Experimental Evaluation of Retrieval Strategies in RAG Architectures Applied to the Judicial Domain

Julio César Bossero [0000-0002-2498-9103]

Universidad Nacional de La Matanza.
Departamento de Ingeniería e Investigación Tecnológicas
jbossero@unlam.edu.ar

Abstract. This paper presents the design, implementation, and experimental evaluation of a Retrieval-Augmented Generation (RAG) system applied to the criminal judicial domain. The main objective is to analyze the performance of the semantic retrieval component in a fully local, secure, and reproducible environment, using a real corpus of 2,592 judicial rulings comprising 46,382 vectorized text fragments.

The implemented pipeline integrates textual segmentation through Recursive Character Text Splitter, dense embeddings generated with all-mpnet-base-v2, persistent vector storage in ChromaDB, and context-conditioned generation using a locally executed LLM. The evaluation focused exclusively on the retrieval module, employing standard Information Retrieval metrics such as Precision@5 and Mean Reciprocal Rank (MRR), complemented by cosine similarity analysis and expert-based manual relevance assessment.

Results indicate strong thematic consistency in retrieval but reveal limitations in identifying specific legal reasoning, particularly under linguistic reformulations of queries. Future improvements include domain-specific legal embeddings, cross-encoder re-ranking techniques, and optimized chunking strategies. The study provides empirical evidence on structural challenges in legal RAG systems and establishes a reproducible baseline for further research.

Keywords: Retrieval-Augmented Generation; Semantic Retrieval; Legal Informatics.

Evaluación Experimental de Estrategias de Retrieval en Arquitecturas RAG Aplicadas al Dominio Judicial

Resumen. Este trabajo presenta el diseño, implementación y evaluación experimental de un sistema Retrieval-Augmented Generation (RAG) aplicado al dominio judicial penal. El objetivo principal fue analizar el desempeño del componente de recuperación semántica en un entorno completamente local, seguro y reproducible, utilizando un corpus real compuesto por 2.592 sentencias judiciales y 46.382 fragmentos vectorizados.

El pipeline implementado integra segmentación textual mediante *Recursive Character Text Splitter*, embeddings densos generados con *all-mpnet-base-v2*, almacenamiento en ChromaDB y generación contextual con un modelo LLM ejecutado localmente. La evaluación se centró exclusivamente en el módulo de *retrieval*, empleando métricas estándar de Information Retrieval como Precision@5 y Mean Reciprocal Rank (MRR), complementadas con análisis de similitud coseno y evaluación manual experta de relevancia.

Los resultados muestran alta coherencia temática en la recuperación, pero limitaciones en la discriminación de fundamentos jurídicos específicos, especialmente ante variaciones lingüísticas en las consultas. Se identifican como líneas de mejora la incorporación de embeddings jurídicos especializados, técnicas de *re-ranking* con *cross-encoders* y optimización del esquema de *chunking*.

El estudio aporta evidencia empírica sobre los desafíos estructurales de los sistemas RAG jurídicos y establece un baseline reproducible para investigaciones futuras.

Palabras clave: Retrieval-Augmented Generation; Recuperación Semántica; Informática Jurídica.

1 Introducción

Los Modelos de Lenguaje de Gran Escala (*Large Language Models,* LLMs) han demostrado avances significativos en múltiples tareas de Procesamiento del Lenguaje Natural (*Natural Language Processing,* NLP). No obstante, su aplicación en dominios especializados, como el jurídico, presenta desafíos particulares derivados de la complejidad terminológica, la estructura jerárquica de los textos normativos y la fuerte dependencia contextual entre disposiciones legales (Ferraris et al., 2024).

En este escenario, las arquitecturas Generación Aumentada por Recuperación (*Retrieval-Augmented Generation,* RAG) han surgido como un enfoque eficaz para complementar a los modelos generativos, es decir, aquellos capaces de producir texto a partir de patrones aprendidos. Estas arquitecturas incorporan mecanismos de recuperación de información desde bases documentales externas, lo que permite que el modelo genere sus respuestas apoyándose en evidencia relevante del corpus y, en consecuencia, mejorar tanto la precisión como la contextualización de la salida producida (Naveed et al., 2024).

