

1. Introducción

El proyecto consiste en un sistema ejecutado a través de la línea de comandos que permite al usuario realizar consultas de texto libre sobre el conjunto de documentos ArguAna. Los modelos utilizados para la recuperación de información son los especificados en el proyecto: Jaccard, TF-IDF y BM25. Los documentos recuperados se ordenan según su relevancia de acuerdo con el modelo seleccionado.

2. Descripción del corpus utilizado

El corpus utilizado es ArguAna de HuggingFace, un conjunto de datos que incluye tanto los qrels como el texto necesario para entrenar y evaluar los modelos. Se trata de un conjunto de debates argumentativos en línea, diseñado específicamente para la evaluación de sistemas de recuperación de información.

Este dataset proporciona tanto el corpus utilizado para la recuperación como los qrels, que permiten medir el desempeño de los modelos implementados.

3. Explicación de las decisiones de diseño

La arquitectura del proyecto se diseñó con un enfoque modular para facilitar la claridad del código y la reutilización de componentes. Se creó el paquete `mceclib` con el propósito de funcionar como una librería interna que agrupa toda la lógica central del sistema.

En este paquete se organizaron los módulos según su responsabilidad: `preprocessing.py` contiene las funciones necesarias para limpiar y normalizar el texto, lo que permite separar esta etapa fundamental del proceso de recuperación y evitar duplicación de código. Los archivos `jaccard.py`, `tfidf.py` y `bm25.py` encapsulan la implementación de cada modelo de recuperación de información, lo que facilita su comparación. Por su parte, `evaluate.py` reúne las métricas de evaluación, como Precision@k, AP y MAP, permitiendo evaluar los modelos de forma coherente y centralizada.

La carpeta `data` almacena los archivos del corpus y las consultas de forma organizada. Finalmente, la interfaz principal del usuario se implementó mediante `interfaz_cli.py`, que actúa como punto de entrada al sistema y permite interactuar fácilmente con los modelos desde la línea de comandos sin exponer la complejidad interna de la librería.

Esta estructura favorece una clara separación de responsabilidades y permite que el notebook `SistemaRecuperacion.ipynb` se enfoque exclusivamente en la explicación conceptual, manteniendo el código aislado y bien organizado.

4. Ejemplos de consultas y resultados

Para la realización de la recuperación, se utilizó una consulta genérica, en este caso "*Linux OS*". Para cada modelo, se entregaron los k documentos especificados, siendo en este caso 10. Estos documentos se muestran en las Figuras 1, 2 y 3, donde se puede observar el texto extraído y el formato generado por cada modelo.

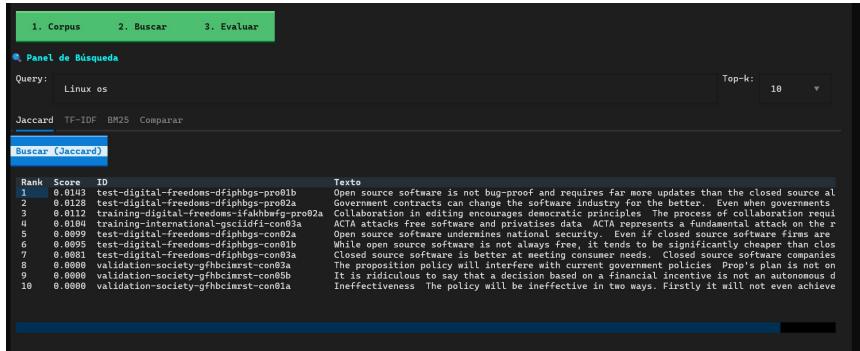


Figura 1: Búsqueda con Jaccard.

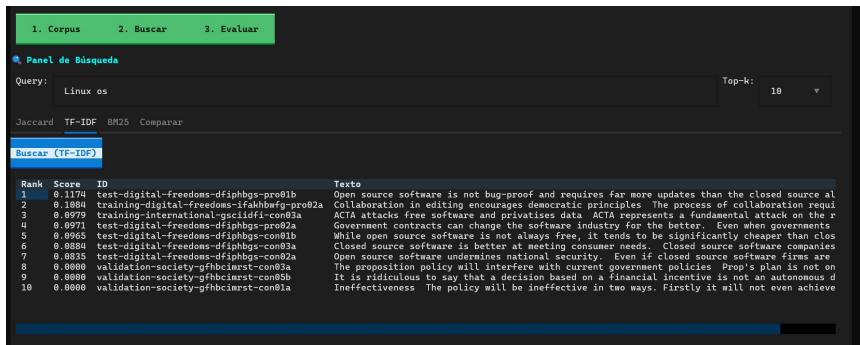


Figura 2: Búsqueda con TF-IDF.

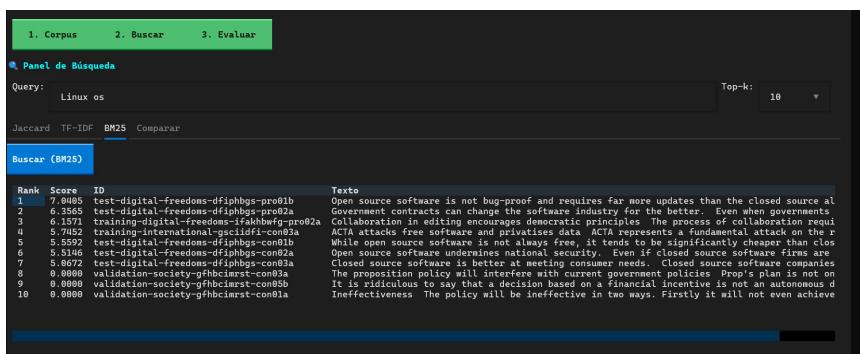


Figura 3: Búsqueda con BM25.

