Project 1: Document Retrieval

Information retrieval involves searching through a collection of documents to find and return those most relevant to a given query. This project centers on the task of text retrieval. Your objective is to rank documents according to their relevance to a specific query, selecting them from the entire provided document collection.

Dataset

We provide you with a corpus of 199334 documents covering 7 different languages: English, French, German, Spanish, Italian, Arabic, and Korean.

Split	# of examples
corpus	268022
train	21875
dev	1400
test	2000

In the dataset folder, you find the following files:

- "`corpus.json`**:** This file contains a collection of documents we want to retrieve from. Each document has a unique id ("docid"), its content ("text"), and the language of the document ("lang").
- "`train.csv`**:** This file is the "train" split of the data containing queries and their relevant documents. The data contains the following information:
  - "query_id": a unique id for each query
  - "query": the text/content of the query
  - "positive_docs": a relevant (positive) document to the given query
  - "negative_docs": a list of irrelevant (negative) documents to the given query
  - "lang": the language of the given query
- "`dev.csv`": This file is the "dev" split of the data containing queries and their relevant documents.
- "`test.csv`": This file is the "test" split of the data containing the queries for which you need to retrieve documents.

Project Evaluation

- 40%: Results - Metrics
- 30%: Code
    - Working code (20%)
    - Code quality and documentation (10%)
- 30%: 2-page Report
    - Originality of approach (10%)
    - Interpretation of results (10%)
    - Report presentation & clarity (10%)

Regarding the Results part, we evaluate the performance of your implementation using two criteria:

1. Computation time (T): One evaluation criterion is the time it takes for your code to retrieve documents for the given test set. If this time is less than a **T0 = 10mins**, you get the full points of this criteria, and for computation time more than T0, you get penalized using this formula **max(0, 1 - T0/T)**. In other words, your score for this part is computed as **1 - max(0, 1 - T0/T).** This criterion counts for **1/4** of your grade for the Results part.

2. For each query given in the test set ("`test.csv`"), you have to retrieve 10 relevant documents from the given corpus.
   We evaluate the performance of your retrieval system using the **Recall@10** metric.

   This criterion counts for **3/4** of your grade for the Results part. From this portion, the last 10% of the teams in the Kaggle competition get 4/6 of the grade, the top 10% get 6/6, and the groups in the middle get 5/6.
   We ran a simple baseline with a final score of **0.50** to give you a better idea of where your approach stands. **You must achieve this performance at least to be considered in the grading explained before.**

Deliverables:

- You submit the retrieved documents for the given test set through Kaggle
- You submit your code on Moodle
- You submit your report on Moodle
    - Please explain the originality of your approach and the intuition behind your design choices, as well as explain and interpret the results. Please report the performance of your models per language as well as an average of all languages.

Requirements:

- You can use any supervised or unsupervised retrieval methods in the project. However, for the supervised methods, you have to train the model yourself (using the given labeled data), and you are not allowed to use any trained models that are publicly available.
- You can use pretrained language models to embed documents and queries.
- You are not allowed to use TF-IDF implementation from sklearn or any other libraries.
- We run your code on Kaggle, so make sure that:
    - Your Kaggle notebook is public (only after the deadline)
    - Your code is ready to run, and we can access all the necessary files.

Submission

Data distribution, running codes, and submission are all done using the Kaggle platform. Please create a Kaggle account with your **EPFL email.** Similar to colab, you can run notebooks (with or without GPU) on Kaggle to investigate data, as well as the final **submission** notebook. This notebook should write the submission file as one of the outputs. We will give more details about the final deliverables during the Thursday exercise session. In order to join the competition (with your **EPFL email Kaggle**), please use this invitation link.
Note: Please don't make your notebooks public **throughout the competition**, and keep them **private**!