Las arquitecturas RAG integran dos componentes interdependientes. En primer lugar, un módulo de recuperación semántica que representa tanto la consulta como los fragmentos del corpus mediante vectores densos, es decir, representaciones numéricas compactas que codifican el significado del texto en un espacio de alta dimensionalidad. Estas representaciones permiten medir similitud semántica más allá de coincidencias literales, de modo que fragmentos conceptualmente cercanos queden ubicados próximos entre sí dentro del índice vectorial. En segundo lugar, un modelo de lenguaje generativo utiliza los fragmentos recuperados como contexto para elaborar la respuesta final, orientando la generación hacia información verificable y relevante del corpus.

En este proceso, la consulta se transforma en un embedding que resume su contenido semántico; ese vector se compara con el índice vectorial construcción a partir de los documentos segmentados, recuperándose los fragmentos más similares. Dichos fragmentos se incorporan como parte del contexto que condiciona la salida del modelo generativo, garantizando que la respuesta producida no dependa únicamente del conocimiento estadístico aprendido, sino también del contenido específico presente en los documentos relevantes (Ahmed Kaleem, 2025).

Diversos estudios han señalado que el desempeño global del sistema depende no solo del modelo generativo, sino también del diseño del índice documental, la calidad de las representaciones vectoriales y el mecanismo de recuperación utilizado (Izacard et al., 2022). En este marco, la segmentación documental (en inglés chunking) adquiere un rol central, ya que define la unidad semántica representada en el espacio vectorial y condiciona directamente la precisión del ranking y la calidad del contexto provisto al modelo generativo.

En el ámbito de la justicia, el problema del *chunking* resulta especialmente crítico. Una segmentación inadecuada puede fragmentar disposiciones normativas, romper la coherencia semántica o perder referencias cruzadas relevantes, afectando la calidad del *retrieval* (Ferraris et al., 2024). Estudios recientes que analizan técnicas automáticas de segmentación —incluyendo *simple text splitting, recursive splitting* basado en expresiones regulares y *semantic chunking*— muestran que ninguna estrategia logra consistentemente altos niveles de relevancia semántica a nivel de fragmento individual, incluso en contextos regulatorios estructurados como el *General Data Protection Regulation* (GDPR) (Ferraris et al., 2024). Estos resultados evidencian que la granularidad de los fragmentos y la preservación de la estructura jurídica son variables determinantes en sistemas RAG aplicados al ámbito legal.

A pesar de la creciente adopción de arquitecturas RAG en aplicaciones jurídicas, aún existe una evaluación experimental limitada sobre el impacto específico de los parámetros de segmentación documental —como el tamaño del fragmento (*chunk size*), que determina cuánta información se incluye en cada segmento de texto, y el solapamiento (*overlap*), que indica cuánto contenido se repite entre segmentos consecutivos para evitar pérdida de contexto— en el desempeño del proceso de recuperación.

Este trabajo presenta una evaluación experimental del desempeño del módulo de recuperación dentro de un *pipeline* RAG, entendido como una secuencia organizada de etapas —como la segmentación documental, la generación de *embeddings*, la búsqueda vectorial y la generación final de la respuesta— que operan de manera encadenada para producir resultados coherentes y contextualizados en el contexto judicial penal. El estudio se realiza sobre un corpus real de sentencias judiciales, utilizando una arquitectura RAG estándar basada en segmentación textual de longitud fija, embeddings generalistas y recuperación por similitud coseno.

El objetivo es analizar el comportamiento del sistema de retrieval en un entorno jurídico real, identificando limitaciones asociadas a la granularidad de segmentación, la representación vectorial y la sensibilidad lingüística de las consultas. De este modo, el estudio aporta evidencia empírica orientada a la configuración reproducible de siste-

mas RAG en contextos jurídicos y contribuye a una mejor comprensión de los desafíos estructurales que presentan estas arquitecturas en dominios especializados.

2 Trabajos Relacionados

El rendimiento de las arquitecturas RAG depende en gran medida de la calidad del componente de recuperación de información, donde las estrategias de segmentación e indexación influyen directamente en métricas como Precision@k y MRR. Diversos trabajos recientes analizan estos factores en dominios especializados.

En relación con las estrategias de segmentación, Günther et al. (2024) proponen el enfoque de *late chunking* para preservar dependencias semánticas entre fragmentos, mientras que Amiri y Bocklitz (2025) señalan la existencia de un *trade-off* entre coherencia contextual y capacidad de discriminación fina según el tamaño de los fragmentos utilizados.

Desde una perspectiva teórica, Smalle et al. (2016) aportan fundamentos cognitivos sobre cómo la granularidad textual influye en los procesos de discriminación semántica. En el ámbito jurídico, Ferraris et al. (2024) advierten que los métodos de segmentación de longitud fija pueden mezclar unidades argumentativas heterogéneas, afectando la precisión de los sistemas de recuperación.

