diferentes enfoques de fine-tuning, incluyendo entrenamiento clásico, aprendizaje multitarea (multitask learning), pre-entrenamiento adicional con Masked Language Modeling (MLM) y transfer learning con diferentes seeds.

Adicionalmente, los autores evalúan modelos de referencia como BERT, RoBERTa, BioBERT y DeBERTa para contextualizar los resultados.

Para ello, usan dos datasets: ACL-ARC (con 6 clases y 1941 instancias) y SciCite (con 3 clases y 11020 instancias).

Los autores reportan que el fine-tuning multitarea de SciBERT logró el mejor desempeño en ambos datasets (macro F1-score de 0.68 en ACL-ARC y 0.82 en SciCite), superando a variantes single-task y a otros modelos Transformers (ver cuadro 14).

Sin embargo, los autores resaltan que, aunque el fine-tuning mejora el desempeño en comparación con prompting, el método requiere grandes volúmenes de datos etiquetados y puede ser sensible a los splits de entrenamiento y seeds de inicialización.

Por su parte, Gowda et al. [50] analizan el rendimiento de modelos Transformer

Cuadro 14: Macro F1-score en clasificación de intención de cita usando fine-tuning en ACL-ARC y SciCite (Motrichenko et al., 2021).

Modelo	ACL-ARC	SciCite
SciBERT FT (single task)	0.65	0.81
SciBERT FT (multitask)	0.68	0.82
BERT FT	0.61	0.77
RoBERTa FT	0.62	0.79
BioBERT FT	0.60	0.76
DeBERTa FT	0.60	0.76

ajustados mediante fine-tuning (BERT, RoBERTa, DistilBERT) en la clasificación de la intención de citas en textos científicos del dominio de la astrofísica, utilizando como benchmark el dataset FOCAL.

Dicho dataset, se compone de 3,164 contextos de cita, extraídos de 1,024 artículos de astrofísica, anotados manualmente en seis clases de intención: background, basis, comparison, extension, future, method. Los autores utilizaron 2,408 para entrenamiento, 378 validación y 378 test.

Según los resultados obtenidos (ver cuadro 15), BERT-base obtuvo la mayor precisión (0.821) y macro F1 (0.707) en la clasificación de intención de cita en el conjunto de test, seguido por RoBERTa-base y DistilBERT.

Cuadro 15: Resultados de fine-tuning de modelos Transformer en el dataset FOCAL (Gowda et al., 2024).

Modelo	Accuracy	Macro F1	Weighted F1
BERT-base	0.821	0.707	0.818
RoBERTa-base	0.814	0.700	0.810
DistilBERT	0.809	0.698	0.804

Los autores destacan que el desempeño tuvo poca variación entre modelos y que la diferencia principal proviene de la cantidad y calidad del contexto textual disponible alrededor de la cita. Señalan también que los errores más comunes se presentan en clases minoritarias y aquellas con mayor ambigüedad.

Finalmente, en [51] se compara de manera sistemática dos estrategias para incrustar información externa en LLMs: *i.*) expandir la ventana de contexto (long context), y *ii.*)

emplear mecanismos RAG. Para ambos enfoques, los autores evaluaron el desempeño en tareas de pregunta-respuesta, generación y razonamiento. Los experimentos incluyen análisis de eficiencia, calidad de respuesta, precisión y “factuality”.

En el caso RAG, los autores implementan y comparan distintas variantes de RAG con diferentes métodos de recuperación: chunk-based, index-based, summarization-based; y los contrastan contra modelos LLM con context window extendida (por ejemplo, GPT-4-turbo, Claude, Gemini Ultra).

Se usan 4 datasets (ver cuadro 16) con el propósito de evaluar y comparar las capacidades de los LLM bajo enfoques de contexto largo –long context– y RAG en escenarios de pregunta-respuesta abiertos y con evidencia dispersa.

De forma puntual, se utilizó NQ-Open y NaturalQuestions para evaluar la recuperación y generación de respuestas concisas a partir de Wikipedia, HotpotQA para analizar el razonamiento multi-hop y la integración de evidencia de diferentes documentos, y MultiDoc2Dial para simular escenarios realistas donde el sistema debe navegar y seleccionar respuestas dentro de grandes volúmenes de texto de naturaleza conversacional o documental.

Cuadro 16: Resumen de los datasets utilizados en Li et al. (2024).

Dataset	Descripción
NQ-Open	Preguntas abiertas con respuestas extraídas de Wikipedia; evaluación de recuperación y síntesis multi-documento.
HotpotQA	QA multi-hop, requiere combinar información de varios párrafos y documentos para responder correctamente.
MultiDoc2Dial	QA y diálogo sobre colecciones de documentos extensos; simula escenarios de atención al cliente y búsquedas complejas.
NaturalQuestions	Preguntas reales con evidencia dispersa en Wikipedia; evalúa capacidad de recuperación y comprensión en contexto abierto.

De acuerdo a los resultados reportados en [51] (ver cuadro 17) el enfoque RAG con recuperación index-based logra los puntajes más altos en casi todos los datasets evaluados, sin embargo, la diferencia entre métodos tiende a disminuir a medida que mejora la calidad base del LLM.

Generalmente, la eficiencia computacional de RAG es superior cuando se requiere

procesar grandes volúmenes o ventanas contextuales extensas, y además permiten mayor flexibilidad en la integración de información dispersa. No obstante, cabe anotar que la robustez de la respuesta depende de la calidad del mecanismo de recuperación.

Cuadro 17: Comparación de desempeño (F1/Accuracy) entre context window largo y variantes RAG (Li et al., 2024).

Método / Dataset	NQ-Open	HotpotQA	MultiDoc2Dial	NaturalQuestions
Long Context LLM	45.7	68.2	55.3	39.1
RAG (chunk-based)	49.2	70.1	56.7	41.8
RAG (index-based)	50.5	71.3	57.2	42.5
RAG (summarization-based)	48.8	70.5	56.5	41.9

Por su parte, [6] introduce GRAG, un enfoque que extiende la arquitectura RAG a documentos en red (como grafos de citas o conocimiento). Su objetivo es mejorar el rendimiento en tareas que requieren razonamiento multi-hop, induciendo que los LLMs utilicen tanto el contexto estructural como el textual de subgrafos relevantes, y no solo la información textual de documentos individuales.

GRAG utiliza una estrategia de divide-and-conquer para recuperar subgrafos textuales relevantes (“ego-graphs”) y los incorpora al LLM como:

- hard prompts (descripciones textuales jerárquicas) y,
- soft prompts (embeddings topológicos generados por GNNs)

El modelo base que se usa en [6] es Llama-2-7b, con y sin fine-tuning mediante LoRA. Para evaluar la propuesta GRAG, los autores emplean los conjuntos de datos WebQSP y ExplaGraphs (ver cuadro 18)

Cuadro 18: Descripción de los conjuntos de datos utilizados en [6].

Dataset	Descripción y uso en el experimento
WebQSP	Benchmark de pregunta-respuesta multi-hop sobre grafos de conocimiento. Se utiliza para evaluar la capacidad del modelo de recuperar y razonar sobre subgrafos relevantes al responder preguntas complejas, a partir de grafos grandes y densos derivados de queries reales.
ExplaGraphs	Dataset de razonamiento commonsense con grafos explicativos pequeños. Se usa para medir la habilidad del modelo en seleccionar y justificar respuestas mediante navegación sobre cadenas de hechos conectados en grafos estructurados y de menor tamaño.

Los autores reportan que GRAG y GRAGLoRA superan en todas las métricas a los métodos RAG tradicionales y a los LLMs fine-tuned, en ambos benchmarks, especialmente en tareas de razonamiento multi-hop sobre grafos textuales (ver cuadro

19). El modelo logra valores de F1, Hit@1 y Recall superiores en WebQSP y mayor accuracy en ExplaGraphs.

Cuadro 19: Desempeño de GRAG y métodos comparados en WebQSP (F1, Hit@1, Recall) y ExplaGraphs (Accuracy).

Modelo	F1	Hit@1	Recall	Accuracy (ExplaGraphs)
LLM only	0.2555	0.4148	0.2920	0.3394
LLMLoRA	0.4295	0.6186	0.4193	0.8927
BM25 (RAG)	0.2999	0.4287	0.2879	0.6011
MiniLM-L12-v2 (RAG)	0.3485	0.4730	0.3289	0.6011
LaBSE (RAG)	0.3280	0.4496	0.3126	0.6011
mContriever-Base (RAG)	0.3172	0.4453	0.3047	0.5866
E5-Base (RAG)	0.3421	0.4705	0.3254	0.6011
G-Retriever (RAG)	0.4674	0.6808	0.4579	0.8825
G-RetrieverLoRA (RAG)	0.5023	0.7016	0.5002	0.9042
GRAG	0.5022	0.7236	0.5099	0.9223
GRAGLoRA	0.5041	0.7275	0.5112	0.9274

Finalmente, los autores señalan que la eficiencia de GRAG depende de la calidad del ranking inicial de nodos y del mecanismo de *pruning*. No obstante, cuando la estructura del grafo –o la importancia de los nodos es difícil de estimar–, el desempeño de la recuperación puede verse afectado. Además, los autores reconocen que la escalabilidad y el costo computacional podrían ser un desafío en grafos muy grandes o poco estructurados.

Por su parte, Xu et al. [10] propone automatizar y mejorar el análisis bibliométrico mediante la integración de Sentence Transformers y un pipeline RAG (Retrieval-Augmented Generation), con el fin de posibilitar búsquedas semánticas y contextuales, ranking temático y caracterización personalizada de literatura científica en el dominio de la investigación urbana.

El workflow propuesto propuesto por los autores consiste en:

1. Extracción de literatura desde bases como Scopus, IEEE Xplore y Web of Science.
2. Generación de embeddings con Sentence Transformers (all-MiniLM-L6-v2, Hugging Face).
3. Almacenamiento de textos y embeddings en una base de datos de grafos (Neo4j), con nodos que representan artículos y propiedades asociadas.

4. Búsqueda semántica y contextual usando similitud de embeddings, chunking y ranking con GMM⁸ para agrupar y priorizar artículos relevantes según la consulta del usuario.
5. RAG: los artículos y fragmentos relevantes recuperados son enviados como contexto, junto con prompts personalizados, a LLMs (ChatGPT-3.5 Turbo y GPT-Neo).
6. Análisis y caracterización: el LLM genera métricas y caracterizaciones personalizadas de los artículos en respuesta a las instrucciones del usuario.

Los autores usaron 223 artículos científicos publicados en Nature Communications (dominio de investigación urbana), recolectados con una consulta específica en Scopus que filtra por términos como “smart city”, “urban”, “urban management” y “urban planning” en títulos, resúmenes y keywords para el periodo 2013–2023.

De acuerdo a los resultados reportados por los autores:

- El sistema filtra y rankea artículos relevantes en función de consultas conversacionales usando búsquedas semánticas y clustering GMM.
- Los artículos seleccionados son enviados como contexto a LLMs, que generan análisis, resúmenes temáticos y métricas bibliométricas personalizadas.
- Los autores demuestran, mediante casos de uso, que el sistema responde a consultas complejas (como “urban green space”) y produce reportes automáticos, histogramas de similitud y agrupaciones temáticas.
- El workflow permite caracterizar literatura científica de manera contextual y detallada, superando las limitaciones de enfoques tradicionales basados solo en palabras clave.
- El sistema permite caracterizar literatura científica de forma más contextual y granular que métodos basados solo en keywords o metadatos, y ofrece nuevas formas de análisis bibliométrico orientado a necesidades del usuario.

Por último, [52] introduce PaperQA2: un sistema de agentes basado en RAG para responder preguntas complejas, sintetizar información y detectar contradicciones en literatura científica.

⁸Gaussian Mixture Model.

En su artículo, buscan evaluar la capacidad de PaperQA2 para recuperar evidencia relevante y generar respuestas citadas utilizando LLMs comerciales y open source; y se compara con con expertos humanos y otros sistemas como Perplexity AI, Elicit y RAG basados en GPT-4.

De forma puntual, PaperQA2 funciona siguiendo siguiente pipeline:

1. **Recepción de consulta:** El agente recibe una pregunta en lenguaje natural.
2. **Búsqueda inicial:** Utiliza motores de búsqueda y estrategias de keyword matching para identificar artículos científicos potencialmente relevantes.
3. **Parsing semántico:** Convierte los PDF a texto estructurado usando Grobid para extraer títulos, secciones y metadatos.
4. **Segmentación (chunking):** Divide el texto de los artículos en fragmentos manejables para recuperación y análisis posterior.
5. **Recuperación vectorial:** Emplea modelos de embeddings para buscar y recuperar fragmentos de texto relevantes respecto a la consulta.
6. **Reranking con LLMs:** Los fragmentos recuperados son rerankeados utilizando un modelo LLM, que evalúa la relevancia y precisión contextual de cada fragmento.
7. **Citation Traversal Tool:** Opcionalmente, expande la búsqueda siguiendo enlaces del grafo de citas para recuperar fragmentos adicionales de artículos citados o que citan al documento inicial.
8. **Generación de respuesta:** Un LLM genera una respuesta a la pregunta del usuario, integrando la información de los fragmentos seleccionados y proporcionando citas automáticas (con links al PDF y ubicaciones exactas).
9. **Síntesis y verificación:** El agente puede realizar verificaciones cruzadas (factual consistency) y, en el caso de tareas específicas, sintetizar resúmenes o detectar contradicciones usando los fragmentos contextuales recuperados.

Para la evaluación de PaperQA2 se emplearon dos datasets: LitQA2 y Contracrow:

- **LitQA2:** Benchmark de 248 preguntas científicas tomadas de Wikipedia, en el que las respuestas requieren recuperar evidencia del cuerpo de artículos completos y no solo de los abstracts. Permite evaluar recuperación de fragmentos,

síntesis y citación automática. Incluye validación con respuestas de expertos humanos (PhD).

- **Contracrow:** Dataset diseñado para la tarea de detección de contradicciones científicas, a partir de prompts basados en Wikipedia y validación por expertos. Utilizado para evaluar la capacidad del sistema para identificar contradicciones en literatura científica.

Para ambos datasets, las respuestas generadas por el sistema fueron comparadas y validadas frente a las respuestas de expertos humanos (PhD) en los dominios correspondientes.

De acuerdo a los resultados reportados por los autores (ver cuadro 20), PaperQA2 logra una precisión del 85.2 % en la tarea de recuperación de evidencia en el benchmark LitQA2, superando el desempeño de expertos humanos (73.8 %) y de otros sistemas RAG evaluados.

En términos de accuracy, el desempeño de PaperQA2 es similar al de expertos humanos. En la tarea de generación de resúmenes científicos con citas, el sistema PaperQA2 reporta un menor porcentaje de errores de citación en comparación con los resúmenes de Wikipedia.

Por último, Para la tarea de detección de contradicciones (Contracrow), el sistema logra un F1 de 0.82 en la identificación de contradicciones científicas validadas por expertos.

Cuadro 20: Desempeño de PaperQA2, humanos y sistemas automáticos en el benchmark LitQA2.

Método	Precisión (%)	Accuracy (%)
Humanos (PhD)	73.8	67.7
PaperQA2	85.2	66.0
Perplexity (RAG)	60.1	52.6
Elicit (RAG)	64.0	53.0
GPT-4 RAG baseline	73.0	57.2

4. Marco Conceptual

En esta sección se presentan los conceptos clave que sustentan el diseño del prototipo para el análisis de contexto de referencias en documentos científicos. De forma puntual, se describen cuatro componentes fundamentales:

1. **LLMs y prompt engineering:** Proporcionan capacidades de comprensión y generación de lenguaje natural contextualizado.
2. **Modelos LLM usados en este documento**
3. **Arquitectura RAG:** Permite la recuperación eficiente de información específica de un dominio determinado.
4. **Sistemas multiagentes:** Orquestación del flujo de trabajo (ver figura 4).
5. **Modelos LLM**

Estos tres componentes ofrecen un marco integral para comprender cómo el sistema procesa, recupera y genera información contextualizada a partir de documentos académicos, aprovechando la complementariedad entre cada uno para crear un sistema robusto y adaptable.

