Assignment 1 - Group 37 [CSI 4107]

Team Members:

- Zanou Rih - 300178068
- Rita Tihani - 300206847

Overview

This assignment implements an Information Retrieval system using the Vector Space Model (VSM) and BM25 ranking,
based on the
Scifact dataset. The system processes queries, retrieves relevant documents, and evaluates performance using
Mean Average Precision (MAP).

Solution Design

1. **Preprocessing (preprocessing.py)**
    - Load the Scifact dataset
    - Tokenize and removes stopwords
    - Apply stemming for better normalization
    - Store processed tokens for indexing
2. **Indexing (indexing.py)**
    - Build an inverted index to map terms to documents
    - Compute document frequencies (DF)
    - Generate BM25 term weights
3. **Retrieval & Ranking (retrieval.py)**

- Use BM25 scoring to rank documents
- Apply pseudo-relevance feedback (optional improvement)
- Perform query expansion with synonym selection

4. **Evaluation (main.py)**
   - Run queries from `queries.jsonl`.
   - Produce results in `Results.txt`.
   - Evaluate using trec_eval to compute MAP.

Task Distribution

The workload for this assignment was split as follows:
Preprocessing & Indexing (Zanou Rih)
Retrieval and Evaluation (Rita Tihani)

1. **Preprocessing (preprocessing.py)**
   - Load the Scifact dataset
   - Tokenize and removes stopwords
   - Apply stemming for better normalization
   - Store processed tokens for indexing

How to Run the Code

Install Dependencies

```
pip install -r requirements.txt
```

Run the Information Retrieval System

```
python main.py
```

Evaluate Performance

We used WSL to use trec_eval, to evaluate the performance, simply navigate to the project folder inside WSL and run this command:

```
trec_eval -m map relevance.txt Results.txt
```

Instead of using the test.tsv directly, we adjusted the format of the file for `trec_eval` to be able to process it, and renamed it `relevance.txt`

Algorithms, Data Structures & Optimizations

Algorithms Used

- **BM25 Ranking**: Improves retrieval effectiveness by adjusting for term frequency and document length.
- **Vector Space Model (VSM)**: Uses **cosine similarity** for ranking.
- **Pseudo-Relevance Feedback**: Expands queries based on top-ranked documents.
- **Query Reformulation**: Replaces query terms with synonyms for better recall.

Data Structures Used

- **Inverted Index (Dictionary of Lists)**: Maps terms to document occurrences.
- **TF-IDF & BM25 Matrices (NumPy Arrays)**: Stores term weights efficiently.
- **Dictionary-Based Query Expansion**: Maps words to synonyms dynamically.

Optimizations Implemented

- **Stopword Removal & Stemming** (Reduces vocabulary size).
- **BM25 + Cosine Similarity Scoring**.
- **Re-ranking using Feedback-Based Query Expansion**.

Sample Results

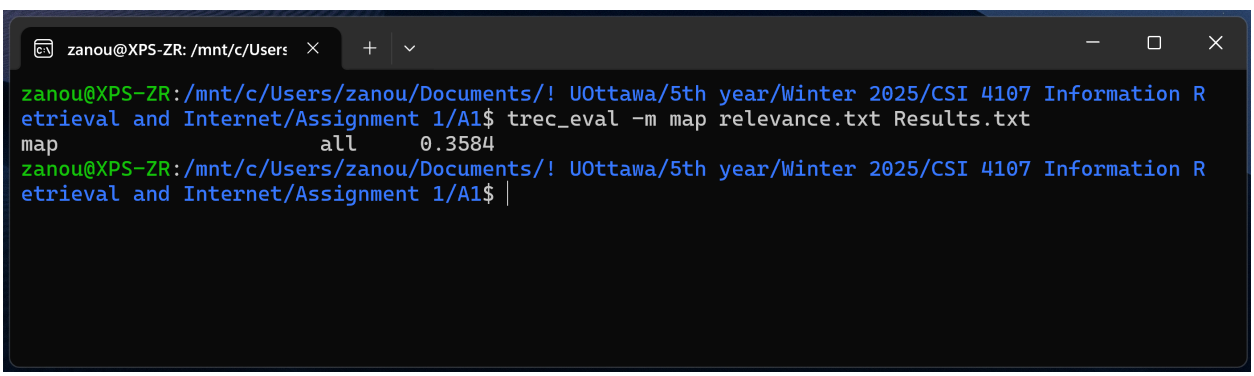
First 10 Answers for First Two Queries

Query 1:
['24700152 (Rank: 1, Score: 133.8348)', '44265107 (Rank: 2, Score: 102.5543)', '1886551 (Rank: 3, Score: 100.3052)', '6477536 (Rank: 4, Score: 93.9739)', '6112053 (Rank: 5, Score: 86.5782)', '14647747 (Rank: 6, Score: 82.3769)', '750781 (Rank: 7, Score: 77.2211)', '9056874 (Rank: 8, Score: 76.2498)', '14376683 (Rank: 9, Score: 74.8042)', '45461275 (Rank: 10, Score: 73.7424)']

Query 2:
['25510546 (Rank: 1, Score: 191.6107)', '8453819 (Rank: 2, Score: 164.0747)', '38477436 (Rank: 3, Score: 140.9763)', '29459383 (Rank: 4, Score: 136.1833)', '35345807 (Rank: 5, Score: 98.0626)', '145383432 (Rank: 6, Score: 98.0626)', '19561411 (Rank: 7, Score: 83.1686)', '5687200 (Rank: 8, Score: 66.9564)', '7198295 (Rank: 9, Score: 66.9564)', '28271439 (Rank: 10, Score: 57.3407)']

Results & Discussion

- **Current MAP Score:** 0.3584



- The IR system performs relatively **well for keyword-based queries**, but struggles with **semantic understanding**.
- **BM25 ranking improved results**, but further refinements are needed to improve the score

---

References

- **Scifact Dataset**: https://github.com/allenai/scifact
- **TREC Eval**: https://github.com/usnistgov/trec_eval