En cuanto a la evaluación experimental, Reuter et al. (2025) y Pipitone y Houir Alami (2024) destacan la importancia de marcos de evaluación sistemáticos y de la disponibilidad de benchmarks especializados como LegalBench-RAG para analizar el rendimiento de sistemas RAG en tareas jurídicas.

A pesar de estos avances, persiste una brecha en evaluaciones empíricas de pipelines RAG completos aplicados a corpus judiciales reales, especialmente en entornos locales orientados a la reproducibilidad experimental. El presente trabajo aborda esta vacancia mediante una evaluación del comportamiento del componente de retrieval en una arquitectura RAG aplicada a un corpus judicial real.

.

3 Metodología Experimental

3.1 Diseño del estudio

A diferencia de propuestas de nuevos *benchmarks*, este estudio adopta un enfoque estrictamente experimental aplicado a un corpus judicial real. El objetivo es evaluar el comportamiento de un *pipeline* RAG bajo una configuración base (*baseline*) estándar, sin técnicas adicionales de optimización (como *re-ranking* o *fine-tuning*) para aislar el desempeño del módulo de recuperación en condiciones controladas. El *baseline* se define por:

- **Segmentación:** Longitud fija de texto.
- **Representación:** Embeddings mediante un modelo generalista.
- **Recuperación:** Similitud coseno sobre un índice vectorial.

- **Generación:** Modelo de lenguaje ejecutado en entorno local.

El diseño busca analizar la robustez semántica del sistema frente a variaciones lingüísticas, utilizando consultas equivalentes que reformulan un mismo problema jurídico para evaluar la estabilidad ante el parafraseo. Así, la investigación se sitúa en la intersección de tres dimensiones:

1. Evaluación experimental del *retrieval* en RAG.
2. Impacto de la segmentación en el ámbito jurídico.
3. Estabilidad del sistema ante reformulaciones de la consulta.

Este enfoque aporta evidencia empírica en un entorno local y reproducible, esencial para la confiabilidad de la IA en el análisis documental judicial.

3.2 Corpus y Preparación de Datos

El estudio se llevó a cabo sobre un corpus compuesto por 2592 sentencias judiciales en formato texto plano (.txt), correspondientes a resoluciones judiciales del fuero penal. Los documentos fueron previamente anonimizados y estructurados en archivos individuales.

Con el objetivo de habilitar la recuperación semántica, cada documento fue segmentado utilizando el algoritmo *RecursiveCharacterTextSplitter*, una estrategia de segmentación ampliamente empleada en pipelines RAG basados en embeddings. Este método divide el texto en fragmentos de longitud controlada preservando, en la medida de lo posible, la continuidad semántica entre unidades textuales.

La elección de tamaños de fragmento de aproximadamente 1000 caracteres con solapamientos parciales constituye una práctica común en sistemas de recuperación semántica basados en embeddings, ya que permite preservar coherencia contextual sin diluir la discriminación semántica entre fragmentos adyacentes (Izacard et al., 2022; Ferraris et al., 2024).

- *chunk_size* = 1000 caracteres
- overlap = 200 caracteres

Esta configuración permite mantener continuidad semántica entre fragmentos adyacentes, reduciendo la fragmentación de unidades argumentativas propias del razonamiento jurídico.

El proceso generó un total de 46.382 fragmentos (*chunks*), los cuales constituyen la unidad básica de recuperación del sistema.

3.3 Representación Vectorial e Indexación

Cada fragmento fue transformado en un embedding denso mediante el modelo *all-mpnet-base-v2* de la biblioteca *Sentence Transformers*, ampliamente utilizado en tareas de búsqueda semántica y recuperación basada en embeddings.

Los vectores resultantes (dimensión 768) fueron almacenados en una base de datos vectorial *ChromaDB*, configurada en modo persistente para garantizar reproducibilidad del experimento. A cada fragmento se le incorporó metadata asociada al documento original, permitiendo trazabilidad en las respuestas generadas.

La arquitectura implementada corresponde a un pipeline RAG clásico compuesto por:

1. Generación de Embedding de la consulta.
2. Búsqueda por similitud coseno en el espacio vectorial.
3. Recuperación de los k fragmentos más similares (k = 5).
4. Generación condicionada por contexto mediante un modelo LLM ejecutado localmente (Mistral vía Ollama).