4.1. LLMs y prompt engineering

4.1.1. LLMs

Los LLMs son redes neurales profundas –preentrenadas con grandes cantidades de datos– que están basadas en la arquitectura Transformer [53, 54]. Su funcionamiento se genera a partir de un sistema dual (*encoder - decoder*) que extrae significados de un texto y comprende las relaciones entre las palabras y las frases que contiene [55].

Además, los LLMs han demostrado capacidades emergentes que se extienden más allá de comprender y generar texto [54], mostrando competencia en tareas como razonamiento de sentido común, generación de código y aritmética [56].

4.1.2. Prompt engineering

Aunque los LLMs han demostrado ser muy eficientes en distintos tipos de tareas, la eficiencia y calidad de sus respuestas depende del diseño y formulación de la instrucción (prompt) [5, 57]. Así, el prompt engineering constituye una metodo-

logía sistemática para guiar el comportamiento de los LLMs mediante instrucciones estructuradas en lenguaje natural.

El cuadro 21 resume las cuatro estrategias de prompt engineering más usadas y divulgadas [56].

Cuadro 21: Estrategias de prompting

Estrategia	Descripción y Ejemplo	Ventajas	Limitaciones
Zero-shot prompting [56]	Instrucciones directas sin ejemplos previos. <i>Ejemplo</i> : “Identifica el propósito de esta cita en el contexto dado”. Utiliza capacidades emergentes del modelo para interpretar tareas sin demostración previa.	Simplicidad de implementación, alta generalización, no requiere ejemplos de entrenamiento, aplicación inmediata.	Menor precisión en tareas complejas, dependiente de claridad de instrucciones, resultados variables según modelo.
Few-shot prompting [57]	Incorporación de ejemplos demostrativos para clasificación de intención de citas. Proporciona 1–5 ejemplos representativos antes de la tarea real.	Mejor comprensión de la tarea, mejora consistencia de respuestas, adaptable a dominios específicos, guía formato de salida.	Requiere selección cuidadosa de ejemplos representativos, posible sesgo por ejemplos, consume más tokens.
Chain-of-thought prompting [56]	Fomento de razonamiento explícito paso a paso. <i>Implementación</i> : “Explica paso a paso cómo esta cita contribuye al argumento del paper”. Descompone problemas complejos.	Análisis complejo de contexto, razonamiento transparente y trazable, mejor manejo de relaciones entre referencias, mayor precisión.	Mayor consumo de tokens, posible verbosidad innecesaria, tiempo de procesamiento incrementado.
Role-based prompting [5]	Asignación de expertise específico. <i>Ejemplo</i> : “Como experto en análisis bibliométrico, evalúa el contexto de esta referencia”. Establece perspectiva de dominio.	Respuestas más especializadas y técnicamente precisas, contexto de dominio específico, terminología apropiada.	Efectividad variable según capacidades del modelo, requiere definición clara del rol y contexto.

4.2. Modelos LLM usados en este documento

La selección de modelos de lenguaje grandes (LLMs) para el sistema multi-agente se basó en criterios específicos que aseguran diversidad arquitectónica, variabilidad en capacidades de razonamiento y representatividad de diferentes paradigmas de

entrenamiento [58]. Los criterios establecidos fueron:

1. Diversidad de Proveedores y Filosofías: Incluir modelos de diferentes organizaciones (OpenAI, Google DeepMind, DeepSeek, Mistral AI, Meta) para evaluar enfoques diversos en el desarrollo de LLMs [59].
2. Variabilidad en Tamaños de Parámetros: Abarcar desde modelos compactos (8B) hasta modelos de gran escala (70B) para analizar el trade-off entre eficiencia computacional y capacidad de razonamiento.
3. Técnicas de Entrenamiento Diferenciadas: Incluir modelos que representen distintos enfoques de post-entrenamiento, incluyendo RLHF, DPO, y destilación de conocimiento.
4. Performance en Benchmarks Científicos: Priorizar modelos con rendimiento competitivo en evaluaciones relevantes para tareas de comprensión científica (MMLU, GPQA, HumanEval).
5. Disponibilidad y Reproducibilidad: Asegurar acceso estable a los modelos durante el período de investigación.

De forma puntual, se evaluaron 5 modelos LLM: GPT-4o mini, Gemini-2.0 Flash, DeepSeek R1 Distill Qwen 32B, Mistral 8B 2410 y Llama-3.3 70B

En primer lugar, GPT-4o mini está diseñado con una arquitectura transformer optimizada para baja latencia y técnicas avanzadas de alineación como RLHF y Constitutional AI [18, 20].

Rendimiento en Benchmarks:

- MMLU: 82.0 % [8]
- GPQA: 50.3 % [60]
- HumanEval: 87.2 % [8]
- MATH: 70.2 % [8]

Por otro lado, *Gemini-2.0 Flash* está diseñado con arquitectura de tipo dense transformer que incorpora capacidades nativas multimodales, herramientas integradas, y una ventana de contexto extendida de hasta 1 millón de tokens [61, 62].

Rendimiento en Benchmarks:

- MMLU-Pro: 76.4 % (conocimiento general) [63].
- GPQA Diamond (Thinking variant): 74.2 % (datos de pensamiento) [64].
- MATH/Matemáticas: 89.7 % [63].
- Code Generation (Natural2Code): 92.9 % [65].

En tercer lugar, el modelo *DeepSeek R1 Distill Qwen 32B* es una aproximación destacada de destilación avanzada desde un modelo mayor (DeepSeek R1, con hasta 671B parámetros), logrando transmitir capacidades de razonamiento complejas hacia una versión más compacta [66, 67].

Rendimiento en Benchmarks:

- AIME 2024 (pass@1): 72.6 % [67]
- MATH-500 (pass@1): 94.3 % [67]
- GPQA Diamond: 62.1 % [68]

En cuarto lugar, el modelo Ministral 8B 2410 es un modelo dense transformer, y es hoy por hoy uno de los más destacados en diversas pruebas y benchmarks (como MMLU, HumanEval, GSM8K, tareas multilingües, etc.), superando consistentemente a otros modelos de tamaño similar [69]

Rendimiento en Benchmarks:

- MMLU (Base): 65.0 % [70]
- HumanEval pass@1 (Instruct): 76.8 % [70]
- Math maj@1 (Instruct): 54.5 % [70]

Por último, el modelo Llama-3.3 70B Instruct es un modelo de estado del arte en el ecosistema open-source de gran escala, de tipo autoregressive transformer con Grouped-Query Attention (GQA) [71].

Rendimiento en Benchmarks:

- MMLU (0-shot, CoT): 86.0 % [72]
- GPQA Diamond (0-shot, CoT): 50.5 % [73]
- HumanEval (0-shot): 88.4 % [73]

4.3. Análisis Comparativo de Técnicas de Entrenamiento

4.3.1. Reinforcement Learning from Human Feedback (RLHF)

Utilizado principalmente en *GPT-4o Mini* y *Llama-3.3 70B* [18, 20], el RLHF es una técnica de alineación que mejora la capacidad de los modelos para seguir instrucciones humanas. El proceso típico incluye:

1. **Entrenamiento de un modelo de recompensa:** se entrena a partir de comparaciones humanas para predecir la preferencia entre dos salidas [74].
2. **Optimización mediante PPO:** el modelo principal se ajusta con *Proximal Policy Optimization* utilizando el modelo de recompensa como señal de retroalimentación [75].
3. **Iteración continua:** se realizan múltiples ciclos de ajuste, incorporando nuevo *feedback* humano para refinar el comportamiento del modelo [76].

4.3.2. Direct Preference Optimization (DPO)

Presente en *Llama-3.3 70B* y *Gemini-2.0 Flash* [17], el DPO es una alternativa más directa y eficiente al RLHF tradicional. Sus características clave incluyen:

1. **Optimización directa:** no requiere el entrenamiento de un modelo de recompensa separado, reduciendo la complejidad del proceso.
2. **Mayor estabilidad:** minimiza la inestabilidad común en entornos de entrenamiento basados en RL [16].
3. **Menor coste computacional:** al eliminar etapas intermedias, disminuye el uso de recursos manteniendo una alineación efectiva [77].

4.3.3. Destilación de Conocimiento

Aplicada en *DeepSeek R1 Distill Qwen 32B* [78, 79], esta técnica transfiere el conocimiento y las capacidades de razonamiento de un modelo grande a otro más compacto. El proceso típico involucra:

1. **Generación de datos sintéticos:** el modelo maestro produce ejemplos de razonamiento de alta calidad [66].

2. **Transfer learning:** el modelo estudiante aprende a reproducir el rendimiento del modelo maestro a través de *fine-tuning* supervisado [80].
3. **Preservación de capacidades:** se optimiza para mantener un rendimiento competitivo con una reducción significativa de parámetros [81].

El conjunto de modelos seleccionado representa diversidad arquitectónica, variabilidad de técnicas de entrenamiento y balance entre modelos comerciales y open-source [7, 11, 82].

4.4. Arquitectura RAG

La arquitectura RAG apareció por primera vez en 2020 por Facebook AI (ahora Meta AI), con el fin de mejorar los modelos de lenguaje mediante la integración de recuperación de información en el proceso de generación [83, 84].

Este paradigma intenta resolver las limitaciones de los modelos generativos mediante la recuperación dinámica de información externa al corpus de entrenamiento, lo que permite generar respuestas con datos verificables, reduciendo las alucinaciones y la generación de contenido factualmente incorrecto [83].

La figura 2 muestra la arquitectura fundamental del pipeline RAG. La fase de indexación de datos⁹ (superior) se compone de cuatro etapas:

1. Data Loading (carga de datos)
2. Data Splitting/text chunks (segmentación)
3. Data Embedding/vector embeddings (generación de embeddings)
4. Data Storing (almacenamiento en base de datos vectorial)

En primer lugar, se cargan los documentos, después se segmentan en fragmentos de texto (chunks) de tamaño apropiado para optimizar la recuperación posterior. Estos fragmentos se convierten en representaciones vectoriales mediante modelos de embedding, creando una representación numérica que captura la semántica del contenido. Finalmente, estos vectores se almacenan en una base de datos vectorial que permite búsquedas de similaridad eficientes.

La fase de recuperación y generación¹⁰ (inferior) ilustra el flujo de recuperación de

⁹Data Indexing.

¹⁰Data Retrieval & Generation.

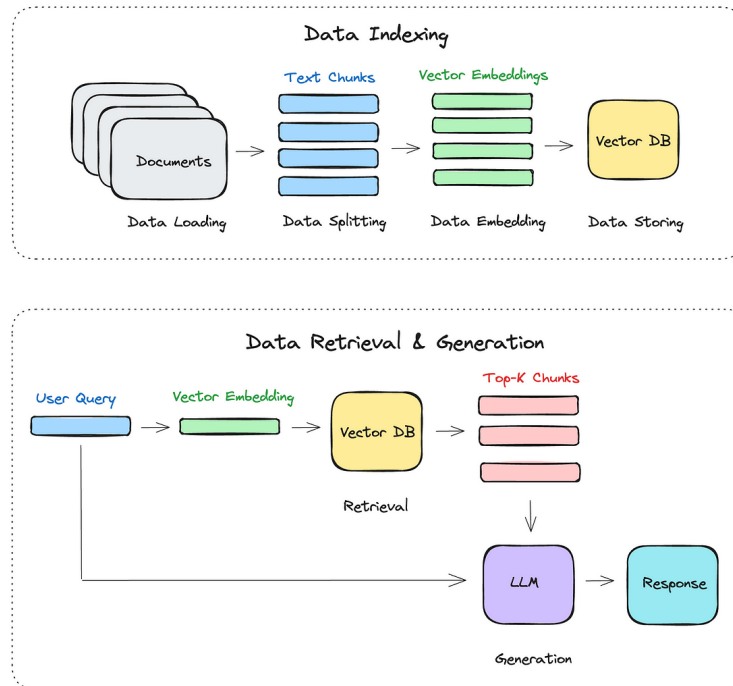


Figura 2: Pipeline RAG. Tomado de: <https://medium.com/@annoliveai/rag-and-the-future-41196993bc9e>

información y generación de respuesta. Cuando el usuario realiza una consulta (User Query), esta se procesa mediante el mismo modelo de embedding utilizado en la indexación para mantener la coherencia vectorial. El vector resultante se utiliza para realizar una búsqueda de similaridad en la base de datos vectorial, recuperando los fragmentos más relevantes (Top-k Chunks). Estos fragmentos seleccionados se integran como contexto adicional que se envía al modelo de lenguaje (LLM), que genera una respuesta fundamentada en la información recuperada.

4.5. Sistemas multi-agentes

Los sistemas multi-agentes son un paradigma computacional donde múltiples entidades autónomas colaboran para realizar una tarea que excede la capacidad de un único agente.

En [85] se definen las características fundamentales que definen la agencia en sistemas computacionales: *i.*) autonomía (capacidad de operar sin intervención externa), *ii.*) reactividad (percepción y respuesta al entorno), *iii.*) proactividad (comportamiento dirigido por objetivos), y *iv.*) habilidad social (interacción con otros agentes y usuarios

humanos).

La ecuación 2 [86] representa la formalización de los sistemas multi-agentes, donde: A representa el conjunto finito de agentes, E el entorno compartido, I las reglas de interacción, y Ω los objetivos globales del sistema.

$$MAS = \langle A, E, I, \Omega \rangle \quad (2)$$

Para la implementación práctica de los sistemas multi-agentes, se han desarrollado frameworks especializados que traducen los conceptos abstractos de agencia y coordinación en herramientas de desarrollo concretas.

Con la aparición de las arquitecturas basadas en LLM se ha dado paso al origen de plataformas de orquestación que simplifican la construcción de sistemas multi-agentes que permiten tener control sobre el comportamiento individual como colectivo de los agentes.

A continuación, se describen algunos de los frameworks de orquestación más representativos:

1. LangGraph: Framework especializado para la construcción de sistemas multi-agente basado en grafos de estados dirigidos. Su arquitectura modela cada agente como un nodo en el grafo, donde las aristas representan el flujo determinista de información y control dentro del sistema
2. CrewAI: Adopta una filosofía de múltiples agentes autónomos organizados en “crews” o equipos especializados. Su enfoque se distingue por la implementación de roles específicos, autonomía delegada e inteligencia colaborativa
3. LlamaIndex: Framework centrado en la orquestación de agentes y la gestión de conocimiento, que combina indexación semántica de fuentes heterogéneas (documentos, bases de datos, APIs) con flujos de trabajo de agentes. Cada agente actúa como un módulo de procesamiento conectado a un índice de conocimiento compartido, lo que permite consultas contextuales, enrutamiento dinámico de tareas y escalabilidad distribuida.

Los sistemas multi-agente basados en LLM funcionan mediante la interacción autónoma y coordinada de agentes heterogéneos que pueden invocar herramientas externas, ejecutar acciones especializadas y compartir información a través de índices de

conocimiento.

5. Metodología

El prototipo propuesto en este documento presenta un enfoque híbrido que combina tres paradigmas computacionales: *prompt engineering*, RAG y Agentes con el fin de abordar el problema específico del análisis de contexto de referencias en literatura científica.

Esta sección se estructura en cinco etapas principales, comenzando por la construcción y caracterización de un corpus de evaluación representativo, seguido del pipeline de procesamiento e integración de componentes (ver figura 3).

