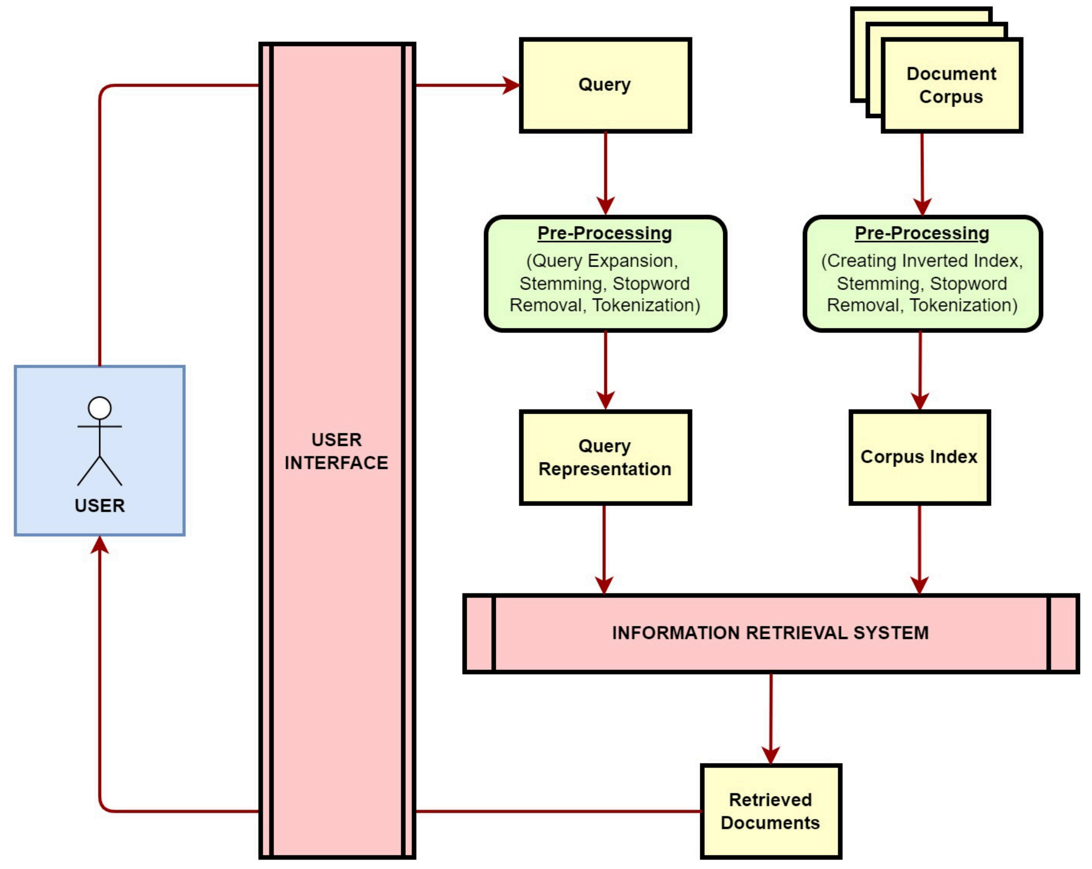
Wiki-Search

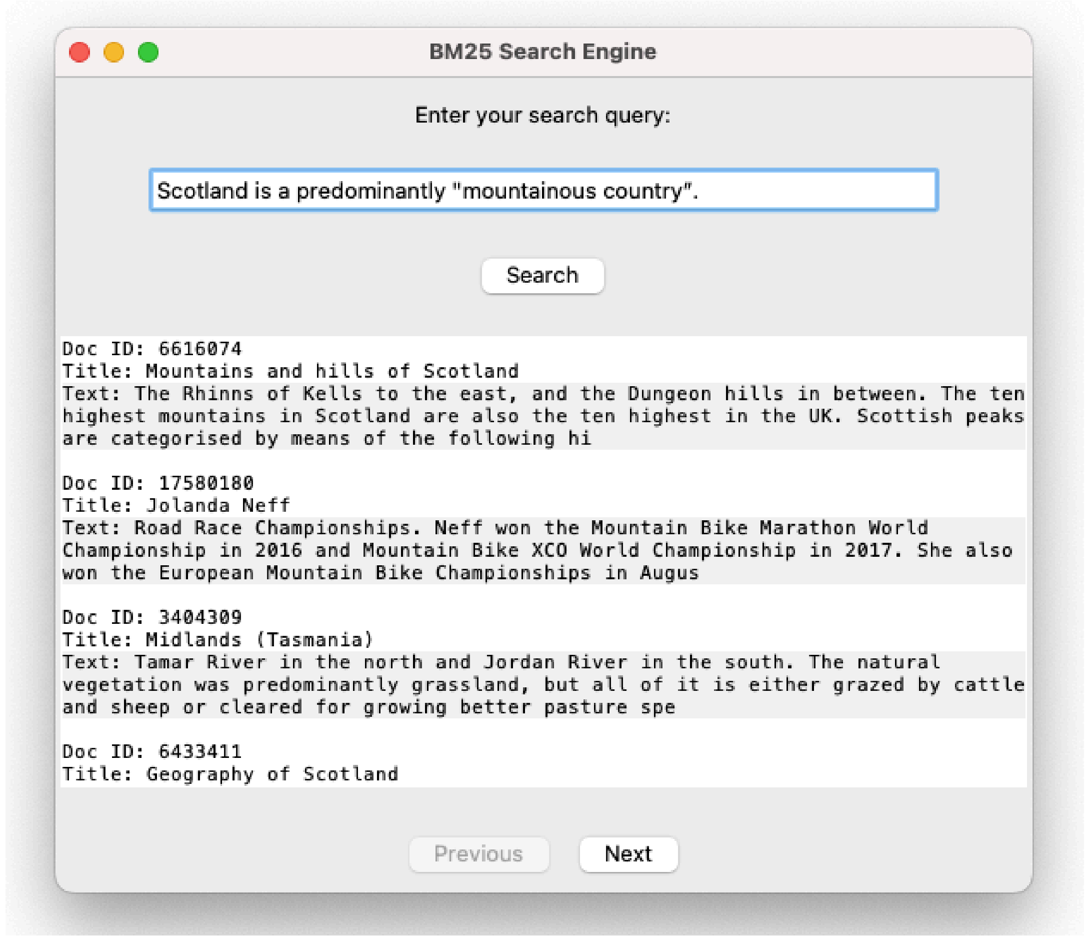
Wiki-Search is a ad-hoc information retrieval system that allows users to query the **dpr-wiki100** dataset. It lists the results of searched queries with a minimal user interface.