3.4 Configuración de Evaluación

Con el fin de clarificar el flujo de procesamiento del sistema implementado, el Algoritmo 1 presenta el pseudocódigo del pipeline experimental utilizado para la construcción del índice vectorial y la ejecución de consultas dentro de la arquitectura RAG.

Algoritmo 1: Pipeline RAG Jurídico Experimental

```
procedure BuildVectorIndex(corpus)
    for each document in corpus do
        chunks ← RecursiveSplit(document, size=1000, overlap=200)
        for each chunk in chunks do
            embedding ← Embed(chunk, model=all-mpnet-base-v2)
            Store(embedding, metadata=document_id)
    end for
end procedure
procedure QueryRAG(query, k)
    q_embedding ← Embed(query, model=all-mpnet-base-v2)
    results ← VectorSearch(q_embedding,
                           metric=cosine_similarity,top=k)
    return results
end procedure
```

Este procedimiento permite separar claramente las etapas de indexación y consulta, siguiendo el esquema clásico de los sistemas de recuperación basados en embeddings utilizados en arquitecturas RAG.

3.5 Métricas

Se emplearon métricas estándar de *Information Retrieval*:

- **Precision@k**: mide la calidad del sistema de recuperación enfocándose en la cantidad de resultados útiles dentro de los primeros *k* elementos entregados.

$$\mathbf{P@}k = \text{Número de fragmentos relevantes en el Top-}k \;/\; k \qquad (1)$$

En los experimentos realizados en este trabajo se utilizó k = 5, evaluando los cinco primeros fragmentos recuperados para cada consulta.

- **Mean Reciprocal Rank (MRR)**: inverso de la posición del primer fragmento relevante.

$$\mathbf{MRR = (1 / |Q|)\ \Sigma\ (1 / Rank_i)} \quad \textbf{(2)}$$

Donde:

- $|Q|$ es el número total de consultas realizadas.
- $Rank_i$ es la posición del primer fragmento relevante para la consulta i.

Si el fragmento clave aparece en el puesto 1, sumas 1. Si aparece en el puesto 2, sumas 0.5. Un MRR cercano a 1 indica un motor de búsqueda altamente eficiente.

3.6 Procedimiento Experimental

El procedimiento experimental se estructuró en las siguientes etapas:

1. Construcción del índice vectorial a partir del corpus judicial previamente segmentado.
2. Validación del espacio de embeddings, verificando la dimensionalidad de los vectores generados, la consistencia en la distribución de valores y el tamaño del índice vectorial persistido en disco.
3. Ejecución de un conjunto de consultas de evaluación previamente definidas.
4. Evaluación manual de relevancia de los fragmentos recuperados por el sistema.
5. Cálculo automático de métricas de desempeño del módulo de recuperación.
6. Análisis comparativo del comportamiento del sistema frente a distintas reformulaciones semánticamente equivalentes de las consultas.

Todos los experimentos fueron ejecutados en un entorno completamente local, utilizando recursos computacionales propios, lo que permitió garantizar tanto la privacidad de los datos judiciales como la reproducibilidad del procedimiento experimental.

El flujo experimental completo fue automatizado mediante scripts en Python desarrollados ad hoc, permitiendo reproducir de manera sistemática las etapas de indexación, consulta y evaluación.

4 Evaluación y Resultados

La evaluación del sistema RAG jurídico se centró en el desempeño del módulo de recuperación semántica, dado que en estas arquitecturas la calidad de la respuesta generada depende directamente de la pertinencia de los fragmentos recuperados (Reuter et al., 2025; Pipitone & Houir Alami, 2024).

Para la medición se emplearon métricas estándar del campo de Information Retrieval, particularmente Precision@k y Mean Reciprocal Rank (MRR), ampliamente utilizadas en la evaluación de sistemas de recuperación (Manning, Raghavan & Schütze, 2008).

Dichas métricas fueron registradas mediante un script experimental propio (evaluacion_retrieval.py), el cual permitió ejecutar consultas reales sobre el corpus judicial, identificar manualmente los fragmentos relevantes recuperados y calcular automáticamente los indicadores correspondientes.

