Examen Final - Recuperación de Información

28 de julio 2025

Objetivo General

Diseñar e implementar un sistema de Recuperación de Información (IR) que:

a) Ingesta y preprocesa un corpus de documentos científicos (subset del 1% de arXiv).

b) Implemente recuperación usando:

- Modelo TF–IDF
- Modelo BM25
- índice vectorial usando FAISS o ChromaDB

c) Integre un módulo RAG (Retrieval-Augmented Generation) con un modelo de lenguaje.

d) Evalúe la calidad de la recuperación comparando los resultados de cada modelo.

Descripción del Dataset

El dataset consiste en un subconjunto de metadatos de artículos científicos de arXiv. Cada registro contiene información estructurada como:

- **id**: identificador único del artículo.
- **title**: título del artículo.
- **abstract**: resumen del contenido.
- **authors**: lista de autores.
- **categories**: áreas temáticas (ej. hep-ph, cs.LG).
- **update_date**: fecha de la última actualización.

Este conjunto de datos se utiliza ampliamente en tareas de recuperación de información científica y permite realizar búsquedas textuales y basadas en embeddings.

Datos Proporcionados

- Corpus de documentos en formato JSON:

```
{
  "id": "0704.0001",
  "title": "...",
  "abstract": "...",
  "authors": "...",
  "categories": "...",
  "update_date": "..."
}
```

- Archivo de consultas (queries.txt):

```
diphoton production cross sections
quantum chromodynamics
higgs boson decay
machine learning for particle physics
top quark production
```

Tareas a Realizar

1. Preprocesamiento

- Convertir todos los textos a minúsculas.
- Eliminar stopwords y signos de puntuación.
- Tokenizar título y resumen para crear el texto indexable.

2. Indexación

- TF–IDF.
- BM25.
- Embeddings: generar vectores y crear un índice con FAISS o ChromaDB.

3. Recuperación

Implementar funciones de búsqueda:

```
search_tfidf(query, top_k=10)
search_bm25(query, top_k=10)
search_faiss(query, top_k=10)
```

Mostrar resultados para cada consulta: identificador, título y fragmento del resumen.

4. RAG

- Tomar el top-3 documentos del índice vectorial.
- Pasar su contenido como contexto a un modelo de lenguaje.
- Generar una respuesta final que resuma la información y justifique la relevancia de los documentos.

5. Evaluación

- Comparar resultados de TF–IDF, BM25 y FAISS/ChromaDB:
  - ¿Cuáles documentos aparecen en común?
  - ¿Qué diferencias hay en el ordenamiento?
- Medir similitud entre rankings contando cuántos documentos del top-10 coinciden.
- Analizar la respuesta generada con RAG: verificar si usa la información recuperada y si responde coherentemente a la consulta.

6. Informe

Entregar un Jupyter Notebook que incluya:

- Implementación de la arquitectura.
- Tabla comparativa de resultados entre modelos.
- Ejemplo de una consulta y su respuesta generada con RAG.
- Diferencias entre modelos y utilidad del RAG.

Criterios de Evaluación (100%)

- Implementación de TF–IDF y BM25 (20%).
- Implementación de FAISS/ChromaDB con embeddings (20%).
- Integración del módulo RAG (20%).
- Análisis comparativo entre modelos (20%).
- Calidad de la implementación (20%).