Escuela Politécnica Nacional
Recuperación de la Información

Informe Técnico

Ozzy Loachamín

Nelson Casa

Miércoles, 10 de diciembre de 2025

1. Introducción

El objetivo de este proyecto fue desarrollar un Sistema de Recuperación de Información (IR) completamente funcional, capaz de procesar un corpus real, aplicar un pipeline de preprocesamiento, indexar los documentos mediante un índice invertido y utilizar tres modelos clásicos de ranking Jaccard, TF-IDF y BM25 para recuperar los documentos más relevantes a una consulta. Además, se implementó una herramienta CLI para realizar búsquedas desde terminal y un módulo de evaluación que calcula métricas estándar como *precision@k*, *recall@k*, *average precision (AP)* y *mean average precision (MAP)*. La intención del informe es explicar de manera directa y práctica qué se realizó en el sistema, sin incluir teoría innecesaria, y enfocándose en el proceso, las decisiones de diseño y los resultados.

2. Descripción del corpus utilizado

Para la construcción del sistema se empleó el conjunto de datos **BEIR – FIQA (Financial Question Answering)** en su partición *test*, disponible mediante la librería `ir_datasets` [1, 2]. Este corpus contiene aproximadamente 57,638 documentos redactados en lenguaje natural sobre temas financieros, incluyendo inversión, ahorro, cuentas bancarias, impuestos, seguros y otros conceptos relacionados con finanzas personales. Adicionalmente, el dataset proporciona consultas reales escritas por usuarios y archivos de relevancias (qrels) en formato TREC, los cuales especifican qué documentos deben considerarse relevantes para cada consulta.

La razón para elegir este corpus es que FIQA ofrece un escenario realista y exigente para sistemas de recuperación, al incluir textos extensos y variados junto con

relevancias verificadas, lo que facilita la evaluación objetiva del desempeño de los modelos implementados.

3. Decisiones de diseño del sistema

Durante el desarrollo del sistema se tomaron decisiones enfocadas en garantizar consistencia entre indexación y consulta, mantener un buen rendimiento sobre el corpus y asegurar que el sistema pudiera ser empleado tanto desde notebook como desde terminal. La filosofía general fue implementar un pipeline claro, modular y eficiente, capaz de procesar textos de forma sistemática y generar estructuras internas reutilizables como el índice y la matriz TF-IDF.

3.1. Preprocesamiento aplicado

Todos los documentos y consultas pasaron por el mismo flujo de limpieza para garantizar consistencia en la forma en que el sistema interpreta el texto. El preprocesamiento incluyó normalización de texto, tokenización, eliminación de stopwords y *stemming*. Estas operaciones se aplicaron usando las funciones de la librería creada para el proyecto, **libJames**. La transición del texto original al texto preprocesado puede observarse en la (Fig. 1), donde se aprecia la reducción del ruido presente en el corpus crudo.

```
docs_df[['textD','text_processed']]
```

	textD	text_processed
0	I'm not saying I don't like the idea of on-the...	say like idea job train expect compani train w...
1	So nothing preventing false ratings besides ad...	noth prevent fals rate besid addit scrutini ma...
2	You can never use a health FSA for individual ...	never use health fsa individu health insur pre...
3	Samsung created the LCD and other flat screen ...	samsung creat lcd flat screen technolog like o...
4	Here are the SEC requirements: The federal sec...	sec requir feder secur law defin term accredit...
...	...	...
57633	>Well, first off, the roads are more than j...	gt well first road hobbi realli go place lot r...
57634	Yes they do. There are billions and billions s...	ye billion billion spent subsidi pharmaceut co...
57635	>It's biggly sad you don't understand human...	gt biggli sad understand human natur noth huma...
57636	"Did your CTO let a major group use ""admin/ad...	cto let major group use admin admin administr ...
57637	Giving the government more control over the di...	give govern control distribut good servic even...

