IR Mini Project No4

Supervisor: Dr. Baba Ali

Yazdan Zandiye Vakili

Introduction

This project aims to implement document rankings based on 3 approaches that I explain below

Vector Space Model

In this model I vectorized documents and given query with TFIDFvectorizer in scikit-learn and after that calculated the cosine similarity between them and at last sorted them with using a max heap and returned top k documents with their corresponding cosine similarities scores in desired format of project.

Probabilistic Model

In this model I implemented BM25 probabilistic algorithm myself (there was pre-built models in libraries) to handle the hyper parameters of it and defined a threshold (10 words) for given query to switch between two versions of it based on the length of the given query.

Language Model

In this section I designed two language models, one light version which uses unigrams only and calculates the scores of each document based on unigrams and one heavy version that calculates trigrams, bigrams and unigrams and calculate the final score as below:

$$Score = 1 \times Unigram_{Score} + 5 \times Bigram_{Score} + 10 \times Trigram_{Score}$$

All the results of the k-grams smoothed by Dirichlet smoothing algorithm implemented by myself in this section and at the end returns the documents and corresponding scores to them.

Evaluation

In this part based on the query, I finds query's id and based on that and results, I calculates relevant documents. After this step recall precision pairs calculated in a separate method in this class and after that the results of this method returns in a method that computes the K points interpolated average precision and returns it to main.ipynb and by storing the values of them, in main.ipynb, I reported Mean Average Precision too.

Note

Because of the long computation time, only the first 10 queries tested on each model.