Asistente Legal con IA Generativa

Karen Alejandra Freire Rosero
kfreire@unicauca.edu.co

I. EXPLICACIÓN DEL CASO

El consultorio legal enfrenta el desafío de optimizar el proceso de consulta de un archivo Excel con 329 sentencias de la Corte Constitucional colombiana relacionadas con redes sociales, libertad de expresión, acoso escolar y otros temas. Actualmente, los abogados deben leer manualmente cada paso para encontrar información relevante, un proceso lento y propenso a errores.

Se implementó una solución basada en arquitectura RAG (Retrieval-Augmented Generation) que permite a los usuarios realizar preguntas en lenguaje natural y obtener respuestas precisas y en lenguaje coloquial, basadas exclusivamente en los datos del archivo.

II. SUPUESTOS

A. Formato de datos: El archivo `sentencias_pasadas.xlsx` contiene las columnas # (ID único), `Providencia`, `Tema - subtema`, `sintesis` y `resuelve`, que son suficientes para la búsqueda semántica.

B. Modelos de IA: Se utilizaron modelos de open-source (`all-MiniLM-L6-v2` para embeddings) para garantizar la reproducibilidad y privacidad de los datos.

C. El modelo de embeddings `all-MiniLM-L6-v2` no está especializado en derecho colombiano, pero sí es capaz de capturar similitud semántica en lenguaje natural.

D. Hardware: Ejecución en entorno local con CPU (sin GPU), lo que demuestra la eficiencia de la solución.

III. FORMAS DE RESOLVER EL CASO Y OPCIÓN TOMADA

Enfoques evaluados:

Enfoque	Ventajas	Desventajas
Búsqueda por palabras clave	Simple y rápida	No entiende sinónimos ni contexto
Búsqueda semántica simple	Comprende significado	El usuario debe interpretar los resultados
RAG con embeddings + LLM generativo	Respuestas naturales y contextualizadas	Mayor complejidad y dependencia de modelo generativo
RAG optimizado (implementado)	Preciso, rápido, con memoria contextual	Requiere configuración inicial

Se desarrolló un sistema RAG con las siguientes características:

1. Indexación: Los 329 documentos se convirtieron en embeddings y se almacenaron en ChromaDB.
2. Memoria conversacional: Implementada con umbral de similitud de 0.50, que permite mantener contexto entre preguntas solo cuando son sobre el mismo tema.
3. Filtrado por relevancia: Solo se consideran documentos con similitud $\geq$ 0.50, garantizando respuestas de calidad.
4. Acceso optimizado: Uso de `collection.get(ids=[...])` para recuperación instantánea por ID cuando se usa memoria.
5. Detección específica: Búsqueda directa de términos como “PIAR” en el texto completo de los documentos.

IV. RESULTADOS DEL ANÁLISIS DE LOS DATOS Y DEL MODELO

Estadísticas generales

Métrica	Valor
Total de sentencias cargadas	329
Documentos indexados	329

Modelo de embeddings	all-MiniLM-L6-v2
Base de datos vectorial	ChromaDB persistente

Resultados por pregunta

Pregunta	Resultado	Precisión
Sentencias de redes sociales	T-394/24, T-063/24, A. 480/24	Alta (scores 0.61, 0.60, 0.59)
¿De qué se trataron?	Usó memoria (similitud 0.76) y recuperó T-394/24 y T-063/24	Perfecto
Sentencia de acoso escolar	Identificó T-249/24 (score 0.55)	Excelente
Detalle de acoso escolar	Usó memoria (similitud 0.71) y mostró T-249/24	Correcto
Casos sobre PIAR	Detectó PIAR en T-249/24 (búsqueda en texto completo)	Muy bueno

V. FUTUROS AJUSTES O MEJORAS

A. Interfaz gráfica (UI):
- Desarrollar una aplicación web con Streamlit o Gradio para facilitar el uso por parte de los abogados.
- Incluir un historial visual de la conversación.

B. Modelo de lenguaje local:
- Integrar un LLM pequeño como Phi-3 o Llama.cpp para generar respuestas aún más naturales.
- Mantener la privacidad de los datos (todo local).

C. Expansión de la base de datos:
- Incluir más sentencias y actualizaciones periódicas.
- Automatizar la carga de nuevos archivos Excel.

D. Mejoras en la detección:
- Implementar un sistema de sinónimos automáticos (ej: “bullying” = “acoso escolar”).
- Añadir ponderación por campos (dar más peso a “sintesis” que a “Temas”).

E. Exportación de resultados:
- Permitir descargar las respuestas y los casos relevantes en PDF o Excel.
- Incluir citas directas a las providencias.

VI. APRECIACIONES Y COMENTARIOS

Logros destacados:

1. Eficiencia: El sistema procesa 329 documentos y responde en segundos, comparado con horas de lectura manual.
2. Precisión semántica: La búsqueda por embeddings encuentra casos relevantes aunque las palabras no coincida exactamente (ej: “acoso escolar” encontró T-249/24).
3. Memoria inteligente: El umbral de 0.50 demostró ser el valor óptimo:
   - Pregunta 2 (mismo tema) ⟶ similitud 0.76 ⟶ usó memoria
   - Pregunta 3 (tema diferente) ⟶ similitud 0.49 ⟶ No usó memoria
4. Optimización crítica: El acceso directo por ID (`collection.get()`) es mucho más rápido que una nueva búsqueda semántica.
5. Detección de PIAR: La búsqueda en texto completo encontró correctamente el caso T-249/24, demostrando la importancia de no limitarse a metadatos.

Lecciones aprendidas:

- El umbral de similitud es fundamental para balancear precisión y recuperación. 0.50 funciona perfectamente.
- La memoria conversacional mejora drásticamente la experiencia del usuario, pero debe activarse solo cuando es relevante.
- La deduplicación de documentos evita respuestas redundantes.