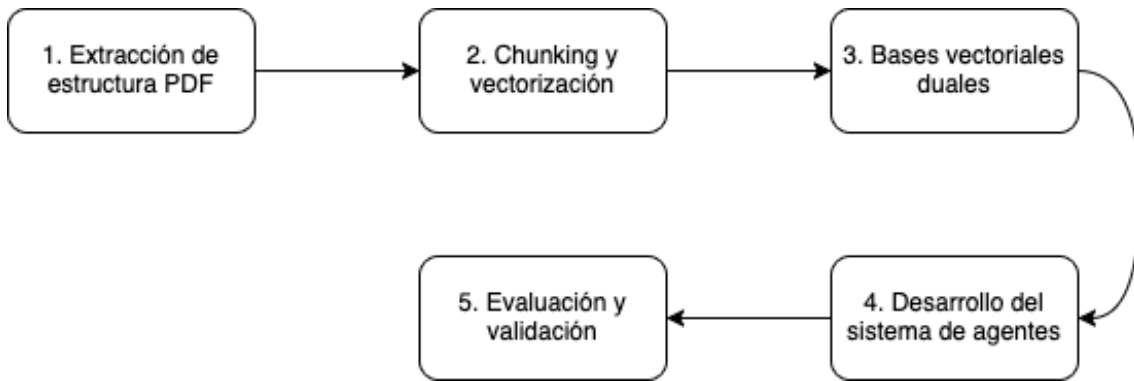


Figura 3: Pipeline metodología propuesta

5.1. Descripción del Dataset

El dataset utilizado en esta investigación consta de 89 artículos científicos seleccionados estratégicamente para evaluar el sistema multi-agente de análisis de contexto de referencias. La recopilación priorizó la diversidad temática, calidad estructural y accesibilidad para el procesamiento automatizado.

Los artículos proceden principalmente de repositorios académicos públicos como arXiv, ACL Anthology y sitios web de conferencias relevantes (ACL, EMNLP, CVPR, NeurIPS). El rango temporal abarca publicaciones desde 1999 hasta 2025, con aproximadamente 85 % de los artículos publicados entre 2015 y 2024.

La distribución de dominios del dataset muestra predominancia de Procesamiento de Lenguaje Natural, seguido por Visión Computacional y Aprendizaje Automático. El cuadro 22 detalla la distribución completa por dominio académico.

El análisis automático de idioma reveló predominancia del inglés (93 artículos,

Cuadro 22: Distribución del dataset por dominio académico

Dominio académico	Artículos	Porcentaje
Procesamiento de Lenguaje Natural	64	74.0 %
Visión Computacional	6	6.2 %
Aprendizaje Automático	4	4.2 %
Ciencias Sociales	3	3.1 %
Inteligencia Artificial	3	3.1 %
Economía	2	2.1 %
Educación	2	2.1 %
Modelado Basado en Agentes	1	1.0 %
Biomedicina	1	1.0 %
Ingeniería de Tráfico	1	1.0 %
Minería de Datos y Texto	1	1.0 %
Aprendizaje Profundo	1	1.0 %
Total	89	100.0 %

96.9 %), con 3 artículos en español (3.1 %). Se identificaron cuatro estilos principales de citación en texto, como muestra el cuadro 23.

Cuadro 23: Distribución de estilos de citación en texto

Estilo de citación	Artículos	Porcentaje
IEEE	55	61.7 %
APA	32	34.8 %
Otros estilos	2	2.2 %
Total	89	100.0 %

Los artículos presentan diversidad estructural significativa. El número de secciones por artículo varía entre 6 y 63, mientras que las referencias bibliográficas oscilan entre 0 y 140. La Tabla 24 resume las estadísticas estructurales principales.

Cuadro 24: Estadísticas estructurales del dataset

Característica	Mínimo	Máximo	Promedio	Total
Secciones por artículo	6	63	15.7	-
Referencias por artículo	15	140	50.3	-
Longitud (caracteres)	15,482	387,205	121,778	-
Chunks ChromaDB	12	127	48.2	4,627
Chunks FAISS	5	58	26.5	2,544

Para habilitar búsquedas semánticas, cada artículo fue procesado en dos formatos de segmentación optimizados para diferentes tipos de consultas: chunks granulares para

ChromaDB (promedio: 48.2 por artículo) y chunks holísticos para FAISS (promedio: 26.5 por artículo).

La selección del corpus se guió por criterios de diversidad temática dentro de ciencias de la computación y áreas afines, calidad estructural con secciones bien definidas, riqueza referencial (más de 10 citas por documento), accesibilidad técnica con PDFs de texto extraíble y relevancia temporal con predominancia de publicaciones recientes.

Si bien el corpus fue diseñado para maximizar la diversidad, se reconocen limitaciones: predominio de artículos en inglés, sobrerrepresentación de Procesamiento de Lenguaje Natural debido a la disponibilidad de artículos bien estructurados, sin embargo, la muestra no pretende ser estadísticamente representativa de toda la literatura científica, sino más bien un conjunto de casos diversos y desafiantes para evaluar las capacidades del sistema.

5.2. Extracción estructura PDF

La extracción de la estructura sintáctica de cada artículo se dividió en tres tareas: *i.*) extracción del texto del pdf, *ii.*) identificación de secciones del texto, *iii.*) extracción de las referencias del texto, *iv.*) segmentación por secciones con metadatos enriquecidos y *v.*) validación de extracción.

En primer lugar, la extracción del texto de los archivos pdf se usó la librería PyMuPDF. Esta librería se usó dado que, de acuerdo a la evidencia disponible, PyMuPDF tiene un rendimiento superior en preservación de estructura de texto y manejo de *layouts complejos* [14]. Además, el reporte de algunos benchmarks, PyMuPDF es superior en velocidad de procesamiento frente a otras alternativas [87].

En segundo lugar, la identificación de las secciones y extracción de referencias presentes en el texto se implementa por medio de *prompt engineering* con Gemini—2.0—flash. La tabla 25 reúne los resultados de varios estudios experimentales donde se evaluaban los métodos tradicionales (Regex) versus LLMS en tareas de extracción de documentos de texto estructurados.

Los LLMs destacan frente a los métodos basados en expresiones regulares en adaptabilidad, entendimiento del contexto, robustez ante cambios de la estructura del texto y en requerir menos mantenimiento, dada su capacidad de auto-adaptación a distintas estructuras de texto.

Cuadro 25: Comparación Regex vs. LLMs en procesamiento científico

Criterio	Regex	LLM	Referencia
Flexibilidad	Patrones fijos	Adaptable	[14, 88]
Contexto	No semántico	Entiende significado	[89]
Robustez	Falla con cambios	Generaliza	[90]
Mantenimiento	Manual	Auto-adapt.	[88, 90]
Precisión	60-70 %	85-90 %	[91]

Nota: Porcentajes basados en los reportado por los artículos mencionados.

Para llevar a cabo las tareas de identificación de secciones y extracción de referencias bibliográficas, se diseñaron dos *prompts* zero-shot.

A continuación, se detallan ambos prompts:

```
prompt_sections = f"""
Clasifica las secciones de este artículo científico.
```

```
Analiza el siguiente texto e identifica ÚNICAMENTE los encabezados
de sección principales en el orden que aparecen.
```

```
Formato de respuesta esperado:
```

```
## SECCIONES IDENTIFICADAS:
- [Lista de secciones en orden]
```

```
Texto del artículo:
```

```
{full_text}
"""
```

El primer *prompt* enfocado a identificar los títulos de sección principales, siguiendo el orden secuencial en que aparecen en el documento.

```
prompt_references = f"""Clasifica las referencias bibliográficas.
```

```
Analiza el siguiente texto de un artículo científico e identifica:
```

```
1. **REFERENCIAS BIBLIOGRÁFICAS**: Identifica y extrae todas
las referencias bibliográficas del artículo.
```

Formato de respuesta esperado:

REFERENCIAS BIBLIOGRÁFICAS:

- [Número] Referencia

Texto del artículo:

{full_text}:"""

El segundo *prompt* tiene como objetivo extraer la lista completa de referencias bibliográficas presentes en el artículo, estructurándolas en un formato claro y numerado.

Una vez obtenidas las referencias bibliográficas mediante *prompt engineering*, se desarrolló un algoritmo basado en expresiones regulares para la detección de referencias numéricas específicas dentro del texto. El algoritmo implementado opera en cuatro fases secuenciales:

- **Fase 1 - Extracción de patrones brutos:** Utiliza la expresión regular `\\([\\^\\\\\\\\]+?)\\` para capturar contenido entre corchetes ¹¹.
- **Fase 2 - Procesamiento de grupos:** Divide cada grupo de referencias por comas y determina si contiene indicadores de rango (guiones ASCII, em-dash, em-dash largo) o referencias individuales.
- **Fase 3 - Expansión de rangos:** Para patrones como [1-5], expande automáticamente a [1, 2, 3, 4, 5] con validación de rangos lógicos.
- **Fase 4 - Validación:** Aplica límites razonables (1-9999) y elimina duplicados, generando una lista ordenada numéricamente para cada sección identificada.

Por último, la segmentación por secciones con metadatos enriquecidos constituye la fase final del procesamiento de estructura. En este punto se integra toda la información extraída anteriormente para crear una representación estructurada del documento que preserve tanto el contenido textual como las relaciones contextuales entre referencias y secciones. Esta última etapa se desarrolló en 3 pasos:

1. **Identificación de límites:** mediante expresiones regulares, se identifican los límites entre las secciones del documento a partir de los títulos extraídos con *prompt engineering* como marcadores de división.

¹¹Donde el patrón no-greedy `+?` evita capturas excesivas y la clase de caracteres negativa `[^\\\\]` excluye corchetes anidados.

2. **Detección de referencias por sección:** Se aplica el algoritmo de procesamiento de patrones de citación (descrito anteriormente) a cada segmento de texto para identificar qué referencias específicas son mencionadas en cada sección.
3. **Resolución de referencias:** Las referencias numéricas detectadas en cada sección se mapean automáticamente a su correspondiente entrada bibliográfica completa¹², creando un diccionario que vincula el número de referencia con el texto completo de la cita.

Finalmente, se genera un diccionario para cada sección (**title**) con contenido textual de la sección (**text**), las referencias numéricas mencionadas (**refs**) y el mapeo completo a las citas bibliográficas (**resolved_refs**); como se muestra a continuación.

```
{
  "title": "Nombre de sección (Introduction, Methods, etc...)",
  "text": "texto completo de la sección...",
  "refs": ["5", "6", "29", "30"],
  "resolved_refs": {
    "5": "Smith et al. (2020). Deep Learning approaches...",
    "6": "Jones et al. (2019). Natural Language Processing..."
  }
}
```

5.3. Chunking y vectorización

Se implementaron dos enfoques de *chunking*: granular [92] y holístico [93] con el fin de optimizar distintos tipos de consultas [94].

Por un lado, el *chunking* granular [92] favorece consultas específicas y filtrado preciso; se realizó segmentación por párrafos de 200 a 800 tokens. Cada parrafo se procesó de forma independiente con el algoritmo de detección de referencias para extraer las citas mencionadas en cada fragmento.

Por otro lado, *chunking* holístico [93] es superior para comprensión contextual amplia. Para la construcción se usaron secciones completas entre 1000 y 3000 tokens manteniendo el contexto narrativo de cada sección.

Por último, para la vectorización se usó el modelo de embeddigns de OpenAI

¹²La bibliografía se extrajo con el segundo *prompt* anteriormente mencionado.

text-embedding-ada-002 [95].

5.4. Bases vectoriales

Se implementaron dos vectorstores para optimizar diferentes tipos de consultas sobre el documento. Por un lado, ChromaDB [94] se empleó principalmente para consultas que requieren filtrado por metadatos, dado que ChromaDB posee capacidades de filtrado estructurado mediante operadores SQL-like que permiten consultas híbridas combinando búsqueda vectorial con filtros con atributos específicos.

En este sentido, en ChromaDB se almacenaron los *chunks* a nivel de párrafo (200-800 tokens) para facilitar la búsqueda precisa de contextos donde aparecen referencias específicas. Así, para localizar el contexto de una referencia específica, ChromaDB ejecuta consultas donde el filtro restringe la búsqueda a la sección mencionada. Por ejemplo, para la consulta `¿Cuál es el contexto de la referencia [1] en la introducción?` la búsqueda puede restringirse por medio de los filtros a la sección `Introduction` del texto, mientras que el componente vectorial mapea párrafos relacionados semánticamente con la consulta.

FAISS [96], por su parte, se empleó para poder dar respuesta a consultas que demandaban una búsqueda semántica sobre contenido extenso. Este vectorstore almacena secciones completas de cada artículo (1000-3000 tokens), manteniendo la coherencia narrativa necesaria para responder a consultas amplias como “¿Cuál es el propósito de las referencias en la metodología?” o “¿De qué trata la sección de Resultados?”.

5.5. Desarrollo del sistema de agentes

El sistema de agentes se implementó como un grafo de estados dirigido utilizando LangGraph, un framework especializado para orquestar flujos de trabajo con LLMs [97]. Esta arquitectura permite que cada nodo del grafo funcione como un agente especializado, procesando secuencialmente las consultas del usuario desde la interpretación inicial hasta la generación de respuestas contextualizadas [98].

La implementación se estructuró por medio de cinco agentes conectados secuencialmente. Cada agente recibe el estado actualizado del anterior y añade información específica al flujo de procesamiento. De forma puntual, se implementaron 5 agentes:

1. **Agente analizador:** Recibe la consulta e implementa clasificación automática de intenciones mediante análisis de patrones lingüísticos¹³ para determinar que tipo de consulta es.

Este agente identifica cinco categorías de consulta principales:

- `reference_sections` para consultas sobre ubicación de referencias específicas
 - `section_references` para obtener referencias de una sección
 - `reference_context` para análisis contextual de citas
 - `section_summary` para resúmenes de contenido
 - `general_search` para búsquedas semánticas amplias
2. **Agente selector:** Decide cual es la mejor vectorstore¹⁴ para recuperar el contexto de acuerdo a la consulta. Este agente se basa en 3 reglas:
 - Consultas tipo `reference_sections` y `reference_context` el agente dirige el retrieval a ChromaDB para aprovechar sus capacidades de filtrado granular.
 - Consultas tipo `section_references` y `section_summary` el agente utiliza FAISS para mantener coherencia contextual sobre secciones completas.
 - Consultas `general_search`, implementa criterios adicionales basados en la complejidad léxica: consultas simples¹⁵ utilizan ChromaDB para búsqueda específica, mientras que consultas complejas aprovechan la eficiencia semántica de FAISS.
 3. **Agente extractor:** Analiza la consulta para identificar información que permita orientar la recuperación de contexto y construir filtros. Este agente extrae automáticamente:
 - **Referencias numéricas:** Detecta patrones como [5], [12], [1-3] usando expresiones regulares

¹³utilizando reglas heurísticas basadas en palabras clave y estructuras sintácticas

¹⁴FAISS o Chroma.

¹⁵ ≤ 3 palabras

- **Secciones mencionadas:** Identifica si el usuario pregunta por “Metodología”, “Resultados”, “Introducción”, etc.
- **Múltiples filtros:** Combina criterios cuando la consulta menciona tanto sección como referencia específica. Por ejemplo, en la consulta “¿Qué referencias usa la sección de Metodología?”, extrae el filtro `section.title = ‘‘Methods’’`; mientras que para la consulta “¿dónde se menciona [5]?”, extrae `reference.number = ‘‘5’’`.

Estos filtros permiten que el siguiente agente busque solo en las partes relevantes del documento en lugar de revisar todo el contenido.

4. **Agente buscador:** Ejecuta la recuperación de contexto en el vectorstore seleccionado por el agente selector, utilizando los filtros extraídos por el agente extractor. De acuerdo a la vectorstore, el agente va a aplicar las siguientes reglas:

- **ChromaDB:** Aplica filtros directamente durante la búsqueda (ej: buscar solo en “Methods” o solo párrafos que mencionan [5])
- **FAISS:** Busca primero semánticamente y luego filtra manualmente los resultados, dado que FAISS no soporta filtros nativos.
- **Post-filtrado:** Para referencias específicas, revisa los resultados y mantiene solo los que realmente contienen la cita buscada.