4.1 Diseño Experimental

El pipeline experimental se implementó en Python 3 mediante scripts ad hoc para construir el índice vectorial, ejecutar consultas y calcular métricas de desempeño. El experimento utilizó un corpus de 2.592 sentencias judiciales en texto plano, fragmentadas en 46.382 unidades mediante RecursiveCharacterTextSplitter (LangChain Community, 2024) con chunk_size = 1000 y overlap = 200. Estos valores se eligieron para equilibrar la cohesión semántica y la compatibilidad con el límite de tokens del modelo de embeddings, evitando tanto la fragmentación excesiva de unidades argumentativas como la pérdida de discriminación semántica fina.
Cada fragmento se convirtió en un embedding usando el modelo all-mpnet-base-v2, basado en Sentence-BERT (Reimers & Gurevych, 2019), almacenándose los vectores en ChromaDB en modo persistente (Chroma Team, 2023). El modelo fue seleccionado por su buen rendimiento en similitud semántica dentro de Sentence-Transformers (s. f.), su desempeño competitivo en español y su reproducibilidad en entornos locales. Aunque no está especializado en español ni en dominio jurídico, se considera adecuado para el estudio, dejando abierta la posibilidad de usar modelos específicos en trabajos futuros. La recuperación se realizó con $k = 5$ usando similitud coseno. La relevancia de los fragmentos fue determinada mediante etiquetado manual binario (relevante / no relevante), basado en la presencia de fundamentos normativos o criterios decisorios vinculados a la consulta. Esto permitió construir un ground truth para calcular Precision@5 y MRR, usando un conjunto de $|Q| = 10$ consultas. Toda la ejecución se llevó a cabo en un entorno local con Ollama (Ollama Team, 2023) y el modelo generativo Mistral 7B (Jiang et al., 2023), garantizando privacidad, control de datos y reproducibilidad.

Tabla 1. Configuración experimental

Variable	Valor
Corpus judicial	2.592 sentencias
Fragmentación	chunk_size=1000, overlap=200
Total de vectores	46.382
Embedding	all-mpnet-base-v2
Base vectorial	ChromaDB (persistente)
Recuperación	k=5, similitud coseno
LLM generativo	Mistral 7B (Ollama, ejecución local)
Protocolo	Evaluación experta + métricas IR

4.2 Consultas Evaluadas

Para evaluar la robustez semántica y la sensibilidad al vocabulario jurídico del sistema, se definieron diez consultas representativas (|Q| = 10). Todas se centran en el instituto de la libertad condicional, pero presentan variaciones léxicas y conceptuales estratégicas para testear los límites del modelo de *embeddings*:

Tabla 2. Consultas utilizadas en la evaluación

Id	Consulta
Q1	¿Por qué se rechaza la libertad condicional?
Q2	Fundamentos para denegar la libertad condicional
Q3	Motivos por los cuales se niega la libertad condicional
Q4	¿En qué casos se concede la libertad condicional?
Q5	Requisitos legales para otorgar la libertad condicional
Q6	¿Qué condiciones exige la ley para acceder a la libertad condicional?
Q7	¿Cuándo corresponde otorgar la libertad condicional?
Q8	Criterios judiciales para evaluar la libertad condicional
Q9	¿Qué factores analiza el juez al decidir la libertad condicional?
Q10	Evaluación del comportamiento del condenado para otorgar libertad condicional

Estas variaciones permiten medir tres dimensiones críticas del módulo de *retrieval*:

- **Recuperación ante terminología técnica:** Uso de verbos específicos del ámbito judicial(p. ej., "denegar", "conceder").
- **Recuperación ante consultas conceptuales:** Capacidad de identificar requisitos o condiciones abstractas (p. ej., "requisitos legales", "criterios judiciales").
- **Robustez frente a la reformulación lingüística:** Estabilidad de las métricas de **Precision@k** y **MRR** (Manning et al., 2008) ante el parafraseo de una misma intención de búsqueda.

4.3 Resultados Cuantitativos

Los resultados obtenidos reflejan el desempeño del módulo de recuperación frente a la variabilidad lingüística de las consultas. La Tabla 3 detalla las métricas para cada una de las diez consultas evaluadas..

