ESCUELA POLITÉCNICA NACIONAL

Facultad de Ingeniería de Sistemas

Recuperación de Información

Integrantes:

- Alexis Guanoluisa
- José Quiros

Fecha: 09/06/2025

Proyecto del primer bimestre: Sistema de Recuperación de Información

---

Introducción

El proyecto consiste en desarrollar un Sistema de Recuperación de Información (SRI) en Python que use el corpus sobre un foro de WordPress. Se detallarán las decisiones de diseño y se mostrará ejemplos de las queries ingresadas y los resultados obtenidos junto con las métricas de evaluación correspondientes.

Descripción del corpus utilizado

El corpus utilizado para el proyecto corresponde al subconjunto beir/cqadupstack/wordpress, que proviene de la colección CQADupStack, y se encuentra disponible en la página web ir-datasets.com. Este corpus está diseñado para la tarea de recuperación de preguntas duplicadas en foros de preguntas y respuestas.

Este subconjunto en particular se extrae del subforo de WordPress en StackExchange, una comunidad especializada en el desarrollo, personalización y solución de problemas relacionados con esta plataforma.

Cada documento del corpus se encuentra en inglés y corresponde a una pregunta con su título y cuerpo textual. Las consultas son preguntas reales realizadas por usuarios, que se utilizan como entradas para recuperar preguntas similares ya respondidas. Las relaciones de duplicidad están etiquetadas, lo que permite la evaluación de los sistemas de recuperación de información.

Las queries representan una consulta y tiene los siguientes atributos: query_id de tipo string, text de tipo string y tags de tipo lista de strings. El corpus contiene un total de 541 queries.

Los docs representan un documento del corpus y tiene los siguientes atributos: doc_id de tipo string, text de tipo string, tittle de tipo string y tags de tipo lista de strings. Existe un total de 49K documentos que conforman el corpus.

Las qrels representan una asociación de relevancia entre una consulta y un documento, es decir, indica si un documento es relevante para una consulta dada. Contiene los siguientes atributos: query_id de tipo string, doc_id de tipo string, relevance de tipo int e iteration de tipo string. El corpus incluye 744 qrels.

Antes de utilizar el corpus, se realizó el preprocesamiento del mismo. Para la etapa de preprocesamiento se empleó la conversión a minúsculas, la normalización y tokenización, eliminación de stopwords, stemming y lematización utilizando spaCy.

Explicación de las decisiones de diseño

Los modelos de recuperación de información que se utilizaron fueron TF-IDF y BM25. La elección de estos modelos permite evaluar la capacidad de estos modelos para satisfacer a la necesidad de información de un usuario. Para TF-IDF se utilizó la similitud coseno para obtener los documentos relevantes para una query dada, mientras que para BM25 las estimaciones (scores) determinan la relevancia de los documentos para una query determinada.

Las librerías requeridas fueron las siguientes: Pandas, NumPy, rank_bm25, sklearn y nltk. Pandas y NumPy, estas se utilizaron para el manejo y procesamiento de los datos.

Pandas facilitó la carga, organización y análisis del corpus, permitiendo estructurar las consultas, documentos y relevancias en dataframes, para un acceso más eficiente y legible. Por su parte, NumPy fue fundamental para realizar operaciones vectoriales y estadísticas, como el cálculo de similitudes, ordenamiento de resultados por puntaje y evaluación de métricas. Ambas librerías permitieron un procesamiento de datos rápido lo cual ayudó a mejorar el tiempo de respuesta del SRI. De rank_bm25 se importó BM25Okapi, la cual es una clase que sirve para implementar el modelo BM25, ya que cuenta con métodos para calcular el IDF y obtener los scores. De sklearn se importaron el cosine_similarity para calcular la similitud coseno y TfidfVectorizer para convertir una colección de documentos a una matriz de características TF-IDF. Finalmente, para el procesamiento de lenguaje natural, se utilizó la librería nltk, de la cual se importó stopwords para eliminar las stopwords, regexp_tokenize para tokenizar los documentos acordes a una expresión regular, SnowballStemmer para el stemming y WordNetLemmatizer para la lematización.

Ejemplos de consultas y resultados

A continuación, se presenta los resultados para 2 consultas, ambas con sus respectivos resultados para los modelos TF-IDF, BM25 y el promedio de ambos modelos.