57638 rows × 2 columns

Figura 1: Procesamiento de documentos

En cuanto a las decisiones específicas, se optó por utilizar **stemming** en lugar de lematización. Esto se debe a que el *stemming* es mucho más liviano computacionalmente, no requiere modelos complejos y permite reducir la *sparsity* del vocabulario, lo cual mejora el rendimiento en modelos basados en bolsa de palabras. La lematización, si bien más precisa, implica un costo considerable en tiempo y memoria, especialmente con corpus grandes como FIQA. Por esta razón, el *stemming* resultó la opción más eficiente para los objetivos del proyecto.

3.2. Construcción del índice invertido

Tras el preprocesamiento se construyó un índice invertido que almacena, para cada término, los documentos en los que aparece y su frecuencia. Este índice es fundamental para los modelos basados en conteo, como Jaccard y BM25. La estructu-

ra generada permitió acceder rápidamente a los documentos relevantes para cada término y calcular estadísticas como la longitud de documentos y el promedio de longitudes del corpus.

3.3. Modelos de recuperación implementados

Se implementaron tres modelos distintos para evaluar sus comportamientos sobre el corpus:

- **Jaccard**, incluido como baseline debido a su simplicidad, basado únicamente en la coincidencia entre conjuntos de términos.
- **TF-IDF**, entrenado con un `TfidfVectorizer`, manejando las matrices en formato disperso (.npz) para optimizar memoria.
- **BM25**, implementado con los parámetros estándar y conocido por su excelente desempeño en documentos largos, especialmente en textos financieros.

La integración de estos tres enfoques permitió comparar bajo las mismas condiciones cómo se comportan modelos con distintos supuestos y niveles de complejidad.

3.4. Interfaz de consulta (CLI)

Se desarrolló el código (`cli_search.py`) que permite realizar consultas desde la terminal sin necesidad de utilizar el notebook. Este script carga el índice, el vectorizador y la matriz TF-IDF almacenados previamente, y ejecuta los tres modelos de recuperación sobre la consulta ingresada por el usuario. La (Fig. 2) incluye un ejemplo del funcionamiento de la CLI y de cómo se muestran los resultados para cada modelo.

```
!python cli_search.py "Where should I park my emergency fund?"

Cargando assets…
Procesando consulta…

=== Resultados de búsqueda (agrupados por modelo) ===


### Modelo: JACCARD ###
 1) ID: 290830    Score: 0.1429    Snippet: fund
 2) ID: 589544    Score: 0.1053    Snippet: invest exist would bank park overnight fund feder reserv int…
 3) ID: 372677    Score: 0.0909    Snippet: fund prospectu good place start
 4) ID: 264740    Score: 0.0909    Snippet: ishar jantzi social index fund
 5) ID: 274859    Score: 0.0833    Snippet: own physic gold assum coin own gold fund
----------------------------------------

### Modelo: BM25 ###
 1) ID: 537111    Score: 12.0534   Snippet: know free sourc year histor data larg set compani singl comp…
 2) ID: 589544    Score: 11.6223   Snippet: invest exist would bank park overnight fund feder reserv int…
 3) ID: 376148    Score: 11.4861   Snippet: bond necessarili safer stock market ultim thing low risk mut…
 4) ID: 241085    Score: 11.3121   Snippet: go talk benefit offic understand deadlin rule program ir enf…
 5) ID: 10374     Score: 10.9456   Snippet: exactli newer set better cheaper past basic airport airlplan…
----------------------------------------

### Modelo: TFIDF ###
 1) ID: 178386    Score: 0.3147    Snippet: look like use employe benefit pay park near home definit qua…
 2) ID: 551764    Score: 0.3069    Snippet: option park money bank work best
 3) ID: 73700     Score: 0.2903    Snippet: inugo park space finder help find earli bird park cbd help f…
 4) ID: 298053    Score: 0.2770    Snippet: thought find auckland park get stress level peak nod agreeme…
 5) ID: 105340    Score: 0.2651    Snippet: best tl dr could make origin http qz com your rent citi apar…
```