Tabla 3. Desempeño del módulo de *retrieval* por consulta ($k=5$) por consulta

Consulta	Precision@5	MRR	similitud coseno
¿Por qué se rechaza la libertad condicional?	0.20	1.00	0.712
Fundamentos para denegar la libertad condicional	0.40	1.00	0.772
Motivos por los cuales se niega la libertad condicional	0.20	1.00	0.762

¿En qué casos se concede la libertad condicional?	0.20	0.50	0.743
Requisitos legales para otorgar la libertad condicional	0.00	0.00	0.773
¿Qué condiciones exige la ley para acceder...	0.20	0.33	0.748
¿Cuándo corresponde otorgar la libertad...	0.20	0.50	0.765
Criterios judiciales para evaluar la libertad...	0.20	0.33	0.776
¿Qué factores analiza el juez al decidir la...	0.20	0.25	0.776
Evaluación del comportamiento del condenado...	0.20	0.50	0.778

Como se observa, la similitud coseno presenta una estabilidad notable en todas las consultas (oscilando entre 0.71 y 0.77), lo que indica que el modelo de embeddings all-mpnet-base-v2 logra agrupar los fragmentos en una región semántica coherente, incluso cuando la precisión de la respuesta relevante es baja (Manning et al., 2008).

A continuación, se resumen los estadísticos generales del experimento:

Tabla 4. Estadísticos generales

Métrica	Promedio	Desvío estándar
Precision@5	**0.200**	**0.094**
MRR	**0.541**	**0.349**
Similitud coseno	**0.758**	**0.021**

4.4 Análisis Cualitativo

A continuación, se presenta una visualización combinada que integra las métricas Precision@5, Mean Reciprocal Rank (MRR) y similitud coseno promedio, con el objetivo de analizar el desempeño del módulo de recuperación (Figura 1).

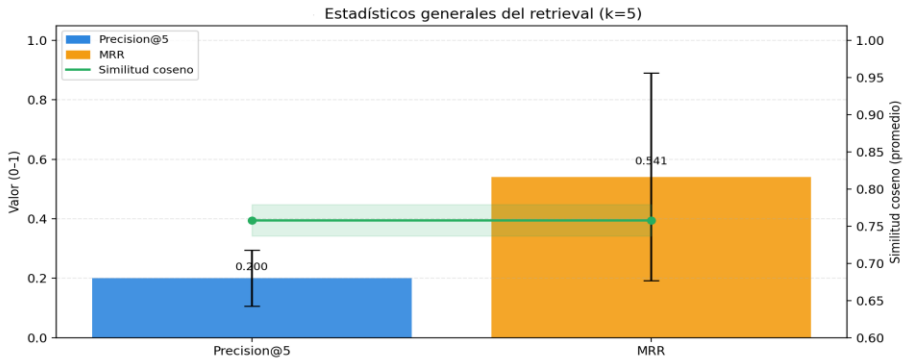

Figura 1. Desempeño del módulo de *retrieval* por consulta (*k = 5*).

El análisis del gráfico y de los registros experimentales permite identificar varios fenómenos relevantes.

- Sensibilidad a la terminología técnica. Se observan picos de desempeño en consultas que emplean terminología jurídica frecuente en el corpus, particularmente

verbos decisorios como *denegar*. En estos casos se alcanzan valores elevados de MRR (hasta 1,0) y mejoras en Precision@5, lo que sugiere que el espacio vectorial captura con mayor eficacia patrones terminológicos recurrentes en las sentencias.
• Degradación ante consultas conceptuales. Cuando la consulta se formula en términos más abstractos —por ejemplo, centrados en *requisitos* o *condiciones generales*— se observa una disminución en las métricas de recuperación, incluyendo casos de Precision@5 = 0, lo que evidencia la dificultad del modelo para identificar fundamentos jurídicos implícitos sin coincidencia terminológica.
• Coherencia semántica estable frente a precisión variable. Los resultados muestran una similitud coseno promedio ≈ 0,76 ($\sigma = 0{,}02$) entre los fragmentos recuperados, indicando una coherencia temática estable. Sin embargo, esta consistencia no garantiza que las primeras posiciones del ranking correspondan a los fundamentos jurídicos más relevantes.
• Confusión de funciones argumentativas. En algunos casos se recuperaron fragmentos introductorios o doctrinales en lugar de segmentos decisorios. Esto refleja una limitación conocida de los embeddings generalistas, que capturan similitud temática pero no siempre discriminan la función argumentativa dentro del texto jurídico (Reimers & Gurevych, 2019).
• Impacto del chunking. El uso de fragmentos de aproximadamente 1000 caracteres tiende a combinar definiciones, normas y hechos en una misma unidad textual, lo que puede reducir la discriminación fina del modelo al diluir la función argumentativa específica.

En conjunto, estos resultados son consistentes con las limitaciones estructurales de los bi-encoders, que calculan representaciones independientes para consultas y documentos, lo que puede dificultar la captura de relaciones semánticas complejas o dependencias contextuales (Manning et al., 2008).

