Introduction to Information Retrieval

Assignment 03: Text-Based Search Engine

Objective:

Develop a text-based search engine in Python that allows users to search for keywords within a collection of text data you crawled from the A1 assignment. The search engine should process text files, create an index for efficient retrieval, and rank results based on relevance to the search query.

Basic Requirements for the Search Engine :

1. **Programming Language**: Use Python for Backend.
2. **Libraries**: you are allowed to use Python libraries for text processing or vectorization such as:
    - `nltk` for text processing.
    - `scikit-learn` for vectorization (e.g., `TfidfVectorizer`).
    - Additional libraries are encouraged but must be documented in your report.
3. **Functionality (80%):**
    - **Indexing (40%):**
        - Tokenize text, remove stop words, and apply stemming/lemmatization to create an efficient index.
        - Support the use of TF-IDF for scoring terms in each document.
        - Store the index as a JSON file.
    - **Searching (40%):**
        - Ranking documents by relevance (euclidean distance/cosine similarity).
        - A graphical user interface (GUI) allows users to enter a text query and get matched results (top K).
        - Optional UI framework or library.
4. **Report (20%)**:

    Write a report (500-700 words) detailing:

    - Your approach to building the search engine (including tools, libraries).
    - Key challenges and solutions.
    - Examples of search queries and sample outputs (screenshots).

Advance Features (Optional but encouraged) (10%):

- Support for phrase searching (e.g., exact matches for multi-word queries).
- Boolean search support (AND, OR, NOT operations).
- Allow ranking customization, such as prioritizing recent documents.

Submission:

1. Python Code: Submit your Python scripts.
2. Indexed Data: Include JSON files or database files for the index.
3. Dataset: Submit a small dataset of text files used for testing.
4. Report: Submit the report as a PDF.