- **Ejemplo 1:**
  Consulta: install a plugin
  Resultados:
  La Figura 1 indica los resultados de los documentos encontrados para el modelo TF-IDF ordenados por su respectivo valor de la similitud coseno.

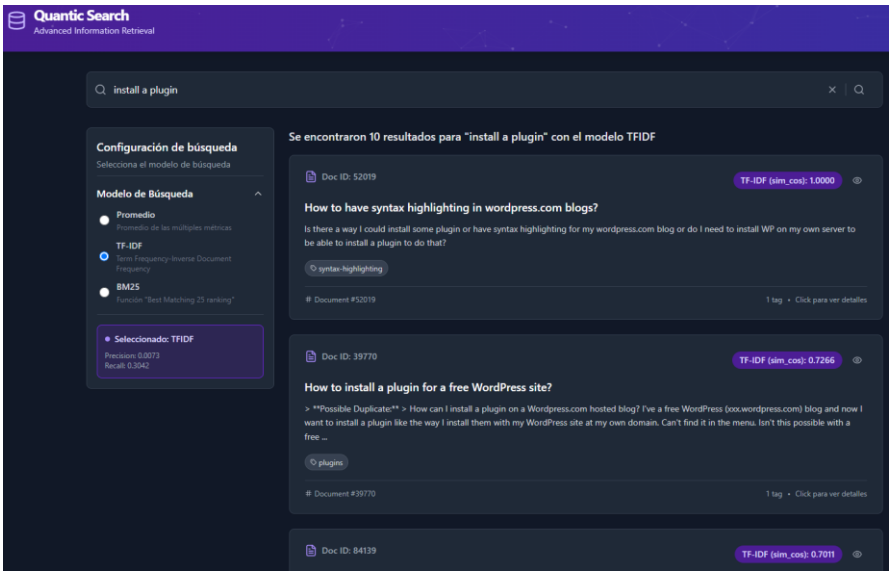

Figura 1. Resultados obtenidos por el modelo TF-IDF.

La Figura 2 muestra los resultados de los documentos encontrados para el modelo BM25 ordenados por su respectivo score.

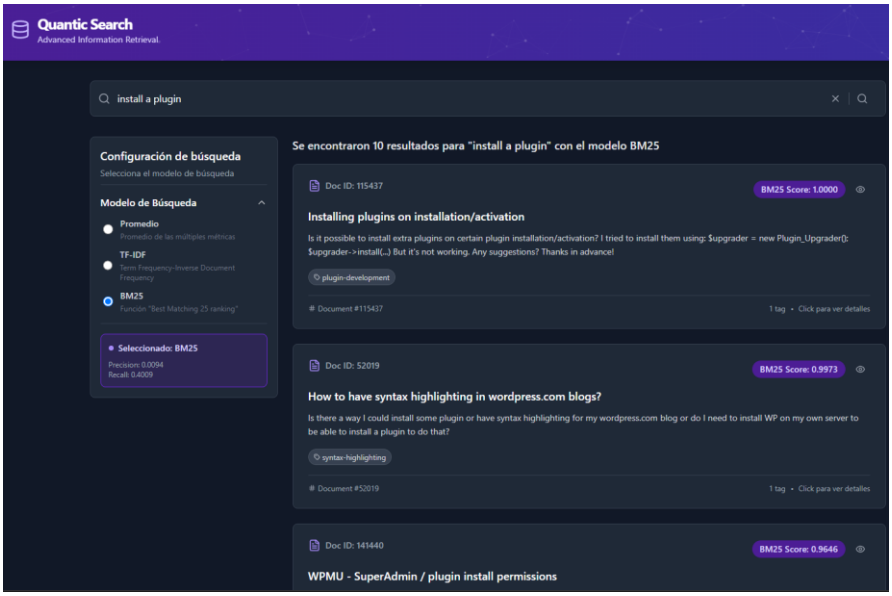

Figura 2. Resultados obtenidos por el modelo BM25.

- **Ejemplo 2:**
  Consulta: add a video an image gallery
  Resultados:
  La Figura 3 indica los resultados de los documentos encontrados para el modelo TF-IDF ordenados por su respectivo valor de la similitud coseno.