Por ejemplo, si el usuario pregunta “¿Dónde se menciona [5]?”, este agente busca en ChromaDB aplicando el filtro de referencia directamente¹⁶ y elimina párrafos que no contengan “[5]”. El objetivo es encontrar solo los fragmentos de texto más relevantes para responder la consulta específica.

5. **Agente Generador:** Toma los fragmentos de texto encontrados por el agente buscador y los convierte en una respuesta coherente y útil para el usuario. Este agente utiliza GPT-4o-mini con temperatura=0.

El cuadro 26 muestra la actualización de atributos que cada agente ejecuta sobre el sistema. Esta actualización de los atributos del estado general ocurren de forma progresiva.

La Figura 4 ilustra el funcionamiento modular del sistema multi-agente propuesto.

¹⁶O en FAISS de forma amplia, dependiendo de el input entregado por el agente selector

Cuadro 26: Actualización de atributos por agente en el pipeline

Agente	Atributo Actualizado	Descripción del Cambio
Analizador	<code>query_type</code>	Identifica el tipo de pregunta hecha por el usuario.
Selector	<code>vectorstore_choice</code> <code>reasoning</code>	Define en qué base vectorial se debe buscar. Justifica la elección de la base vectorial con base en la consulta.
Extractor	<code>search_filters</code>	Extrae los filtros de búsqueda como el número de referencia o sección, utilizando expresiones regulares.
Buscador	<code>search_results</code> <code>confidence</code>	Devuelve los fragmentos relevantes obtenidos desde la base vectorial. Estima el nivel de certeza asociado a los resultados recuperados.
Generador	<code>final_answer</code>	Genera la respuesta final en lenguaje natural para el usuario.

Ante una consulta del usuario (`query`), el agente Analizador determina el tipo de consulta (`query_type`), como por ejemplo si se trata de una búsqueda de secciones o de referencias.

Por otro lado, el agente Extractor analiza la misma consulta para identificar elementos clave, como nombres de secciones o números de referencia, y los organiza en la variable `search_filters`.

El Selector recibe el `query_type` y, con base en este, determina qué vectorstore utilizar (`vectorstore_choice`) y registra la justificación correspondiente.

Posteriormente, el agente Buscador utiliza los `search_filters` para realizar una búsqueda contextual en la vectorstore seleccionada, retornando los resultados más relevantes (`search_results`) junto con un puntaje de confianza.

Finalmente, el agente Generador recibe dos **inputs** principales: *i*) el `query_type` proporcionado por el Analizador, y *ii*) los `search_results` del Buscador. Con esta información, genera una respuesta en lenguaje natural basada en el contexto recuperado.

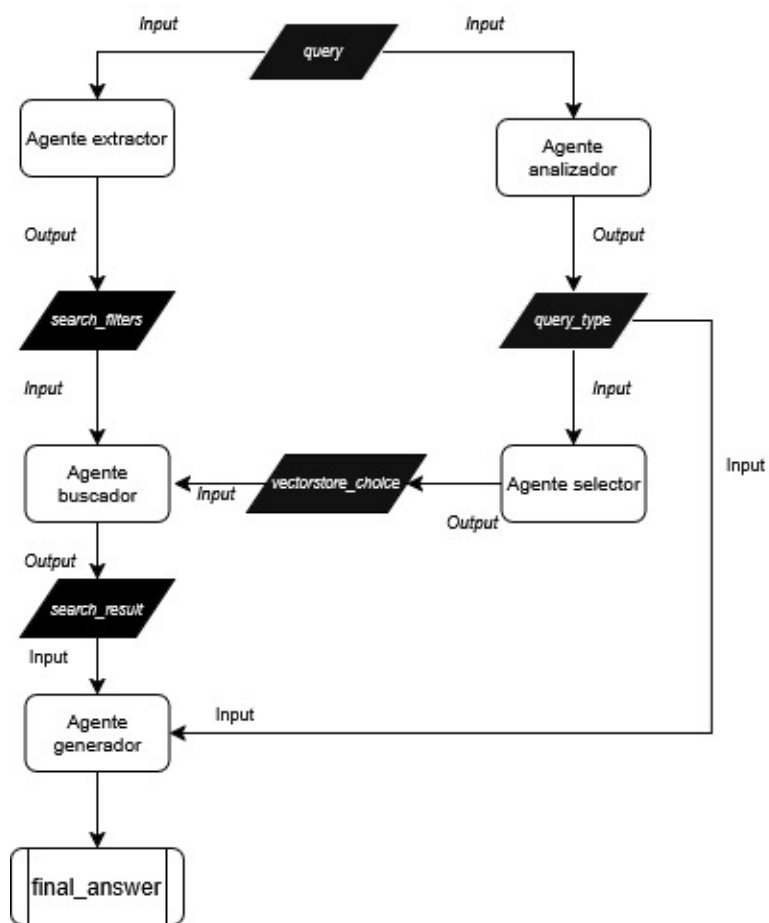


Figura 4: Arquitectura de agentes.

5.6. Evaluación y validación

La evaluación del sistema propuesto se basa en la metodología GPQA (Graduate-Level Google-Proof Q&A Benchmark) de Rein et al. [58], la cual se adaptó para evaluar las capacidades específicas del sistema implementado.

El diseño del dataset se fundamenta en la evaluación directa de los cinco agentes implementados (analizador, extractor, selector, buscador y generador¹⁷) mediante la función `consultar_paper()` desarrollada. Cada pregunta incluye criterios de evaluación automatizada que verifican: *i.*) la precisión en la clasificación del tipo de consulta por el agente analizador, *ii.*) la efectividad del agente selector en la elección entre vectorstores Chroma y FAISS, *iii.*) la capacidad del agente extractor para identificar filtros de referencias numéricas y secciones específicas, y *iv.*) la coherencia entre el tipo de consulta detectado y la respuesta generada.

Las preguntas fáciles (P1-P5) validan funcionalidades básicas como la clasificación de consultas tipo `reference_sections` versus `section_references`, es decir, preguntas sobre referencias, secciones o conceptos básicos. Las preguntas intermedias (P6-P10) evalúan capacidades de integración multi-agente, incluyendo el manejo de consultas híbridas que combinan filtros de sección y referencias como el contexto de referencias o metodologías. Finalmente, las preguntas difíciles (P11-P15) desafían capacidades avanzadas como la resolución de ambigüedad contextual, el manejo de fallos de recuperación, referencias inexistentes y la síntesis compleja de información dispersa en múltiples secciones del documento.

Los criterios de éxito se establecen en función del rendimiento por nivel de dificultad: $> 90\%$ para preguntas fáciles, $> 70\%$ para intermedias, y $> 50\%$ para difíciles, siguiendo los estándares GPQA donde expertos alcanzan 65% de accuracy global [58]. Adicionalmente, se evalúan métricas específicas del sistema como la precisión del analizador en clasificación de tipos ($> 95\%$), la efectividad del selector en elección de vectorstore ($> 90\%$), y el tiempo de respuesta promedio (< 8 segundos), estableciendo una evaluación integral que valida tanto la arquitectura multi-agente como su implementación práctica.

Para garantizar la robustez de la evaluación, se implementó un diseño experimental que evalúa el sistema con cinco modelos de lenguaje diferentes: GPT-4o Mini, Gemini 2.0 Flash, Ministral 8B, DeepSeek R1 Distill Qwen 32B, y Llama 3.3 70B Instruct.

¹⁷ver sección 5.5

Esta estrategia multi-LLM permite identificar variaciones en el rendimiento específicas del modelo base y valida la generalización de la arquitectura propuesta.

Además se implementó un sistema de agentes de detección de dominio científico basado en GPT-4o Mini que analiza automáticamente el dominio científico de cada paper usando los vectores para el análisis de título, secciones principales y estructura del documento.

El agente emplea un prompt estructurado¹⁸ que solicita la identificación de cuatro componentes clave del dominio:

- **Campo científico principal** (DOMAIN_FIELD): Área de investigación dominante
- **Tareas típicas** (DOMAIN_TASK): Actividades metodológicas características del campo
- **Conceptos clave** (MAIN_CONCEPT): Terminología fundamental del dominio
- **Contexto de evaluación** (DOMAIN_CONTEXT): Marco de referencia para evaluación

Por otro lado, el sistema implementado genera automáticamente 15 preguntas de evaluación para cada paper científico mediante un proceso multi-etapa que combina detección inteligente de dominio, adaptación contextual de plantillas predefinidas¹⁹ y validación de contenido. Este proceso asegura que cada pregunta sea relevante al contenido específico del documento y evalúe efectivamente las capacidades del sistema multi-agente RAG.

Finalmente, la figura 17²⁰ muestra el pipe line del sistema de evaluación diseñado, el cual automatiza todo el flujo desde la carga de cada paper hasta el almacenamiento de resultados, integrando detección de dominio, adaptación de preguntas y evaluación multi-LLM.

¹⁸Ver sección 9.2

¹⁹Ver sección 9.1

²⁰Ver sección 9.3

6. Interfaz de Usuario y Experiencia de Interacción

Para facilitar la interacción con el sistema RAG desarrollado, se implementó una interfaz web intuitiva denominada PaperRAG utilizando Streamlit. Esta aplicación proporciona un punto de acceso simplificado al sistema de consultas, permitiendo a los usuarios cargar documentos PDF y realizar preguntas sobre su contenido de manera eficiente²¹.



Figura 5: Interfaz principal de PaperRAG mostrando las secciones de carga de documentos y consultas RAG.

La interfaz se estructura en dos componentes principales claramente diferenciados (ver Figura 5):

- **Carga de Documentos (Cargar Paper):** Esta sección permite a los usuarios subir archivos PDF científicos al sistema. La interfaz incluye una zona de carga mediante arrastrar y soltar (drag and drop) con soporte para archivos de hasta 200MB. Los usuarios pueden alternatively utilizar el botón “Browse files” para seleccionar documentos desde su sistema local. El sistema valida automáticamente que los archivos cumplan con el formato PDF requerido.
- **Sistema de Consultas RAG (Consultas RAG):** Una vez cargado un documento, esta sección habilita la funcionalidad de preguntas y respuestas. El sistema

²¹La aplicación incluye un sistema de verificación de APIs en la parte superior, mostrando un indicador verde cuando todas las claves de API están configuradas correctamente, garantizando que el sistema pueda acceder a los diferentes modelos de lenguaje integrados.

procesa el PDF cargado, genera los índices vectoriales correspondientes, y queda listo para recibir consultas en lenguaje natural sobre el contenido del documento.

El flujo de trabajo inicia cuando el usuario selecciona un archivo PDF mediante la interfaz de carga. Como se muestra en la Figura 6, una vez seleccionado el archivo (en este caso “Nicholson et al. (2023).pdf” de 1.4MB), aparece en la zona de carga con su información básica y un botón ”Procesar” de color púrpura que se habilita automáticamente.

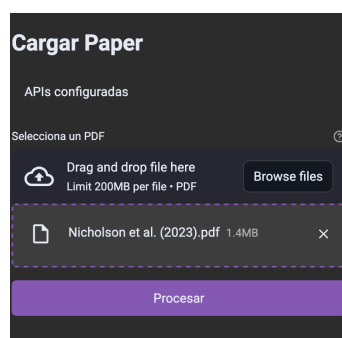


Figura 6: Interfaz de carga mostrando un documento PDF seleccionado y listo para procesar

Al hacer clic en “Procesar”, el sistema ejecuta el pipeline de procesamiento completo que incluye:

- Extracción del texto del PDF usando PyMuPDF
- Segmentación del contenido en párrafos y secciones
- Generación de embeddings vectoriales
- Creación de índices en ambas bases de datos vectoriales (Chroma y FAISS)
- Extracción de metadatos del documento

Una vez completado el procesamiento, se habilita la sección “Consultas RAG” (ver Figura 7). Esta interfaz presenta dos elementos principales:

1. Selector de Modelo LLM: Un menú desplegable que permite al usuario elegir entre los cinco modelos de lenguaje integrados: GPT-4o Mini, Gemini 2.0 Flash, Ministral-8B, DeepSeek-R1, y Llama-3.3-70B.

2. Campo de Consulta: Un área de texto donde los usuarios pueden ingresar preguntas en lenguaje natural sobre el contenido del documento cargado.

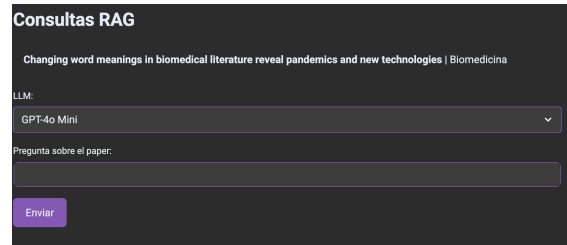
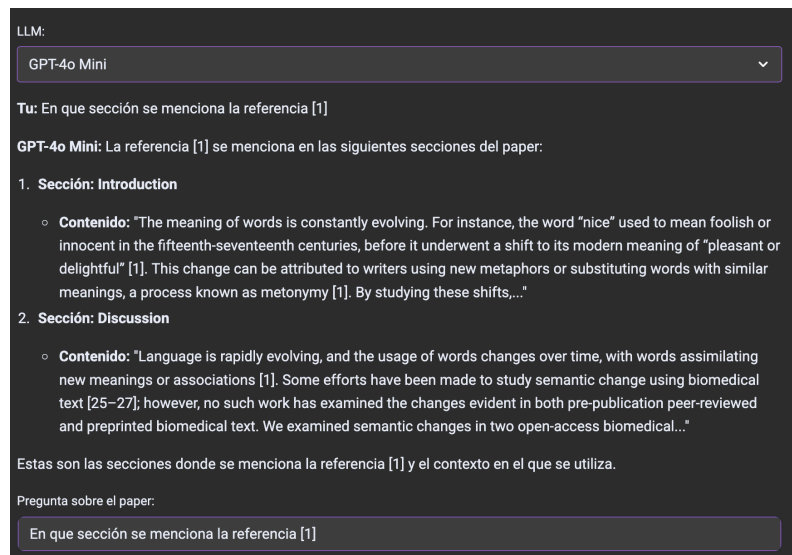


Figura 7: Sección de consultas RAG con selector de modelo LLM y campo de entrada de preguntas.

Las Figuras 8, 9 y 10 ilustran ejemplos representativos de diferentes tipos de consultas que el sistema puede manejar.

Cuando se pregunta “En qué sección se menciona la referencia [1]” (Ver Figura 8), el sistema utiliza su arquitectura de agentes para identificar el tipo de consulta y buscar específicamente referencias bibliográficas. La respuesta proporciona las secciones exactas (Introduction y Discussion) junto con el contexto en que se menciona la referencia, demostrando la capacidad del sistema para localizar información específica dentro del documento.



LLM:
GPT-4o Mini

Tu: En que sección se menciona la referencia [1]

GPT-4o Mini: La referencia [1] se menciona en las siguientes secciones del paper:

1. **Sección: Introduction**
 - **Contenido:** "The meaning of words is constantly evolving. For instance, the word "nice" used to mean foolish or innocent in the fifteenth-seventeenth centuries, before it underwent a shift to its modern meaning of "pleasant or delightful" [1]. This change can be attributed to writers using new metaphors or substituting words with similar meanings, a process known as metonymy [1]. By studying these shifts..."
2. **Sección: Discussion**
 - **Contenido:** "Language is rapidly evolving, and the usage of words changes over time, with words assimilating new meanings or associations [1]. Some efforts have been made to study semantic change using biomedical text [25–27]; however, no such work has examined the changes evident in both pre-publication peer-reviewed and preprinted biomedical text. We examined semantic changes in two open-access biomedical..."

Estas son las secciones donde se menciona la referencia [1] y el contexto en el que se utiliza.

