About

This project aims to enhance the Information Retrieval (IR) system by implementing various Document Ranking models. The primary focus is on exploring multiple ranking approaches, including the Probabilistic Model, and Language Model. For this project, we will utilize the MS MARCO dataset as our main source of queries and documents.

Instructions

- **Document Preprocessing**: Your project will begin by reading and preprocessing a collection of text documents from the MS MARCO dataset. For each document, only the relevant text will be retained, along with essential tags or metadata. This dataset also includes queries and relevance judgments for each query, which will help evaluate the performance of the document ranking models during the evaluation phase, however, you just need the document and query text and whether the document is selected or not.

- **Document Ranking – Probabilistic Model**: You will implement a function for document ranking using the Okapi BM25 basic weighting approach. This function will take a query text and an integer specifying the number of top documents to retrieve. You do not need to fine-tune parameters b, k_1, and k_3, but you should explain why the default values make sense for this application. Additionally, to handle long queries more effectively, the function can optionally switch between different Okapi BM25 approaches based on query length, optimizing the ranking results for varying query sizes.

- **Document Ranking – Language Model** In this component, you will develop a function for ranking documents using a language model approach. This function will take a query text and an integer that specifies the number of top-ranked documents to retrieve. To handle the issue of zero probabilities, you can apply either Dirichlet or Jelinek-Mercer smoothing techniques. While fine-tuning of the smoothing parameters (for Jelinek-Mercer and for Dirichlet) is not required, you should provide a rationale for the chosen smoothing method and explain why it is suitable for this project.

- **Document Ranking – Hybrid Model** This component will combine the strengths of the Vector Space Model, Probabilistic Model (BM25), and Language Model to create a hybrid ranking function. The hybrid model will take a query text and an integer specifying the number of top documents to retrieve. By integrating the scoring mechanisms from each individual model, this approach aims to enhance retrieval effectiveness by leveraging the complementary strengths of each model. Different weighting schemes will be explored to balance the contribution of each model, and the final configuration will be optimized based on evaluation metrics. The hybrid model is expected to provide a more robust and accurate ranking of documents across diverse types of queries.

- **Comparing Document Ranking Models** You will evaluate and compare the effectiveness of these three document ranking approaches using the evaluation techniques introduced in Lecture 7. A set of queries and their relevant documents are provided, which should be used during the evaluation process. It is recommended to use 11-point interpolated average precision for a subset of queries, but other evaluation metrics may also be applied, and multiple queries should be tested for thoroughness.

Deliverables

1. Four functions handling document ranking as described. (Functions can be in *.py* files, however they need to be called in a *.ipynb* file which is your main file)
2. A comparison of these models based on an evaluation function on the provided query-relevant document pairs.
3. Comprehensive documentation describing the functions' architecture and components.
4. A report summarizing key findings, challenges faced, and enhancements made to the IR system during the project's development (you may append this to your previous reports).

Submission:
Please submit your answers to the Quera containing *.ipynb* file, *.py* file, and a single *PDF* file for codless answering.