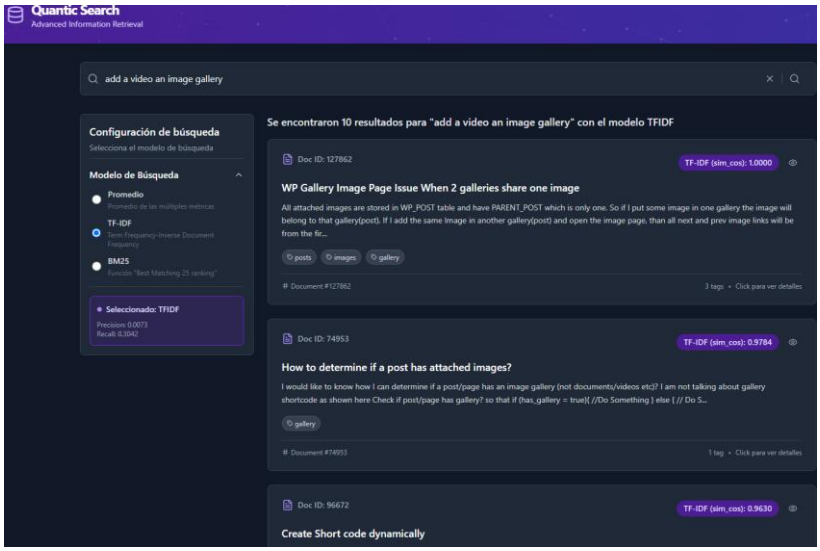

Figura 3. Resultados obtenidos por el modelo TF-IDF.

La Figura 4 muestra los resultados de los documentos encontrados para el modelo BM25 ordenados por su respectivo score.

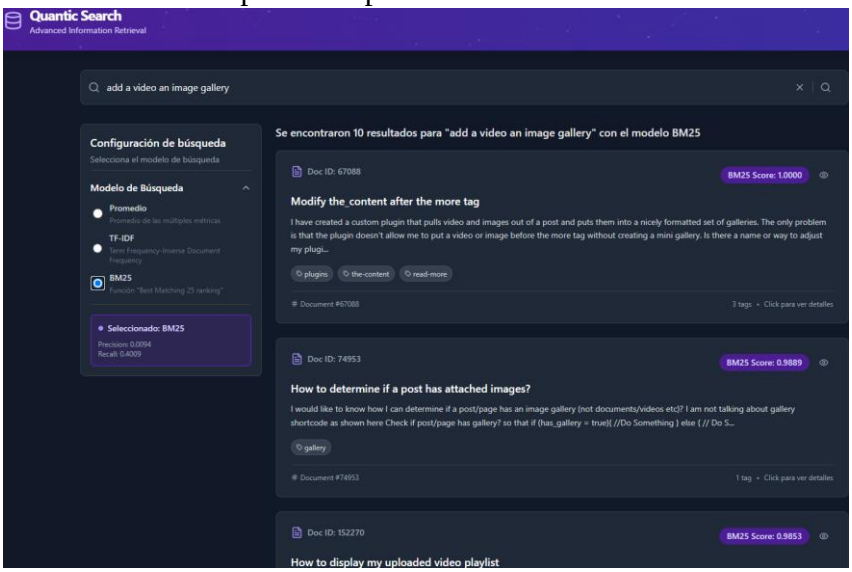

Figura 4. Resultados obtenidos por el modelo TF-IDF.

Análisis de métricas de evaluación

Para evaluar el rendimiento del sistema de recuperación de información, se utilizaron las etiquetas de relevancia (qrels) proporcionadas por el corpus.

Se compararon dos enfoques: un modelo TF-IDF y otro modelo que utiliza el algoritmo BM25, ambos aplicados sobre el mismo índice de documentos. A continuación, en la Tabla 1 se presentan los resultados que se obtuvieron:

Modelo	Precisión	Recall
TF-IDF	0.007320	0.304157
BM25	0.009427	0.400941

Tabla 1. Comparación de las métricas de evaluación entre los modelos.

Los resultados muestran que BM25 supera a TF-IDF tanto en precisión como en recall, lo que indica una mejor capacidad para recuperar documentos relevantes. Sin embargo, se observa que ambos modelos presentan una precisión baja, lo que indica que muchos documentos recuperados no son relevantes.

El recall relativamente más alto indica que ambos modelos son capaces de recuperar una proporción razonable de documentos relevantes, en especial BM25, lo que lo convierte en una mejor opción para el SRI. Futuras mejoras podrían incluir el uso de representaciones semánticas, como embeddings para captar mejor las similitudes semánticas entre preguntas formuladas de manera distinta.