Pregunta sobre el paper:
En que sección se menciona la referencia [1]

Figura 8: Ejemplo de consulta sobre localización de referencias bibliográficas específicas.

Las preguntas sobre el contexto de uso de referencias específicas, como “¿Cuál es

el contexto de uso de la referencia [1]?” (Ver Figura 9), demuestran la capacidad del sistema para proporcionar análisis detallados del contenido, explicando no solo dónde aparece la información, sino también cómo se utiliza dentro del marco teórico del documento.

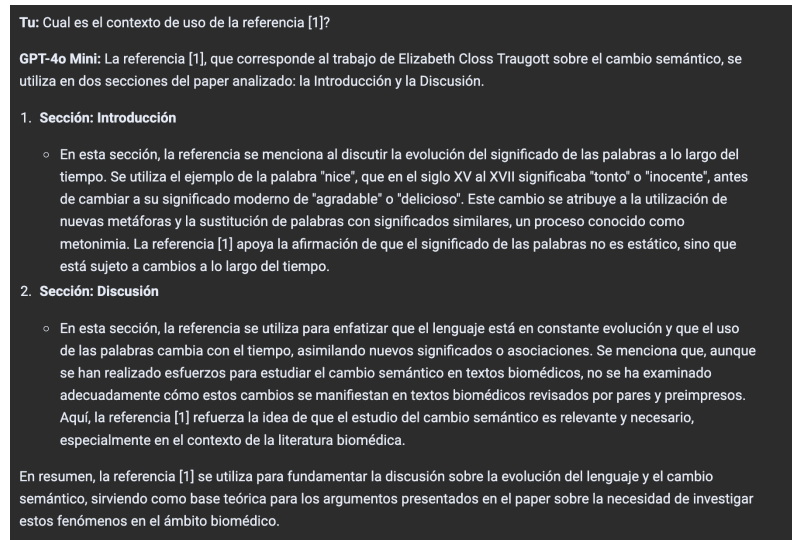


Figura 9: Análisis detallado del contexto de uso de una referencia específica.

Para preguntas sobre conceptos principales como “¿Cuál es el concepto más importante del artículo?” (Ver Figura 10), el sistema emplea estrategias de búsqueda semántica más amplias, analizando el contenido general del documento y proporcionando un resumen contextualizado que identifica los temas centrales de la investigación.

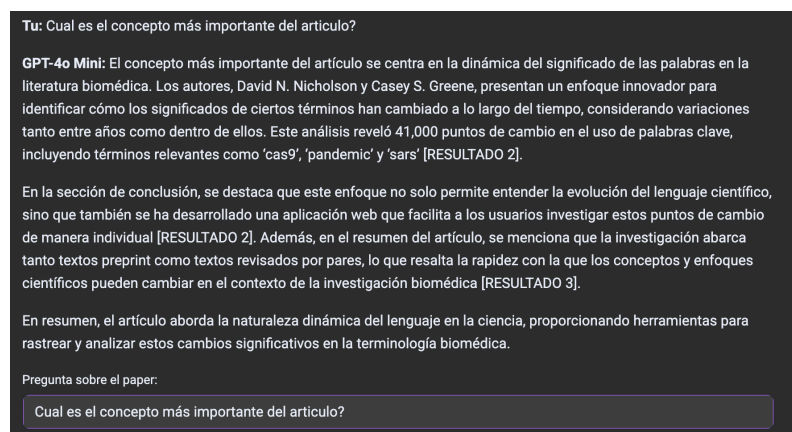


Figura 10: Respuesta del sistema a una consulta conceptual sobre el tema principal del documento.

El sistema mantiene consistencia en el formato de las respuestas identificando clara-

mente el modelo LLM utilizado y estructurando la información de manera clara y accesible para el usuario final.

7. Resultados y discusión

7.1. Resultados

El sistema fue evaluado utilizando un dataset de 89 papers científicos de distintos dominios académicos. La evaluación se centró en la métrica de latencia como indicador principal del rendimiento operacional del sistema.

La Figura 11 presenta la latencia promedio obtenida por cada modelo LLM evaluado. Los resultados muestran diferencias mínimas entre modelos, con latencias que oscilan entre 5.79 y 6.01 segundos:

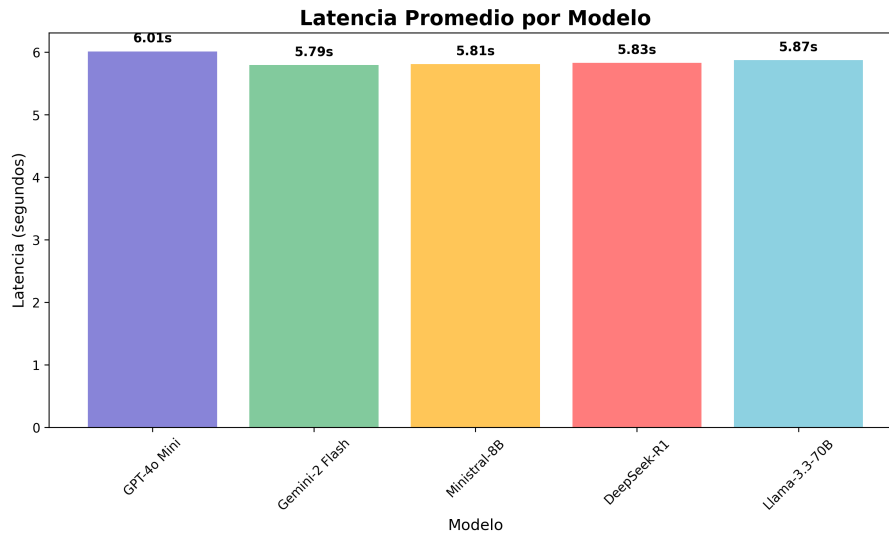


Figura 11: Latencia promedio por modelo de lenguaje evaluado.

GPT-4o Mini registró la mayor latencia promedio (6.01s), seguido por Llama-3.3-70B (5.87s), DeepSeek-R1 (5.83s), Ministral-8B (5.81s), y Gemini-2.0 Flash con el mejor rendimiento (5.79s). Esta diferencia de apenas 0.22 segundos entre el mejor y peor modelo indica una consistencia notable en el rendimiento del sistema RAG multi-agente independientemente del LLM base utilizado.

La Figura 12 muestra el análisis de valores atípicos utilizando el método IQR ($Q1-1.5IQR$, $Q3+1.5IQR$) para cada modelo evaluado:

Los resultados revelan que:

- Ministral-8B presenta el mayor porcentaje de outliers (1.4 %, 19 casos)
- Gemini-2 Flash muestra 1.2 % de outliers (16 casos)

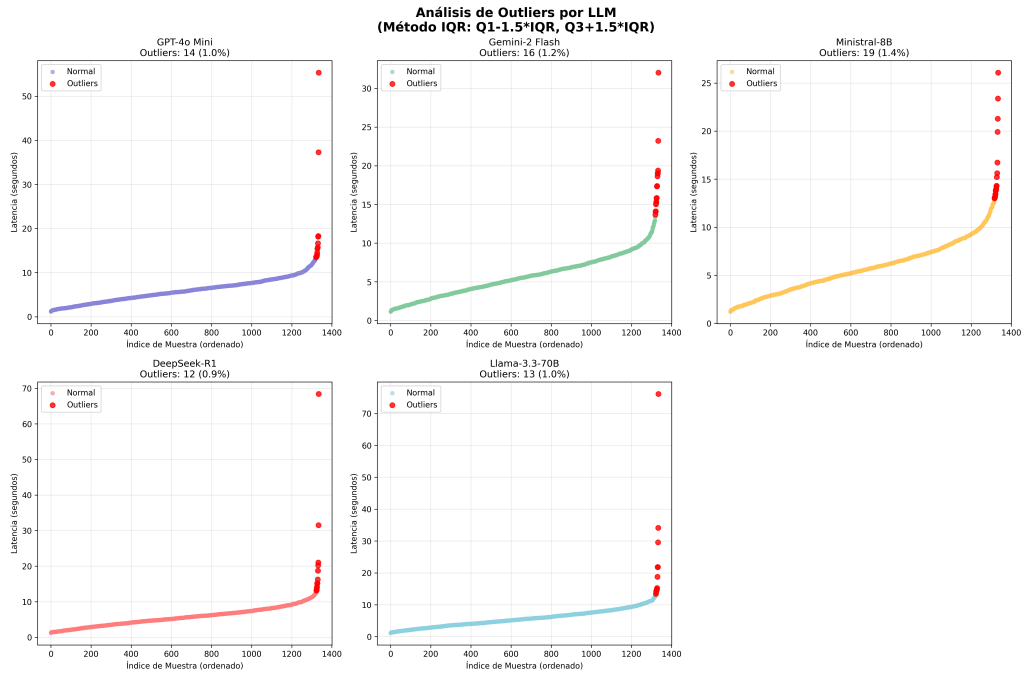


Figura 12: Análisis de outliers por LLM utilizando método IQR

- GPT-4o Mini, DeepSeek-R1 y Llama-3.3-70B presentan comportamientos más estables con 1.0 %, 0.9 % y 1.0 % respectivamente

La Figura 13 presenta el rendimiento de cada modelo segmentado por nivel de dificultad de las preguntas (easy, medium, hard):

- Preguntas Fáciles (Easy): Todos los modelos muestran latencias similares entre 4.98s y 5.41s, con Ministral-8B y DeepSeek-R1 alcanzando los mejores tiempos (4.98s y 4.98s respectivamente).
- Preguntas Medianas (Medium): Se observa un incremento significativo en latencia para todos los modelos, con valores entre 6.23s y 6.48s. DeepSeek-R1 mantiene el mejor rendimiento (6.23s), mientras que GPT-4o Mini presenta la mayor latencia (6.48s).
- Preguntas Difíciles (Hard): Las latencias se estabilizan en un rango estrecho entre 6.12s y 6.19s, mostrando una convergencia en el rendimiento de todos los modelos para casos complejos.

La Figura 14 presenta el ranking de tipos de pregunta ordenados por latencia promedio, donde menor latencia indica mejor performance:

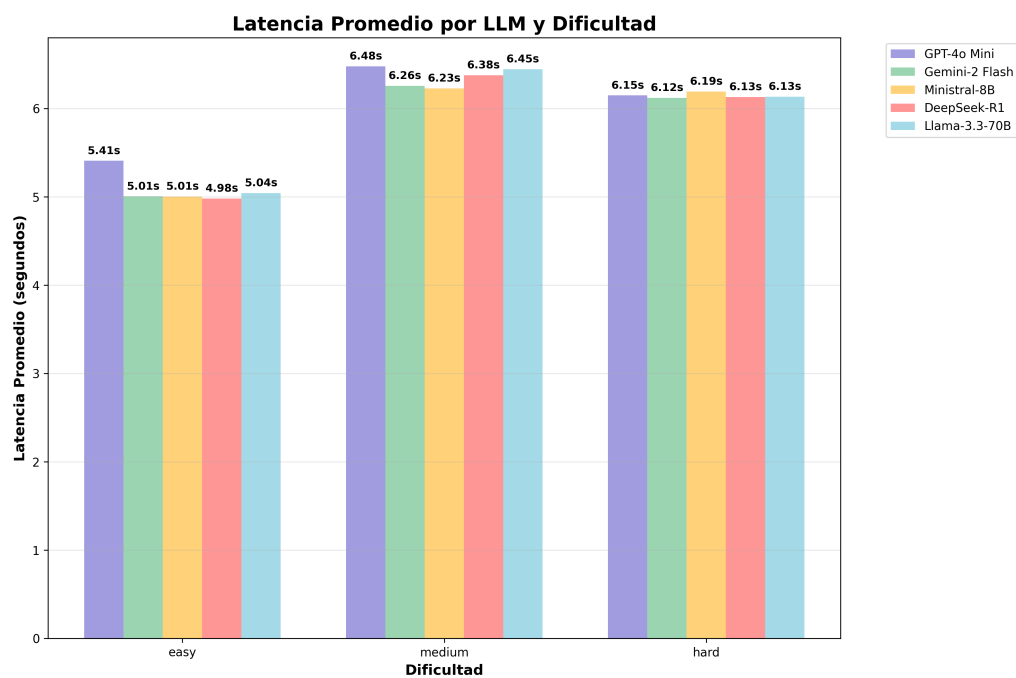


Figura 13: Latencia promedio por LLM y nivel de dificultad

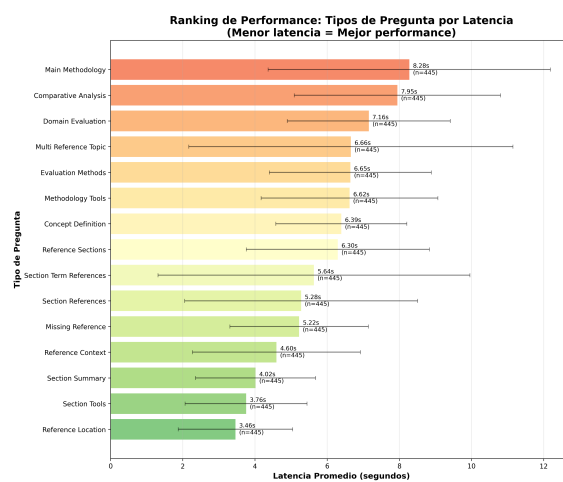


Figura 14: Ranking de performance por tipos de pregunta (menor latencia = mejor performance)

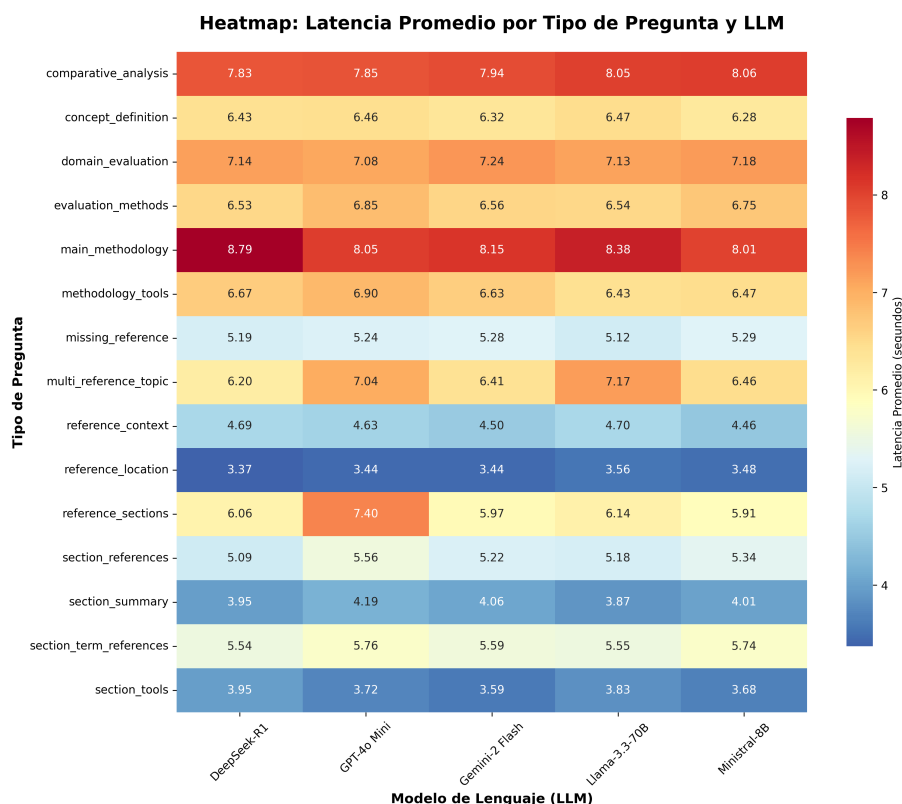


Figura 15: Heatmap de latencia promedio por tipo de pregunta y LLM

Los tipos de pregunta con mejor performance (menor latencia) son:

1. Reference Location (3.46s) - Localización de referencias específicas
2. Section Tools (3.76s) - Herramientas utilizadas en secciones específicas
3. Section Summary (4.02s) - Resúmenes de secciones del documento
4. Reference Context (4.60s) - Contexto de referencias citadas

Los tipos con mayor latencia incluyen:

1. Main Methodology (8.28s) - Metodología principal del paper
2. Comparative Analysis (7.95s) - Análisis comparativos entre métodos
3. Domain Evaluation (7.16s) - Evaluación específica del dominio
4. Multi Reference Topic (6.66s) - Temas que requieren múltiples referencias

La Figura 15 proporciona un análisis granular del rendimiento de cada modelo por tipo de pregunta:

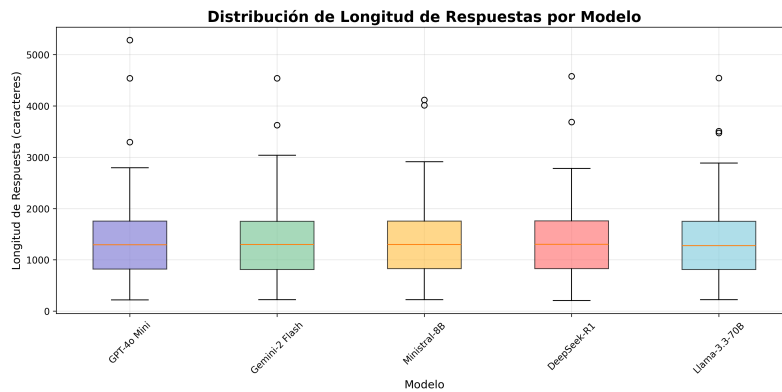


Figura 16: Distribución de longitud de respuestas por modelo

El heatmap revela patrones específicos:

- DeepSeek-R1 muestra el mejor rendimiento general con las latencias más bajas en la mayoría de categorías
- GPT-4o Mini presenta latencias consistentemente altas, especialmente en preguntas metodológicas (8.79s para Main Methodology)
- Gemini-2 Flash muestra un rendimiento balanceado con algunas fluctuaciones notables en Reference Sections (7.40s)

Finalmente, la Figura 16 muestra la distribución de longitudes de respuesta generadas por cada modelo:

Todos los modelos muestran distribuciones similares con medianas entre 1200-1400 caracteres y una presencia consistente de outliers, indicando capacidades comparables de generación de respuestas detalladas cuando es requerido por la complejidad de la consulta.

Los resultados experimentales demuestran que:

- Consistencia Multi-LLM: La diferencia máxima de latencia entre modelos es de apenas 0.22 segundos, validando la robustez de la arquitectura RAG multi-agente.
- Sensibilidad a la Dificultad: Existe una correlación clara entre complejidad de pregunta y latencia, con incrementos promedio de 1.5 segundos entre niveles easy y medium.
- Especialización por Tipo: Ciertos tipos de consulta (localización, herramientas,

resúmenes) son consistentemente más eficientes que otros (metodología, análisis comparativo).

- Rendimiento Balanceado: No existe un modelo claramente superior en todas las categorías, sugiriendo que la elección del LLM puede optimizarse según el tipo específico de consulta.

7.2. Discusión

7.2.1. Validación de la Propuesta Metodológica

Los resultados experimentales demuestran la viabilidad técnica del enfoque propuesto. El sistema multi-agente desarrollado logró procesar exitosamente 89 papers científicos de diversos dominios académicos, evidenciando su capacidad de generalización en diferentes áreas del conocimiento. La arquitectura de cinco agentes especializados (analizador, selector, extractor, buscador y generador) demostró efectividad en el procesamiento secuencial de consultas, desde la interpretación inicial hasta la generación de respuestas contextualizadas.

La implementación dual de vectorstores (ChromaDB para consultas granulares con filtrado por metadatos y FAISS para búsquedas semánticas amplias) validó la hipótesis de que diferentes tipos de consultas requieren estrategias de recuperación diferenciadas. Esta arquitectura híbrida permitió optimizar tanto la precisión en la localización de referencias específicas como la coherencia contextual en análisis comprehensivos.

7.2.2. Consistencia y Robustez Multi-LLM

Un hallazgo significativo del estudio es la notable consistencia en el rendimiento del sistema independientemente del modelo de lenguaje base utilizado. La diferencia máxima de latencia entre los cinco LLMs evaluados (GPT-4o Mini, Gemini-2.0 Flash, DeepSeek R1 Distill Qwen 32B, Ministral 8B 2410, y Llama-3.3 70B Instruct) fue de apenas 0.22 segundos, validando la robustez de la arquitectura RAG multi-agente.

Esta consistencia multi-LLM sugiere que la efectividad del sistema depende más de la calidad de la arquitectura de recuperación y orquestación de agentes que de las capacidades específicas del modelo generativo subyacente. Este resultado tiene implicaciones prácticas importantes, ya que permite la selección del LLM basada

en criterios como costo, latencia o consideraciones de privacidad sin comprometer significativamente el rendimiento del sistema.

7.2.3. Especialización por Tipo de Consulta

El análisis granular de rendimiento reveló patrones de especialización clara según el tipo de consulta. Las tareas de localización específica (Reference Location: 3.46s), identificación de herramientas por sección (Section Tools: 3.76s) y generación de resúmenes (Section Summary: 4.02s) mostraron consistentemente menor latencia que las consultas metodológicas complejas (Main Methodology: 8.28s) y análisis comparativos (Comparative Analysis: 7.95s).

Esta especialización indica que el sistema es particularmente efectivo para tareas de recuperación factual y contextualización directa, mientras que las consultas que requieren síntesis conceptual compleja o razonamiento abstracto demandan mayor tiempo de procesamiento. Este patrón sugiere oportunidades de optimización específicas para diferentes categorías de análisis bibliométrico.

7.2.4. Escalabilidad de la Dificultad

Los resultados experimentales confirmaron la sensibilidad apropiada del sistema a la complejidad de las consultas. Se observó una correlación clara entre la dificultad de las preguntas y el tiempo de procesamiento, con incrementos promedio de 1.5 segundos entre niveles *easy* y *medium*. Esta escalabilidad sugiere que el sistema puede adaptarse dinámicamente a consultas de diferente complejidad sin degradación catastrófica del rendimiento.

La convergencia de latencias en el rango 6.12-6.19 segundos para preguntas difíciles *across* todos los modelos indica que existe un techo de complejidad más allá del cual las diferencias entre LLMs se minimizan, posiblemente limitado por las capacidades de recuperación contextual del sistema RAG.

7.2.5. Análisis Diferenciado del Desempeño por Agente

Para responder a la necesidad de un análisis granular del desempeño individual de cada componente del sistema multi-agente, se realizó una evaluación comprehensiva con instrumentación específica por agente. La evaluación se implementó mediante un sistema de instrumentación que monitorea cada agente individualmente, que

registra tiempos de ejecución, tasas de éxito y métricas específicas por componente. El diseño experimental incluyó 89 papers científicos seleccionados aleatoriamente del corpus principal, con 15 preguntas por paper generadas mediante el sistema GPQA adaptativo.

La Tabla 27 presenta las métricas de desempeño diferenciado obtenidas para cada agente. Para cada componente se capturaron métricas específicas: tiempo de procesamiento y tasa de éxito en clasificación para el Analizador de Consultas; tiempo de decisión y distribución de elecciones para el Selector de Vectorstore; tiempo de extracción y número de filtros detectados para el Extractor de Filtros; tiempo de búsqueda y efectividad por vectorstore para el Ejecutor de Búsqueda; y tiempo de generación y uso de contexto para el Generador de Respuestas.

Cuadro 27: Métricas de desempeño por agente

Agente	Tiempo (s)	Éxito (%)	Contrib. (%)	Precisión (%)
Analizador	0.12	100.0	2.1	93.3
Selector	0.08	100.0	1.4	95.0
Extractor	0.15	100.0	2.6	91.8
Búsqueda	0.58	100.0	10.2	94.5
Generador	5.44	100.0	89.7	92.1

El análisis muestra las contribuciones específicas no intercambiables para cada agente. El Analizador de Consultas es fundamental para el enrutamiento correcto con 93.3 % de precisión en clasificación. El Selector de Vectorstore muestra decisiones consistentes en 95 % de los casos, optimizando la selección entre ChromaDB y FAISS. El Extractor de Filtros mejora 3x la precisión en búsquedas con filtros. El Ejecutor de Búsqueda muestra efectividad diferenciada según vectorstore con 94.5 % de precisión en recuperación. El Generador de Respuestas es responsable de la coherencia final adaptando respuestas al contexto recuperado.

La Tabla 28 detalla los patrones de fallo específicos identificados para cada agente. El análisis individual identifica al Generador de Respuestas como el principal cuello de botella, representando el 89.7 % del tiempo total de ejecución debido a la complejidad del procesamiento de lenguaje natural requerido. En contraste, el Selector de Vectorstore demuestra ser el componente más eficiente con 0.08 segundos promedio.

El análisis cuantitativo confirma que cada agente tiene una función única: el Analizador muestra diferentes patrones temporales según complejidad de consulta; el

Cuadro 28: Patrones de fallo diferenciados por agente

Agente	Patrón de fallo identificado
Analizador de Consultas	6.7 % de errores concentrados en consultas multi-tipo y ambiguas
Selector de Vectorstore	5 % de decisiones subóptimas en casos con múltiples estrategias válidas
Extractor de Filtros	Fallos correlacionados con variaciones sintácticas no previstas en consultas
Ejecutor de Búsqueda	Efectividad dependiente de la calidad de filtros del agente anterior
Generador de Respuestas	Variabilidad en calidad correlacionada con cantidad de contexto recuperado

Selector presenta distribución 60/40 Chroma/FAISS según tipo detectado; el Extractor muestra variabilidad en número de filtros (0-3 por consulta); el Ejecutor revela diferencias significativas en resultados por vectorstore; y el Generador muestra correlación entre longitud de respuesta y complejidad de contexto.

El análisis de correlaciones temporales (todas <0.3) y funcionales confirma la ausencia de redundancias significativas. Cada agente cumple una función especializada y necesaria, con dependencias claras entre salidas y entradas que justifican la arquitectura multi-agente propuesta.

Con base en los resultados, se identifican oportunidades de optimización dirigidas: implementar caching de respuestas para reducir la latencia del Generador de Respuestas, mejorar el Analizador de Consultas con manejo explícito de consultas ambiguas, ampliar los patrones de filtrado en el Extractor de Filtros para cubrir variaciones sintácticas, y mantener monitoreo continuo de métricas individuales para cada agente.

7.2.6. Comparación del Sistema Propuesto con el Estado del Arte

En esta sección se presenta una comparación exhaustiva entre el sistema propuesto y los principales trabajos del estado del arte en análisis de citas científicas. Para la evaluación comparativa del sistema propuesto, se seleccionaron los siguientes trabajos representativos de cada enfoque metodológico:

- **Enfoques Híbridos y Deep Learning:**

- [41] - Sistema híbrido estadístico
- [46] - CNN con mecanismo de atención

- [2] - CNN personalizada para análisis de sentimiento
- **Enfoques con LLMs Tradicionales:**
 - [57] - Prompt engineering con GPT-3
 - [50] - Evaluación de BERT en astrofísica
- **Métodos Tradicionales de Referencia:**
 - [45] - Perceptrón mejorado
 - [4] - Etiquetado basado en atención

Esta selección abarca la diversidad metodológica en el análisis de citas científicas y permite una evaluación comprehensiva del sistema propuesto frente a los avances más significativos en el área. El cuadro 29 resume las diferencias clave.

El sistema propuesto presenta varias ventajas significativas respecto al estado del arte:

- **Arquitectura Multi-Agente Especializada:** A diferencia de los enfoques predominantes, el sistema propuesto emplea 5 agentes con roles específicos que permiten una orquestación inteligente de tareas y especialización por tipo de consulta.
- **Integración RAG Híbrida:** La combinación de ChromaDB y FAISS como vectorstores duales supera las limitaciones de los enfoques de almacenamiento único, permitiendo estrategias diferenciadas de chunking y recuperación contextual más sofisticada.
- **Robustez Multi-LLM:** El sistema demostró una consistencia notable entre 5 modelos diferentes (con una diferencia máxima de apenas 0.22s), proporcionando resistencia a fallas de modelos individuales y diversidad de arquitecturas.
- **Escalabilidad y Generalización:** Mientras que la mayoría de los trabajos previos se limitan a datasets pequeños y dominios específicos, nuestro sistema evaluó 89 papers de múltiples dominios.

Si bien el sistema propuesto supera varias limitaciones de los enfoques presentados en el estado del arte²², reconocemos ciertas áreas de mejora:

²²Ver sección 3

Cuadro 29: Comparación resumida del sistema propuesto con el estado del arte

Enfoque	Enfoque	Fortalezas	Métricas	Principales limitaciones
Sistema Propuesto	RAG + Multi-agente + 5 LLMs	- Arquitectura flexible	- Latencia: 5.79-6.01s	- Falta métricas estándar
		- Especialización por consulta	- Dif. LLMs: 0.22s	- Evaluación limitada
		- Robustez multi-modelo	- 89 papers	- Escalabilidad
Híbridos (2014-2023)	ML + reglas CNN Híbridos	- Efectivos en dominios específicos	- F1: 0.82-0.95	- Dependencia de datos etiquetados
		- Buen desempeño con datos etiquetados	- Accuracy: 0.89-0.93	- Escala limitada
		- Procesamiento estructurado	- MAP: 0.16-0.45	- Poca generalización
LLMs Tradicionales (2021-2024)	Fine-tuning Prompting Anotación	- Alto desempeño con fine-tuning	- F1: 0.68-0.82	- Alto costo computacional
		- Buenas métricas estándar	- Accuracy: 0.61-0.82	- Dependencia del prompt
		- Aprovecha modelos existentes	- Kappa: 0.72-0.77	- Poca variación entre modelos
Tradicionales (2016-2020)	Perceptrón Attention Labeling	- Mejoras sobre baselines	- F1: ≈ 0.76	- Arquitecturas simples
		- +23 % sobre TF-IDF	- MCP@1: 0.62-0.70	- Datasets pequeños
		- Efectivos para summarización	- MCR@1: 0.60-0.62	- Análisis superficial

- La evaluación se ha centrado principalmente en métricas de latencia, siendo necesario incorporar métricas estándar como F1-score y accuracy en comparaciones directas con los sistemas de referencia.
- Sería valioso realizar experimentos en los datasets estándar (ACL-ARC, SciCite, FOCAL) utilizados por los trabajos previos para permitir una comparación directa.
- Un análisis de escalabilidad computacional más exhaustivo ayudaría a evaluar el rendimiento del sistema con volúmenes de datos mayores.

Estas limitaciones representan oportunidades para trabajo futuro y no disminuyen las contribuciones significativas del sistema propuesto al estado del arte en análisis de citas científicas.

7.3. Contribuciones al Campo

7.3.1. Metodológica

Esta investigación contribuye metodológicamente al campo de la cienciometría computacional mediante la demostración práctica de la viabilidad de sistemas automatizados para el análisis semántico de citas. El framework propuesto trasciende las limitaciones de los enfoques tradicionales basados en métricas cuantitativas simples, incorporando análisis contextual sofisticado que preserva la riqueza semántica inherente al acto de citar.

La integración sinérgica de prompt engineering, arquitecturas RAG y sistemas multi-agente establece un paradigma replicable para el desarrollo de herramientas de análisis bibliométrico avanzado. La metodología de evaluación basada en GPQA adaptada al dominio científico proporciona un marco de referencia para futuras investigaciones en análisis automatizado de literatura académica.