Berk Güler & Mustafa Barış Emektar

Introduction

In the ever-expanding digital landscape, efficient access to relevant information has become increasingly challenging. Whether for academic research, professional endeavors, or personal curiosity, the ability to swiftly navigate the vast sea of available content is crucial. This project centers around enhancing the performance of core BM25 ranking schemes—a critical aspect of information retrieval. While existing methods serve as a baseline, the project aims to optimize and improve the effectiveness and efficiency of simple retrieval systems through various enhancements.


USER
USER INTERFACE
Query
Document Corpus
Pre-Processing
(Query Expansion, Stemming, Stopword Removal, Tokenization)
Pre-Processing
(Creating Inverted Index, Stemming, Stopword Removal, Tokenization)
Query Representation
Corpus Index
INFORMATION RETRIEVAL SYSTEM
Retrieved Documents



```
Doc ID: 6616074
Title: Mountains and hills of Scotland
Text: The Rhinns of Kells to the east, and the Dungeon hills in between. The ten
highest mountains in Scotland are also the ten highest in the UK. Scottish peaks
are categorised by means of the following hi

Doc ID: 17580180
Title: Jolanda Neff
Text: Road Race Championships. Neff won the Mountain Bike Marathon World
Championship in 2016 and Mountain Bike XCO World Championship in 2017. She also
won the European Mountain Bike Championships in Augus

Doc ID: 3404309
Title: Midlands (Tasmania)
Text: Tamar River in the north and Jordan River in the south. The natural
vegetation was predominantly grassland, but all of it is either grazed by cattle
and sheep or cleared for growing better pasture spe

Doc ID: 6433411
Title: Geography of Scotland
```

Evaluation & Results

The effectiveness of the system was evaluated using Mean Average Precision (**MAP**) and Normalized Discounted Cumulative Gain (**NDCG@20**).

As seen in Table, the performance measurements with the modified index (stopword removals and stemming) are almost 1.5 times better.

It can be observed that adding query expansion on top of the modified index improves the performance a bit more.

	AP	NDCG@20
BM25 (Non-Modified Index)	0.151913	0.195565
BM25 (Modified Index)	0.208434	0.269578
BM25 (Modified Index with Q.E.)	0.223405	0.281933

MIDDLE EAST TECHNICAL UNIVERSITY

System Architecture

The development of the project is done in Python. **Pyterrier** and **NLTK** libraries are used. Pyterrier library is used in the indexing the dataset, as well as the ranking and performance evaluation phases. **BM25** was chosen as the information retrieval model. Also, NLTK library is used for other operations like query transformation. Also; one of Python's libraries, **tkinter** , was used to create the user interface.

Data Preprocessing

- # of documents: 21,015,324
- # of terms: 4,023,297
- # of postings: 1,000,890,089
- # of fields: 2
- # of tokens: 1,286,954,378

As it can be seen from the values given, the **dpr-wiki100** dataset is a very large dataset which is almost 5GB.

Query Processing

The **nltk** library was utilized to carry out following operations.

- **Tokenization**
- **Stopword Removal**
- **Stemming**
- **Punctuation Removal**

Query operations also support the **exact phrase query** capability.

The system supports the **query expansion** by implementing **pseudo relevance feedback.**

Indexing component

BM25 model is applied to two different indices are created using PyTerrier:

- **Modified:** Stemming and Stopword Removal applied.
- **Non-Modified:** No stemming and Stopword Removal applied.

User Interface

The **tkinter** library was used to create the user interface.

The user interface has a search bar with the search results listed below it. Each document has a **name** and **a snippet** from the document. Also, the user interface supports **paging**. The query results are divided into pages with **10 documents per page** and the user can navigate back and forth between pages

Conclusion

In this project, we developed a small search engine using the DPR Wiki100 dataset, utilizing NLTK, Pyterrier, and tkinter libraries. We successfully implemented data parsing, indexing, and query processing features such as tokenization, stopword removal, stemming, and exact phrase querying. Although we faced challenges with the Pyterrier library and dataset size, we managed to create a functional user interface and ranking system based on BM25. Future improvements include adding support for wildcard queries and enhancing snippet relevance.