Final Project Part A (70%) – Ranking Competition

Homework Submission Guidelines

1. **Due date: 22/01/26**
2. The challenge can be done in groups of **up to** 2 students
3. **Submission instructions at the bottom**
4. Submission should be done via Moodle in the corresponding area (by **only** one of the students)
5. **Late submission will not be considered**

Background

Part A of the final project for the "Text Retrieval and Search Engines" course is a hands-on competition designed to apply the concepts and techniques learned throughout the semester. In this project, you will be tasked with building the most effective search engine and evaluating its performance against your peers.

Each team will implement three different information retrieval methods, where at **least one** of these methods must go beyond the material directly taught in class, requiring you to research and implement an advanced or alternative retrieval technique independently. Examples could include neural ranking models, learning-to-rank techniques, hybrid approaches, fusion of several models, etc. The goal is to explore and compare different retrieval strategies to determine which is the most effective in retrieving relevant documents from a given dataset.

You will require experimenting with different parameters, conducting evaluations, and analyzing the trade-offs between all your selected approaches so eventually you will submit 3 different retrieval attempts.

The competition will encourage innovation, will provide valuable experience in building end-to-end information retrieval systems, allowing you to see firsthand how various retrieval techniques can influence the relevance and quality of search results.

The project not only tests technical proficiency but also fosters creativity and adaptability—key skills for success in the rapidly evolving world of search technologies. As such, your grade will be determined based on **the quality of your system** and based on **creativity and knowledge of the suggested ranking models**.

HAVE FUN!

The challenge

Inside the **Final_Project_A/files/** on Moodle:

a) "queriesROBUST.txt" – 249 queries.
b) "qrels_50_Queries" – the ROBUST relevance judgments for the first 50 queries.

The collection for retrieval is ROBUST and is available using the following pyserini commands:

```
from pyserini.index.lucene import IndexReader
from pyserini.search.lucene import LuceneSearcher
index_reader = IndexReader.from_prebuilt_index('robust04')
lucene_searcher = LuceneSearcher.from_prebuilt_index('robust04')
```

Note – the index was created using Porter stemming and no stopword removal.

Goal:
Achieving the highest retrieval effectiveness as measured using **MAP**.

Task:
There are 249 queries, but we provide the relevance judgments for the first 50 queries only (50 queries - train, 199 queries – test). You are required to return a ranked list of documents from the collection ordered in the decreasing order of relevance.
For each test query (of the 199 queries), you submit a ranking of the top scored **1,000 documents**. Your retrieval will be evaluated based on the 199 test queries.

The relevance judgments for the first 50 queries can be used to train your retrieval methods, adjust free parameters, and compare the effectiveness of different ranking models.

Deliverables

1. You **must** submit **3** different result lists (runs).
   Each file name is of the form: run_i.res (i is set to a value in {1,2,3}).
   The format of each file is the standard 6 columns TREC format:

```
630 Q0 ZF08-175-870  1 0.7 run1
630 Q0 ZF08-306-044  2 0.5 run1
630 Q0 ZF09-477-757  3 0.3 run1
630 Q0 ZF08-312-422  4 0.1 run1
630 Q0 ZF08-013-262  5 -0.3 run1
 etc.
```

   where:

   - the first column is the topic number.
   - the second column is currently unused and should always be "Q0".
   - the third column is the official document identifier.
   - the fourth column is the rank the document is retrieved
   - the fifth column shows the score (integer or floating point) that generated the ranking. These scores MUST be in descending (non-increasing) order.
   - the sixth column is called the "run tag" (can be set to your run_i)

All runs must be **compressed (zip)** and named:
Final_Project_Part_A_Student_1_email_Student_2_email.zip

2. Each team will be required to prepare and deliver **(in person)** a presentation on **Tuesday, January 27, between 17:00 – 20:00**. This presentation will provide a concise overview of the retrieval methods implemented, the rationale behind their choices, and any notable findings or challenges encountered during the project.

   Key instructions for the presentation:

   a. **Presentation Content**: Your presentation should clearly explain the three retrieval methods implemented, with a special focus on the method not directly taught in class. Highlight any innovative aspects of your approach and key performance results.

   b. **Duration**: Each presentation should be approximately 5 minutes, followed by a 2-minute Q&A session.

   c. **Assessment Criteria**: Your grade will be based on the clarity and quality of your presentation, your ability to answer questions effectively, and your understanding of the material.

   Important Note: Attendance during the final lecture is **mandatory** for all students. This ensures an engaging and collaborative environment where everyone can learn from their peers' experiences and methods. **There is no need to submit your presentation slides; simply prepare to present them in class.**

Tools:

1. Pyserini toolkit for interacting with the index.
2. You can write your own ranking algorithm using any programing language that you feel most comfortable or to use those appeared in HW.
3. **Important:** your algorithm must be reproducible.
4. The project is designed to be completed using the freely available resources on colab. As the resources for GPUs are limited, use it wisely.
5. Tips for GPU usage (if needed):
   - Before connecting to GPU, make sure your code executes without errors on CPU
   - Connect to the GPU runtime only when you are ready to run your code.
   - Disconnect from your runtime if you are going on a break.

Good Luck