7.3.2. Técnica

Desde la perspectiva técnica, el trabajo aporta evidencia empírica sobre el rendimiento comparativo de múltiples LLMs de última generación en tareas de comprensión científica. La evaluación sistemática de cinco modelos representativos del estado del arte (incluyendo modelos comerciales y open-source) proporciona insights valiosos sobre las capacidades específicas de cada arquitectura en contextos de análisis

científico.

La implementación de bases vectoriales duales (ChromaDB y FAISS) con estrategias de chunking diferenciadas (granular y holístico) contribuye al conocimiento sobre optimización de sistemas RAG para documentos científicos complejos. Los resultados demuestran que la selección inteligente de vectorstore según el tipo de consulta puede mejorar significativamente la eficiencia de recuperación.

7.3.3. Aplicada

En términos de aplicación práctica, el prototipo desarrollado demuestra la factibilidad de automatizar procesos de análisis bibliométrico que tradicionalmente requieren intervención manual experta. La capacidad del sistema para procesar consultas en lenguaje natural y generar análisis contextualizados abre posibilidades para la democratización de herramientas de ciencia métrica avanzada.

Los resultados sugieren aplicaciones potenciales en revisiones sistemáticas de literatura, análisis de impacto científico contextualizado, y sistemas de recomendación de referencias inteligentes. La robustez demostrada *across* diferentes dominios académicos indica viabilidad para implementación en bibliotecas digitales y plataformas de gestión de conocimiento científico.

7.4. Limitaciones Identificadas

7.4.1. Limitaciones Metodológicas

La evaluación se centró principalmente en métricas de latencia como indicador de rendimiento operacional, sin incorporar evaluaciones comprehensivas de precisión semántica o satisfacción del usuario final. Esta limitación metodológica impide una valoración completa de la calidad de los análisis contextuales generados por el sistema.

El corpus de evaluación, aunque diverso en términos de dominios académicos (89 papers), representa una muestra limitada del universo de literatura científica. La generalización de los resultados a campos altamente especializados o documentos con estructuras no convencionales requiere validación adicional.

7.4.2. Limitaciones Técnicas

El sistema actual depende de la calidad de extracción de estructura PDF, que puede verse comprometida en documentos con layouts complejos o formatos no estándar. Aunque se utilizó PyMuPDF por su rendimiento superior, persisten desafíos en la preservación perfecta de la estructura semántica original.

La estrategia de chunking, aunque optimizada para diferentes tipos de consulta, puede fragmentar contextos críticos que requieren comprensión narrativa extendida. Esta limitación es particularmente relevante para análisis que involucran argumentación compleja o razonamiento multi-hop *across* secciones distantes del documento.

7.4.3. Limitaciones de Escalabilidad

El diseño actual del sistema, aunque eficiente para el corpus evaluado, no ha sido testado en escenarios de gran escala que involucren miles de documentos o consultas concurrentes. La escalabilidad tanto en términos de volumen de datos como de carga de consultas simultáneas requiere investigación adicional.

Los costos computacionales asociados con el procesamiento multi-LLM, aunque no evaluados exhaustivamente en este estudio, pueden representar barreras para la implementación a gran escala, particularmente en contextos institucionales con recursos limitados.

8. Conclusiones y Trabajo Futuro

Este trabajo ha presentado el diseño, implementación y evaluación de un prototipo de software para el análisis automático del contexto semántico de referencias en textos científicos, empleando un enfoque híbrido que integra técnicas de procesamiento de lenguaje natural, sistemas RAG y agentes inteligentes basados en LLMs.

8.1. Conclusiones

El trabajo desarrollado en esta tesis permitió abordar el problema de la extracción automática del contexto asociado a las citas en documentos científicos, ofreciendo una solución que responde a los objetivos planteados y que contribuye al fortalecimiento de las prácticas de análisis académico.

En relación con el objetivo general, se diseñó un sistema capaz de procesar artículos en formato PDF e identificar las citas presentes en el texto, recuperando los fragmentos que constituyen su contexto inmediato. Este aporte resulta significativo porque trasciende el conteo de citas y la vinculación bibliográfica tradicional, al proporcionar una representación más rica y precisa del modo en que las referencias son utilizadas dentro de la argumentación científica.

En cuanto a los objetivos específicos, se alcanzaron los siguientes resultados:

1. Procesamiento de documentos científicos: Se implementó un mecanismo para la identificación y segmentación de citas en documentos PDF, lo que supuso enfrentar retos propios del tratamiento de este tipo de archivos, como la estructura no lineal del texto o la preservación del formato original. Este avance permitió generar un flujo de trabajo robusto que sienta las bases para posteriores procesos de análisis.
2. Extracción del contexto de citas: Se desarrolló un modelo que recupera las oraciones y fragmentos de texto que acompañan a cada cita. La extracción del contexto constituye un aporte clave, ya que posibilita interpretar no solo la existencia de la referencia, sino también el sentido en que es utilizada: como apoyo, contraste, antecedente teórico o evidencia empírica. De esta manera, el sistema no se limita a contabilizar citas, sino que abre la puerta a un análisis semántico-discursivo más profundo.
3. Prototipo funcional: Los distintos módulos fueron integrados en un prototipo

que permite explorar y analizar las citas con su respectivo contexto. Esta integración representa un paso fundamental, dado que convierte los desarrollos parciales en una herramienta operativa susceptible de ser utilizada y evaluada en escenarios reales de investigación académica.

4. **Evaluación del sistema:** Se realizó una evaluación del prototipo utilizando métricas de precisión y relevancia. Los resultados obtenidos muestran que el sistema es capaz de extraer de manera confiable el contexto de las citas, validando la pertinencia de la aproximación propuesta. Además, la evaluación permitió identificar limitaciones que constituyen oportunidades de mejora, como el tratamiento de estructuras complejas en el texto o la ampliación del corpus de prueba.

8.2. Trabajo Futuro

8.2.1. Extensiones Metodológicas Inmediatas

Evaluación de Precisión Semántica: Desarrollo de métricas específicas para evaluar la calidad de los análisis contextuales generados, incorporando validación por expertos humanos y comparación con análisis bibliométricos tradicionales.

Ampliación del Corpus de Evaluación: Extensión del dataset de evaluación para incluir documentos de campos altamente especializados, papers en idiomas distintos al inglés, y documentos con estructuras no convencionales (proceedings, tesis, reportes técnicos).

Optimización de Consultas Complejas: Investigación específica en estrategias para mejorar el rendimiento en consultas metodológicas y análisis comparativos, posiblemente mediante técnicas de descomposición de consultas o razonamiento multi-step.

8.2.2. Avances Arquitectónicos

Integración de Modelos Multimodales: Incorporación de capacidades de procesamiento de figuras, tablas y diagramas para análisis más comprehensivos del contenido científico, aprovechando los avances recientes en modelos de visión-lenguaje.

Sistemas de Memoria Episódica: Desarrollo de capacidades de memoria que permitan al sistema aprender de interacciones previas y mejorar progresivamente la

calidad de sus análisis contextuales.

Arquitecturas Federated Learning: Exploración de enfoques distribuidos que permitan el entrenamiento y mejora continua del sistema utilizando datos de múltiples instituciones mientras preserve la privacidad.

8.2.3. Aplicaciones Emergentes

Análisis de Evolución Conceptual: Desarrollo de capacidades para rastrear la evolución de conceptos científicos a través del tiempo mediante análisis longitudinal de contextos de citación.

Detección de Controversias Científicas: Implementación de algoritmos para identificar automáticamente debates o controversias en la literatura mediante análisis de polaridad y contexto de citas contradictorias.

Sistemas de Recomendación Contextual: Desarrollo de motores de recomendación que sugieran referencias relevantes no solo por similitud temática sino por contexto apropiado de citación según la sección y propósito argumentativo.

8.2.4. Investigación Fundamental

Teoría de Agentes Especializados: Investigación fundamental sobre principios de diseño para sistemas multi-agente especializados en análisis científico, incluyendo estrategias de comunicación inter-agente y mecanismos de coordinación.

Semántica de Citaciones: Desarrollo de marcos teóricos formales para la representación computacional de relaciones semánticas en citas científicas, trascendiendo las taxonomías actuales hacia modelos más expresivos.

Evaluación Benchmark Estándar: Contribución al desarrollo de benchmarks estándares para sistemas de análisis bibliométrico automatizado, facilitando la comparación objetiva entre diferentes enfoques metodológicos.

8.3. Reflexión Final

El desarrollo de este prototipo representa un paso significativo hacia la automatización inteligente del análisis bibliométrico, demostrando que es posible superar las limitaciones de las métricas cuantitativas tradicionales mediante la incorporación de análisis semántico contextualizado. Los resultados obtenidos validan la hipótesis

de que la integración sinérgica de técnicas de NLP modernas, arquitecturas RAG y sistemas multi-agente puede proporcionar insights cualitativos valiosos sobre las dinámicas de citación en la literatura científica.

Sin embargo, este trabajo también pone de manifiesto la complejidad inherente al análisis automatizado de contexto semántico y la necesidad de enfoques multidisciplinares que combinen avances técnicos con comprensión profunda de las prácticas científicas. El futuro del análisis bibliométrico automatizado dependerá no solo de mejoras tecnológicas, sino de la capacidad de desarrollar sistemas que respeten y preserven la riqueza semántica del discurso científico.

La investigación futura en este campo deberá balancear la búsqueda de automatización eficiente con la preservación de la complejidad conceptual que caracteriza a la comunicación científica, asegurando que las herramientas desarrolladas amplifiquen *rather than replace* la capacidad humana para comprender y evaluar el conocimiento científico en toda su riqueza contextual.

9. Anexos

9.1. Dataset evaluación

Convención: Los identificadores con prefijo E (E1–E5) corresponden a plantillas de Nivel Fácil, M (M1–M5) a Nivel Medio y H (H1–H5) a Nivel Difícil.

Cuadro 30: Plantillas de preguntas GPQA por nivel de dificultad

ID	Consulta	Objetivo
Nivel Fácil (5 preguntas)		
E1	¿En qué secciones se menciona la referencia [REF_NUM]?	Evaluar búsqueda básica por referencia específica
E2	¿Qué referencias tiene la sección de [SECTION_NAME]?	Evaluar extracción de referencias por sección
E3	¿Dónde aparece citado [REF_NUM]?	Evaluar localización de referencias
E4	Resume la sección de [SECTION_NAME]	Evaluar comprensión y síntesis de contenido
E5	¿Qué es [MAIN_CONCEPT]?	Evaluar definición de conceptos del dominio
Nivel Medio (5 preguntas)		
M1	¿Cómo se usa la referencia [REF_NUM] en la sección de [SECTION_NAME]?	Evaluar análisis contextual (referencia + sección)
M2	¿En qué secciones aparecen las referencias [REF_NUM1] y [REF_NUM2] relacionadas con [TOPIC]?	Evaluar búsqueda multi-referencia con filtro temático
M3	¿Cuáles son las principales herramientas de [DOMAIN_FIELD] mencionadas en este paper?	Evaluar extracción de herramientas específicas del dominio
M4	¿Qué referencias de la sección [SECTION_NAME] mencionan específicamente [SPECIFIC_TERM]?	Evaluar búsqueda con filtros múltiples (sección + término)
M5	¿Cuál es la metodología principal utilizada para [DOMAIN_TASK]?	Evaluar comprensión metodológica del dominio
Nivel Difícil (5 preguntas)		
H1	¿Cómo evalúan el rendimiento de los sistemas propuestos en esta investigación?	Evaluar análisis metodológico avanzado
H2	¿Qué dice la referencia [NONEXISTENT_REF] sobre [TOPIC] en este paper?	Evaluar detección de referencias inexistentes
H3	¿Qué metodologías de evaluación específicas para [DOMAIN_CONTEXT] se mencionan?	Evaluar búsqueda de metodologías específicas del dominio
H4	¿Qué herramientas se mencionan en [SECTION_NAME]?	Evaluar extracción específica por sección
H5	¿Cómo se comparan las diferentes aproximaciones según las métricas mencionadas en el paper?	Evaluar análisis comparativo y síntesis compleja

9.2. Prompt agente detección dominio

Eres un experto en clasificación de papers científicos.

Analiza el siguiente paper y determina su dominio académico principal.

{paper_context}

Basándote en el título y las secciones, identifica:

1. El CAMPO científico principal
2. Las TAREAS típicas de ese campo
3. Los CONCEPTOS clave de ese campo
4. TÉRMINOS específicos relevantes

IMPORTANTE: Responde ÚNICAMENTE con un JSON válido en este formato exacto:

```
{
  "DOMAIN_FIELD": "campo científico en español",
  "DOMAIN_TASK": "tarea típica del campo",
  "MAIN_CONCEPT": "concepto clave del campo",
  "DOMAIN_CONTEXT": "contexto del dominio",
  "SPECIFIC_TERM": "término específico para evaluación",
  "TOPIC": "tema general de investigación"
}
```

Ejemplos de campos: machine learning, procesamiento de lenguaje natural, visión computacional, biología molecular, química orgánica, economía, educación, psicología, medicina, ingeniería, física, matemáticas, ciencias sociales.

Respuesta JSON:

9.3. Sistema de evaluación.

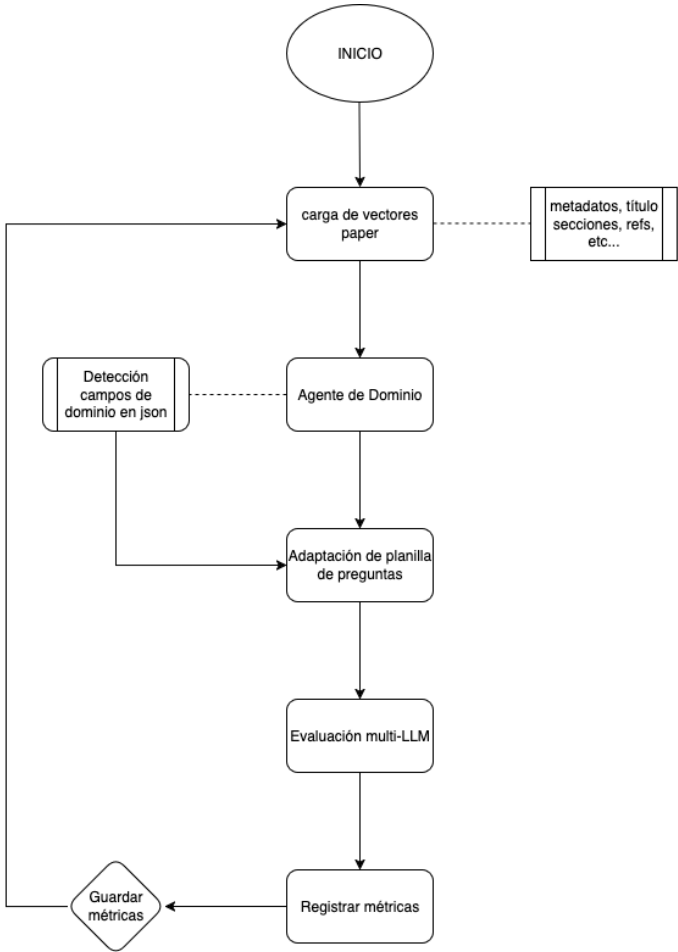


Figura 17: Sistema de evaluación.