Figura 2: Funcionamiento de la CLI

3.5. Persistencia de modelos y estructuras

Con el fin de evitar recomputar el índice y recalcular la matriz TF-IDF cada vez, se almacenaron las estructuras principales en archivos independientes: el índice invertido y los documentos procesados se guardaron en `retrieval_assets_small.pkl`, el vectorizador en `vectorizer.joblib` y la matriz TF-IDF en `tfidf_matrix.npz`.

Este enfoque permitió realizar pruebas rápidas durante el desarrollo y ejecutar la CLI sin necesidad de reindexar el corpus.

4. Ejemplos de consultas y resultados

Consulta 1 — "pleas explain use histor exampl"

A pesar de los errores ortográficos, el sistema recuperó documentos pertinentes gracias al preprocesamiento (*stemming* y normalización).
BM25 y TF-IDF ofrecieron resultados más consistentes relacionados con explicaciones y ejemplos históricos.
Jaccard fue el modelo menos robusto ante la mala escritura.

```
!python cli_search.py "pleas explain use histor exampl"

Cargando assets…
Procesando consulta…

=== Resultados de búsqueda (agrupados por modelo) ===


### Modelo: JACCARD ###
 1) ID: 169028    Score: 0.2500    Snippet: pleas explain use histor exampl would purchas debt solid fin…
 2) ID: 584801    Score: 0.1667    Snippet: use stockchart spread chart take question exampl chart appl …
 3) ID: 381310    Score: 0.1429    Snippet: folk explain comment
 4) ID: 425250    Score: 0.1304    Snippet: gt challeng view pleas provid exampl countri anywher world p…
 5) ID: 421112    Score: 0.1304    Snippet: dafuq read think would better explain excel chart idea wrote…
----------------------------------------

### Modelo: BM25 ###
 1) ID: 169028    Score: 25.3023   Snippet: pleas explain use histor exampl would purchas debt solid fin…
 2) ID: 5360      Score: 17.6643   Snippet: problem comment e declin uk manufactur foreign polici crude …
 3) ID: 82021     Score: 16.5534   Snippet: fair point histor fuck lol pleas explain sinc layman mayb di…
 4) ID: 425250    Score: 14.7395   Snippet: gt challeng view pleas provid exampl countri anywher world p…
 5) ID: 104580    Score: 14.5108   Snippet: soviet union purchas debt shine exampl histor empir work min…
----------------------------------------

### Modelo: TFIDF ###
 1) ID: 169028    Score: 0.5811    Snippet: pleas explain use histor exampl would purchas debt solid fin…
 2) ID: 127845    Score: 0.3509    Snippet: gt anyth point say someth like sure number mean sit back bou…
 3) ID: 506104    Score: 0.3148    Snippet: surpris hear colleagu difficult time explain basic guess lik…
 4) ID: 82021     Score: 0.2915    Snippet: fair point histor fuck lol pleas explain sinc layman mayb di…
 5) ID: 253574    Score: 0.2815    Snippet: gt pleas provid specif thing help economi give specif thing …
----------------------------------------
```

Figura 3: Ejemplo 1 de consulta

Consulta 2 — "Where should I park my emergency fund?"