En la evaluación del modelo, para una misma consulta, se calculó el MAP, que mide el rendimiento global de la calidad de los modelos a lo largo de un conjunto de consultas. En este caso, se observa que, al utilizar la consulta "benefits of vegetarian diet", se muestra el MAP de los tres modelos evaluados.

En este apartado, se calculan las métricas de evaluación del modelo con respecto a los

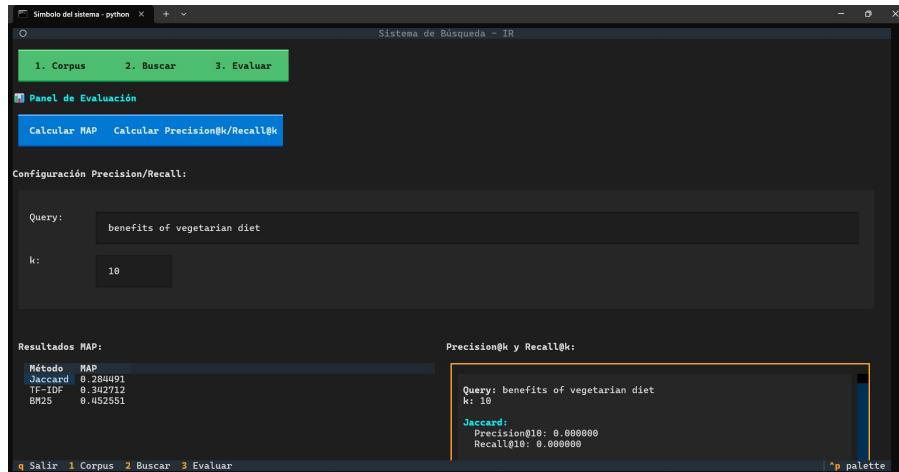


Figura 4: Evaluación MAP.

qrels, lo que indica que los modelos TF-IDF y BM25 extrajeron correctamente los documentos relevantes dentro de los k documentos recuperados.

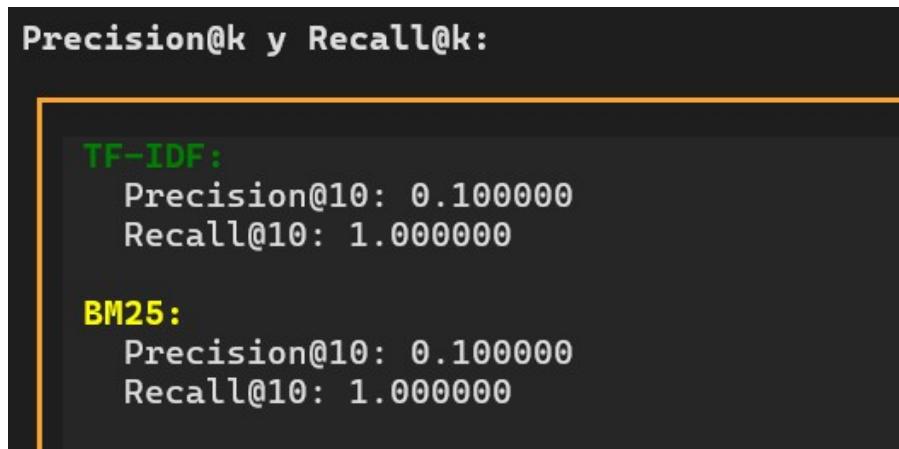


Figura 5: Precision y Recall.

5. Análisis de métricas de evaluación

Para el análisis de MAP, se utilizaron múltiples consultas con el fin de evitar sesgos asociados a una sola consulta. Los resultados obtenidos muestran que **Jaccard** alcanza valores entre **0.26** y **0.28**, **TF-IDF** obtiene valores entre **0.31** y **0.33**, y **BM25** logra puntajes entre **0.44** y **0.46**. Estos resultados indican que **BM25** recupera y ordena los documentos relevantes de manera más efectiva que TF-IDF, y este último supera a Jaccard. Cabe recordar que un valor de MAP igual a 1 implica una recuperación perfectamente relevante, mientras que un valor de 0 implica que no se recuperan documentos relevantes.

Para evaluar la *precisión* y el *recall*, se utilizaron los qrels proporcionados por el dataset. Usando las mismas consultas, si el modelo devuelve los documentos etiquetados como relevantes, se considera que está funcionando correctamente. En particular, se empleó la consulta "benefits

of vegetarian diet” y se evaluaron los primeros diez documentos recuperados. Los resultados fueron los siguientes:

- **Jaccard**: precisión = 0, recall = 0
- **TF-IDF**: precisión = 0.1, recall = 1
- **BM25**: precisión = 0.1, recall = 1

Estos valores reflejan que Jaccard no logró ubicar el documento relevante —según los qrels— dentro de los primeros diez resultados, mientras que tanto TF-IDF como BM25 sí lo hicieron. Esto es consistente con su bajo MAP en comparación con los otros dos modelos, lo que implica que Jaccard tiene menor probabilidad de recuperar documentos relevantes.