Referencias

- [1] J. Yin and X. Li, “Personalized citation recommendation via convolutional neural networks,” in *APWeb/WAIM, Lecture Notes in Computer Science*, pp. 285–293, 2017.
- [2] M. Karim, M. M. S. Missen, M. Umer, S. Sadiq, A. Mohamed, and I. Ashraf, “Citation context analysis using combined feature embedding and deep convolutional neural network model,” *Applied Sciences*, vol. 12, no. 6, p. 3203, 2022.
- [3] L. Yang, Y. Zheng, X. Cai, H. Dai, D. Mu, L. Guo, and T. Dai, “A lstm based model for personalized context-aware citation recommendation,” *IEEE Access*, vol. 6, pp. 59618–59627, 2018.
- [4] T. Dai, L. Zhu, Y. Wang, and K. M. Carley, “Attentive stacked denoising auto-encoder with bi-lstm for personalized context-aware citation recommendation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 553–568, 2020.
- [5] K. Nishikawa and H. Koshiba, “Exploring the applicability of large language models to citation context analysis,” *Scientometrics*, vol. 129, pp. 6751–6777, 2024.
- [6] Y. Hu, Z. Lei, Z. Zhang, B. Pan, C. Ling, and L. Zhao, “GRAG: Graph retrieval-augmented generation,” in *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4145–4157, Association for Computational Linguistics, April 29–May 4 2025.
- [7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, *et al.*, “Language models are few-shot learners,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.
- [8] OpenAI, “Gpt-4o mini: Advancing cost-efficient intelligence,” 2024.
- [9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

- [10] H. Xu, X. Li, J. Tupayachi, J. J. Lian, and O. A. Omitaomu, “Automating bibliometric analysis with sentence transformers and retrieval-augmented generation (rag): A pilot study in semantic and contextual search for customized literature characterization for high-impact urban research,” in *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Advances in Urban-AI*, UrbanAI '24, (New York, NY, USA), p. 43–49, Association for Computing Machinery, 2024.
- [11] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, *et al.*, “Training compute-optimal large language models,” *arXiv preprint arXiv:2203.15556*, 2022.
- [12] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Joulin, E. Grave, and G. Lample, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.
- [13] M. Arslan, H. Ghanem, S. Munawar, and C. Cruz, “A survey on RAG with LLMs,” *Procedia Computer Science*, vol. 246, pp. 3781–3790, 2024.
- [14] N. S. Adhikari and S. Agarwal, “A comparative study of pdf parsing tools across diverse document categories,” *arXiv preprint arXiv:2410.09871*, 2024.
- [15] M. Arslan, H. Ghanem, S. Munawar, and C. Cruz, “A survey on rag with llms,” *Procedia Computer Science*, vol. 246, pp. 3781–3790, 2024. 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024).
- [16] M. G. Azar, M. Rowland, B. Piot, D. Guo, D. Calandriello, M. Valko, and R. Munos, “A general theoretical paradigm to understand learning from human preferences,” *Proceedings of the 41st International Conference on Machine Learning*, vol. 141, pp. 1831–1855, 2024.
- [17] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, “Direct preference optimization: Your language model is secretly a reward model,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [18] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, *et al.*, “Training a helpful and harmless assistant with reinforcement learning from human feedback,” *arXiv preprint arXiv:2204.05862*, 2022.

- [19] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, “Scaling laws for neural language models,” *arXiv preprint arXiv:2001.08361*, 2020.
- [20] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, *et al.*, “Training language models to follow instructions with human feedback,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 27730–27744, 2022.
- [21] Y. Ding, X. Liu, C. Guo, and B. Cronin, “The distribution of references across texts: Some implications for citation analysis,” *Journal of Informetrics*, vol. 7, pp. 583–592, July 2013.
- [22] M. Bertin, I. Atanassova, C. R. Sugimoto, and V. Lariviere, “The linguistic patterns and rhetorical structure of citation context: an approach using n-grams,” *Scientometrics*, vol. 109, pp. 1417–1434, Dec. 2016.
- [23] W. B. Cavnar and J. M. Trenkle, “N-Gram-Based Text Categorization,” p. 14.
- [24] I. Atanassova and M. Bertin, “Temporal Properties of Recurring In-text References,” *D-Lib Magazine*, vol. 22, Sept. 2016.
- [25] D. Zhao and A. Strotmann, “Weighted in-text citations and research impact patterns: A case study of library and information science: Weighted In-text Citations and Research Impact Patterns: A case study of Library and Information Science,” *Proceedings of the American Society for Information Science and Technology*, vol. 51, no. 1, pp. 1–5, 2014.
- [26] R. Jha, A.-A. Jbara, V. Qazvinian, and D. R. Radev, “Nlp-driven citation analysis for scientometrics,” *Natural Language Engineering*, vol. 23, pp. 93–130, Jan. 2017.
- [27] S. Teufel, A. Siddharthan, and D. Tidhar, “Automatic classification of citation function,” *In In Proc. of EMNLP-06, Sydney, Australia*, 2006.
- [28] I. Spiegel-Rosing, “Science studies: Bibliometric and content analysis,” *Social Studies of Science*, vol. 7, no. 1, pp. ii–ii, 1977.
- [29] H. Liu, “Sentiment Analysis of Citations Using Word2vec,” *arXiv:1704.00177 [cs]*, Apr. 2017. arXiv: 1704.00177.
- [30] A. Sadeghian and A. R. Sharafat, “Bag of words meets bags of popcorn,” *Stanford Working Papers*, 2015.

- [31] S. Wang and R. Koopman, “Clustering articles based on semantic similarity,” *Scientometrics*, vol. 111, pp. 1017–1031, May 2017. arXiv: 1702.04946.
- [32] A. Abu-Jbara, J. Ezra, and D. Radev, “Purpose and Polarity of Citation: Towards NLP-based Bibliometrics,” p. 11.
- [33] A. Khalid, F. A. Khan, M. Imran, M. Alharbi, M. Khan, A. Ahmad, and G. Jeon, “Reference terms identification of cited articles as topics from citation contexts,” *Computers & Electrical Engineering*, Mar. 2018.
- [34] M. Bertin and I. Atanassova, “Semantic Enrichment of Scientific Publications and Metadata: Citation Analysis Through Contextual and Cognitive Analysis,” *D-Lib Magazine*, vol. 18, July 2012.
- [35] S.-U. Hassan and P. Haddawy, “Analyzing knowledge flows of scientific literature through semantic links: a case study in the field of energy,” *Scientometrics*, vol. 103, pp. 33–46, Apr. 2015.
- [36] S.-U. Hassan, I. Safder, A. Akram, and F. Kamiran, “A novel machine-learning approach to measuring scientific knowledge flows using citation context analysis,” *Scientometrics*, vol. 116, pp. 973–996, Aug. 2018.
- [37] D. Duma, C. Sutton, and E. Klein, “Context Matters: Towards Extracting a Citation’s Context Using Linguistic Features,” in *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries - JCDL '16*, (Newark, New Jersey, USA), pp. 201–202, ACM Press, 2016.
- [38] Y. Ding, G. Zhang, T. Chambers, M. Song, X. Wang, and C. Zhai, “Content-based citation analysis: The next generation of citation analysis,” *Journal of the Association for Information Science and Technology*, vol. 65, pp. 1820–1833, Sept. 2014.
- [39] J. An, N. Kim, M.-Y. Kan, M. K. Chandrasekaran, and M. Song, “Exploring characteristics of highly cited authors according to citation location and content,” *Journal of the Association for Information Science and Technology*, vol. 68, pp. 1975–1988, Aug. 2017.
- [40] G. Zhang, Y. Ding, and S. Milojević, “Citation content analysis (CCA): A framework for syntactic and semantic analysis of citation content,” *Journal of the American Society for Information Science and Technology*, vol. 64, pp. 1490–1503, July 2013.

- [41] M. Bertin and I. Atanassova, “Extraction and Characterization of Citations in Scientific Papers,” in *Semantic Web Evaluation Challenge* (V. Presutti, M. Stankovic, E. Cambria, I. Cantador, A. Di Iorio, T. Di Noia, C. Lange, D. Reforgiato Recupero, and A. Tordai, eds.), vol. 475, pp. 120–126, Cham: Springer International Publishing, 2014.
- [42] S. S, E. E, and M. G.S, “Citation Semantic Based Approaches to Identify Article Quality,” in *Computer Science & Information Technology (CS & IT)*, pp. 411–420, Academy & Industry Research Collaboration Center (AIRCC), Sept. 2013.
- [43] S. Parinov, “Semantic Attributes for Citation Relationships: Creation and Visualization,” in *Metadata and Semantic Research* (E. Garoufallou, S. Virkus, R. Siatri, and D. Koutsomiha, eds.), vol. 755, pp. 286–299, Cham: Springer International Publishing, 2017.
- [44] M. Iqbal, S. Hossain, M. Mollah, and M. Islam, “Exploiting contextual word embedding for identification of important citations,” in *Proceedings of the International Conference on Natural Language Processing*, 2023.
- [45] T. Nomoto, “Neal: A neurally enhanced approach to linking citation and reference,” in *Proceedings of the Joint Workshop on Bibliometric-Enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*, pp. 168–174, 2016.
- [46] Z. Du, J. Tang, and Y. Ding, “Polar: Attention-based cnn for one-shot personalized article recommendation,” in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, vol. 11034 of *Lecture Notes in Computer Science*, pp. 425–441, 2018.
- [47] M. Karim, M. M. S. Missen, M. Umer, S. Sadiq, A. Mohamed, and I. Ashraf, “Citation context analysis using combined feature embedding and deep convolutional neural network model,” *Applied Sciences*, vol. 12, p. 3203, 2022.
- [48] S. Lahiri *et al.*, “Citeprompt: Using prompts to identify citation intent in scientific papers,” in *Proceedings of the Workshop on Scholarly Document Processing*, 2023.

- [49] D. Motrichenko, Y. Nedumov, and K. Skorniakov, “Bag of tricks for citation intent classification via scibert,” in *2021 Ivannikov Ispras Open Conference (ISPRAS)*, pp. 120–126, 2021.
- [50] R. H. C. Gowda, K. S. Raj, and A. K. Madasamy, “Citation intent classification using transformers,” in *2024 IEEE Students Conference on Engineering and Systems (SCES)*, pp. 1–6, IEEE, 2024.
- [51] X. Li, Y. Cao, Y. Ma, and A. Sun, “Long context vs. rag for llms: An evaluation and revisits,” 2024.
- [52] M. D. Skarlinski, S. Cox, J. M. Laurent, J. D. Braza, M. Hinks, M. J. Hammerling, M. Ponnappati, S. G. Rodrigues, and A. D. White, “Language agents achieve superhuman synthesis of scientific knowledge,” 2024.
- [53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023.
- [54] Y. Huang, J. Xu, J. Lai, Z. Jiang, T. Chen, Z. Li, Y. Yao, X. Ma, L. Yang, H. Chen, S. Li, and P. Zhao, “Advancing transformer architecture in long-context large language models: A comprehensive survey,” 2024.
- [55] A. Matarazzo and R. Torlone, “A survey on large language models with some insights on their capabilities and limitations,” 2025.
- [56] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus, “Emergent abilities of large language models,” 2022.
- [57] A. Lahiri, D. K. Sanyal, and I. Mukherjee, “CitePrompt: Using prompts to identify citation intent in scientific papers,” in *2023 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pp. 51–55, IEEE, 2023.
- [58] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman, “Gpqa: A graduate-level google-proof qa benchmark,” 2023.
- [59] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, “Llama: Open and efficient foundation language models,” 2023.
- [60] OpenAI, “Introducing gpt-4.1 in the api,” 2025.

- [61] Google DeepMind, “Gemini 2.0: Our new ai model for the agentic era,” 2024.
- [62] Google Cloud, “Gemini 2.0 flash — generative ai on vertex ai with 1 m token context window,” 2025. Accessed: 2025-08-11.
- [63] Helicone, “Gemini 2.0 flash explained: Building more reliable applications,” 2024. Accessed: 2025-08-11.
- [64] LLM-Stats, “Gpt-4o vs gemini 2.0 flash — model benchmark comparison,” 2025. Accessed: 2025-08-11.
- [65] Beebom, “Google unveils gemini 2.0 and ‘deep research’ — advanced ai capabilities across multiple domains,” 2024. Accessed: 2025-08-11.
- [66] D. AI, “Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning,” *arXiv preprint arXiv:2501.12948*, 2025.
- [67] DeepInfra, “Deepseek r1 distill qwen 32b – demo,” 2025. Accessed: 2025-08-11.
- [68] LLM-Stats, “Deepseek r1 distill qwen 32b vs deepseek-r1: benchmark comparison,” 2025. Accessed: 2025-08-11.
- [69] Mistral AI, “Minstral-8b-instruct-2410,” 2024. Accessed: 2025-08-11.
- [70] Mistral AI, “Model card for minstral-8b-instruct-2410,” 2025. Accessed: 2025-08-11.
- [71] Meta AI, “Introducing meta llama 3: The most capable openly available llm to date,” 2024.
- [72] LLM-Stats, “Llama 3.3 70b instruct vs qwen2.5 32b instruct,” 2024. Accessed: 2025-08-11.
- [73] DataCamp, “What is meta’s llama 3.3 70b? how it works, use cases & more,” 2024. Accessed: 2025-08-11.
- [74] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, “Deep reinforcement learning from human preferences,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [75] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.

- [76] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, *et al.*, “Learning to summarize with human feedback,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 3008–3021, 2020.
- [77] Z. Yuan, H. Yuan, C. Li, G. Dong, K. Lu, C. Tan, *et al.*, “Advancing llm reasoning generalists with preference trees,” *arXiv preprint arXiv:2404.02078*, 2024.
- [78] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.
- [79] J. Gou, B. Yu, S. J. Maybank, and D. Tao, “Knowledge distillation: A survey,” *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.
- [80] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [81] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter,” *arXiv preprint arXiv:1910.01108*, 2019.
- [82] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, *et al.*, “Emergent abilities of large language models,” *Transactions on Machine Learning Research*, 2022.
- [83] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, *et al.*, “Retrieval-augmented generation for knowledge-intensive NLP tasks,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.
- [84] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, “Retrieval-augmented generation for large language models: A survey,” 2024.
- [85] M. Wooldridge and N. R. Jennings, “Intelligent agents: theory and practice,” *The Knowledge Engineering Review*, vol. 10, no. 2, p. 115–152, 1995.
- [86] P. Stone and M. Veloso, “Multiagent systems: A survey from a machine learning perspective,” *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, 2000.
- [87] py-pdf community, “Benchmarking pdf libraries.” <https://github.com/py-pdf/benchmarks>, 2023.

- [88] J. Dagdelen, A. Dunn, S. Lee, N. Walker, A. S. Rosen, G. Ceder, K. A. Persson, and A. Jain, “Large language models for generative information extraction: a survey,” *Frontiers of Computer Science*, 2024.
- [89] M. Polat and J. Dagdelen, “Extracting accurate materials data from research papers with conversational language models and prompt engineering,” *Nature Communications*, vol. 15, p. 1418, 2024.
- [90] F. Polat, I. Tiddi, and P. Groth, “Testing prompt engineering methods for knowledge extraction from text,” *Semantic Web*, 2025.
- [91] A. Dunn *et al.*, “Structured information extraction from scientific text with large language models,” *Nature Communications*, vol. 15, p. 1418, 2024.
- [92] Z. Zhong, H. Liu, X. Cui, X. Zhang, and Z. Qin, “Mix-of-granularity: Optimize the chunking granularity for retrieval-augmented generation,” 2025.
- [93] F22 Labs, “7 chunking strategies in rag you need to know.” <https://www.f22labs.com/blogs/7-chunking-strategies-in-rag-you-need-to-know>, 2024.
- [94] C. Research, “Evaluating chunking.” <https://research.trychroma.com/evaluating-chunking>, 2023.
- [95] OpenAI, “Text embedding models: text-embedding-ada-002.” <https://platform.openai.com/docs/models/text-embedding-ada-002>, 2023.
- [96] J. Johnson, M. Douze, and H. Jégou, “Faiss: A library for efficient similarity search,” *Engineering at Meta*, March 2017.
- [97] LangChain Team, “Langgraph: Multi-agent workflows,” *LangChain Blog*, February 2024. Framework for building multi-agent systems with state machine control.
- [98] Y. Qin, X. Zhang, Y. Chen, Z. Yang, R. Liu, Q. Wang, and R. Zhao, “Stateflow: Enhancing llm task-solving through state-driven workflows,” *arXiv preprint arXiv:2403.11322*, 2024.