La consulta sobre fondos de emergencia devolvió resultados financieros relevantes.
TF-IDF y especialmente BM25 ubicaron documentos que explican dónde guardar fondos de emergencia, mostrando puntajes altos y buena coincidencia semántica.
Jaccard recuperó coincidencias básicas, pero con menor precisión que los otros modelos.

```
!python cli_search.py "Where should I park my emergency fund?"

Cargando assets…
Procesando consulta…

=== Resultados de búsqueda (agrupados por modelo) ===


### Modelo: JACCARD ###
 1) ID: 290830    Score: 0.1429    Snippet: fund
 2) ID: 589544    Score: 0.1053    Snippet: invest exist would bank park overnight fund feder reserv int…
 3) ID: 372677    Score: 0.0909    Snippet: fund prospectu good place start
 4) ID: 264740    Score: 0.0909    Snippet: ishar jantzi social index fund
 5) ID: 274859    Score: 0.0833    Snippet: own physic gold assum coin own gold fund
----------------------------------------

### Modelo: BM25 ###
 1) ID: 537111    Score: 12.0534   Snippet: know free sourc year histor data larg set compani singl comp…
 2) ID: 589544    Score: 11.6223   Snippet: invest exist would bank park overnight fund feder reserv int…
 3) ID: 376148    Score: 11.4861   Snippet: bond necessarili safer stock market ultim thing low risk mut…
 4) ID: 241085    Score: 11.3121   Snippet: go talk benefit offic understand deadlin rule program ir enf…
 5) ID: 10374     Score: 10.9456   Snippet: exactli newer set better cheaper past basic airport airlplan…
----------------------------------------

### Modelo: TFIDF ###
 1) ID: 178386    Score: 0.3147    Snippet: look like use employe benefit pay park near home definit qua…
 2) ID: 551764    Score: 0.3069    Snippet: option park money bank work best
 3) ID: 73700     Score: 0.2903    Snippet: inugo park space finder help find earli bird park cbd help f…
 4) ID: 298053    Score: 0.2770    Snippet: thought find auckland park get stress level peak nod agreeme…
 5) ID: 105340    Score: 0.2651    Snippet: best tl dr could make origin http qz com your rent citi apar…
```

Figura 4: Ejemplo 2 de consulta

5. Evaluación del sistema

Para evaluar el rendimiento del sistema se procesaron todas las consultas incluidas en FIQA junto con sus relevancias. El proceso consistió en obtener los rankings generados por cada uno de los tres modelos y compararlos con los documentos marcados como relevantes en los qrels. A partir de esa comparación se calcularon métricas estándar de recuperación, como *precision@k*, *recall@k*, *average precision* por consulta y el **MAP** general por modelo. Estos resultados se almacenaron en archivos CSV que permiten un análisis más detallado si se requiere.

Los resultados obtenidos reflejaron un comportamiento consistente con lo esperado en la literatura del área. **BM25** fue el modelo con mejor desempeño global, alcanzando los mayores valores de MAP, gracias a su capacidad para equilibrar frecuencia y longitud de documentos. **TF-IDF** mostró un rendimiento intermedio y se comportó de manera efectiva en consultas donde los términos clave aparecían con relativa frecuencia. Finalmente, el modelo basado en **Jaccard** presentó los valores más bajos, lo cual es coherente con el hecho de que no utiliza pesos ni frecuencias y depende únicamente de la coincidencia binaria entre términos.

6. Conclusiones

El sistema desarrollado cumple con todos los objetivos planteados para el proyecto, al integrar preprocesamiento consistente, indexación, modelos de ranking clásicos, ejecución desde CLI y evaluación completa mediante métricas estándar. Las decisiones tomadas, como el uso de *stemming*, matrices dispersas y persistencia de estructuras, permitieron construir un sistema eficiente, modular y reproducible. Además, la comparación de los tres modelos permitió entender mejor su comportamiento sobre un corpus amplio y realista como FIQA, identificando a BM25 como el enfoque más robusto en este escenario.

Referencias

[1] Maia M, Handschuh S, Freitas A, Davis B, McDermott R, Zarrouk M, et al. WWW'18 Open Challenge: Financial Opinion Mining and Question Answering. Companion Proceedings of the The Web Conference 2018. 2018.

[2] Thakur N, Reimers N, Rücklé A, Srivastava A, Gurevych I. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. arXiv preprint arXiv:210408663. 2021 4. Available from: https://arxiv.org/abs/2104.08663.