Para mitigar estas limitaciones, la literatura recomienda incorporar una etapa adicional de re-ranking mediante cross-encoders, que evalúan conjuntamente la pareja consulta-documento y mejoran la precisión en las primeras posiciones del ranking (Nogueira & Cho, 2019; Khattab & Zaharia, 2020; Sentence-Transformers Documentation, 2024).

5 Discusión

Los resultados indican que el pipeline RAG jurídico presenta una estructura semántica estable, aunque con limitaciones en la discriminación argumentativa fina. La alta similitud coseno sugiere que los fragmentos recuperados pertenecen al mismo ámbito temático; sin embargo, esta estabilidad no garantiza precisión en la identificación de fundamentos jurídicos específicos, lo que se refleja en valores moderados de Precision@5 y MRR.

Asimismo, el sistema muestra sensibilidad a la formulación lingüística de la consulta, con mejor desempeño ante terminología propia del discurso judicial. El esque-

ma de segmentación adoptado tiende además a integrar múltiples funciones discursivas en un mismo fragmento, reduciendo la especificidad argumentativa del ranking.

En conjunto, el sistema constituye un baseline reproducible para exploración temática en el contexto jurídico, aunque aún no garantiza recuperación decisoria de alta precisión.

6 Conclusiones

El presente trabajo demuestra la viabilidad de implementar un sistema RAG jurídico completamente local, seguro y reproducible para el procesamiento de documentación judicial. La arquitectura modular adoptada constituye un marco metodológico sólido para aplicaciones en informática jurídica sin dependencia de servicios en la nube.

Desde el punto de vista científico, los resultados validan la consistencia estructural del espacio vectorial construido y confirman que un enfoque basado en embeddings generalistas permite exploración temática estable del corpus. No obstante, la evidencia experimental muestra que alcanzar niveles de precisión adecuados para identificación de fundamentos jurídicos específicos requiere técnicas adicionales de especialización y refinamiento del ranking.

En este sentido, el trabajo no solo confirma la factibilidad técnica del enfoque, sino que delimita con claridad los desafíos pendientes en sistemas RAG jurídicos locales, estableciendo un punto de partida sólido para desarrollos futuros orientados a mayor discriminación argumentativa y menor sensibilidad léxica.

7 Trabajo Futuro

Los resultados obtenidos permiten identificar líneas de mejora orientadas a incrementar la precisión y robustez del sistema RAG jurídico. Si bien la arquitectura desarrollada constituye un baseline reproducible, el análisis experimental evidenció limitaciones principalmente en la representación semántica, el ranking y la segmentación documental.

Una restricción relevante proviene del uso de un embedding generalista (all-mpnet-base-v2), adecuado para similitud temática global pero limitado para discriminar funciones argumentativas del discurso judicial. La incorporación de modelos entrenados en corpus jurídicos, como variantes de Legal-BERT o modelos ajustados para español, podría mejorar la identificación de fundamentos normativos y criterios decisorios. Asimismo, la integración de un módulo de re-ranking mediante cross-encoders permitiría evaluar conjuntamente consulta y fragmento, mejorando métricas como Precision@5 y MRR y alineando el sistema con arquitecturas RAG más avanzadas.

Finalmente, el esquema actual de fragmentación, basado en bloques de aproximadamente 1000 caracteres con solapamiento, genera unidades textuales heterogéneas que pueden combinar definiciones, hechos y fundamentos jurídicos dentro de un mismo fragmento, reduciendo la especificidad argumentativa del proceso de recuperación

8 Agradecimientos

Expreso mi especial agradecimiento al Ing. Osvaldo Spositto, director de los proyectos ProInce en los que participo, por su orientación académica y acompañamiento constante, fundamentales para el desarrollo del presente trabajo.
Agradezco asimismo al Departamento de Ingeniería e Investigaciones Tecnológicas (DIIT) por promover iniciativas de investigación interdisciplinaria y brindar el marco institucional para la articulación entre ciencia de datos, derecho e inteligencia artificial.
Extiendo mi reconocimiento a la Universidad Nacional de La Matanza (UNLaM) por fomentar un entorno académico orientado a la innovación y la experimentación con tecnologías emergentes.
Finalmente, agradezco al Juzgado de Ejecución N.º 2 del Departamento Judicial Morón por facilitar el acceso a los expedientes que permitieron conformar el corpus jurídico utilizado en esta investigación.

Nota de integridad y uso de IA.

Durante la preparación de este trabajo utilicé M365 Copilot exclusivamente para mejorar el lenguaje y la legibilidad del manuscrito. Tras emplear esta herramienta, revisé y edité manualmente todo el contenido según fue necesario y asumo plena responsabilidad por la totalidad de la publicación. Asimismo, utilicé asistencia de M365 Copilot para generar algunos scripts de visualización empleados únicamente con fines experimentales y de presentación de resultados dentro del estudio. Del mismo modo, supervisé, verifiqué y edité manualmente toda la salida generada, asumiendo plena responsabilidad por su validez y uso dentro de este trabajo.

Referencias

Ahmed Kaleem, M. (2025). Estudio de técnicas, parámetros y modelos en un sistema RAG [Trabajo de Fin de Grado, Universitat Politécnica de València]. Escuela Técnica Superior de Ingeniería Informática. Disponible en: https://riunet.upv.es/server/api /core/bitstreams/ fecd1e71-a9bf-4a92-a5ca-49a4649d45df/content

Amiri, M., & Bocklitz, T. (2025). Chunk twice, embed once: A systematic study of segmentation and representation trade-offs in chemistry-aware retrieval-augmented generation. arXiv. https://arxiv.org/abs/2506.17277

Chase, H. (2022). LangChain [Software]. https://github.com/hwchase17/langchain

Chroma Team. (2023). Chroma: The AI-native open-source embedding database [Software]. https://www.trychroma.com/

Ferraris, A. F., Audrito, D., Siragusa, G., & Piovano, A. (2024). Legal chunking: Evaluating methods for effective legal text retrieval. En J. Savelka et al. (Eds.), Legal Knowledge and Information Systems (pp. 275–281). IOS Press. https://doi.org/10.3233/FAIA241255

Günther, M., Mohr, I., Williams, D. J., Wang, B., & Xiao, H. (2024). Late chunking: Contextual chunk embeddings using long-context embedding models. arXiv. https://arxiv.org/abs/2409.04701.

Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., & Grave, E. (2022). Few-shot learning with retrieval augmented language models. arXiv preprint arXiv:2208.03299. https://arxiv.org/abs/2208.03299

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Renard Lavaud, L., Lachaux, M.-A., Stock, P., Le Scao, T., Lavril, T., Wang, T., Lacroix, T., & El Sayed, W. (2023). Mistral 7B. arXiv preprint arXiv:2310.06825.

Khattab, O., & Zaharia, M. (2020). ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. arXiv preprint arXiv:2004.12832. https://doi.org/10.48550/arXiv.2004.12832

LangChain Community. (2024). Text splitters: Recursive character. LangChain Documentation. Disponible en:https://python.langchain.com/docs/modules/data_connection/document_transformers/recursive_character_text_splitter/

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2024). A comprehensive overview of large language models. arXiv preprint arXiv:2307.06435. https://arxiv.org/abs/2307.06435.

Nogueira, R., & Cho, K. (2019). Passage Re-ranking with BERT. arXiv:1901.04085.

Ollama Team. (2023). Ollama [Software]. https://ollama.com

Pipitone, N., & Houir Alami, G. (2024). LegalBench-RAG: A benchmark for retrieval-augmented generation in the legal domain. arXiv preprint arXiv:2408.10343.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of EMNLP.

Reuter, M., Lingenberg, T., Liepina, R., Lagioia, F., Lippi, M., Sartor, G., Passerini, A., & Sayin, B. (2025). Towards reliable retrieval in RAG systems for large legal datasets. arXiv preprint arXiv:2510.06999.

Sentence-Transformers. (n.d.). Pretrained models. https://www.sbert.net/docs/ sentence_transformer/pretrained_models.html

Sentence-Transformers. (n.d.). Retrieve & Re-Rank. https://sbert.net/examples/ sentence_transformer/applications/retrieve_rerank/README.html

Smalle, E. H. M., Bogaerts, L., Simonis, M., Duyck, W., Page, M. P. A., Edwards, M. G., & Szmalec, A. (2016). Can chunk size differences explain developmental changes in lexical learning? Frontiers in Psychology, 6, 1925. https://doi.org/10.3389/fpsyg.2015.01925

Smith, B., & Troynikov, A. (2024). Evaluating chunking strategies for retrieval. Chroma Technical Report. https://research.trychroma.com/evaluating-chunking

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). Transformers: State-of-the-Art Natural Language Processing. In Proceedings of EMNLP: System Demonstrations.