INFORMATION RETRIEVAL: DEFAULT PROJECT PARTS II & III

SHRUTI SHREYASI 19EC10086 GROUP 1

PART II

A. TF-IDF Vectorization

1. tf, df computation:
    a. On the updated model_queries_1.bin file, computed and stored term frequency and document frequency in a dictionary structure. Pickled and saved the dictionaries to save memory. Deleted all the dictionaries when they were not needed.
2. TF-IDF vector computation:
    a. Wrote the code for implementing lnc.ltc, Lnc.Lpc and anc.apc TF-IDF vectors. Faced with the issue of memory limit exceeded when processing the vectors document first, hence switched to Kaggle kernel as it has more RAM.
    b. For overcoming the problem of memory exceeded, the computation of TF-IDF the computation was done query wise and the matrix was not stored (size N*V) and only (size Q*V) was stored, where N is the number of documents, V is the vocabulary size and Q is the number of queries.
    c. We are not saving the TF-IDF vectors for the top documents for part 3 as that computation is not a determining step in the time complexity and can be recomputed.

B. Evaluation:

Wrote argument parser and helped in debugging in the NDCG part. The error was caused by passing a wrong list to the function NDCG_k. The same functions are also called in the Part III of the assignment.

Main problems faced:

Memory error while storing the TF-IDF vectors: this was handled by reading the vectors query wise and deleting the redundant dictionaries.

PART III

A. Relevance feedback:

1. Rocchio's algorithm for modified query vector:
    a. Wrote an argument parser for reading the paths of the files as given by the user.
    b. Implemented the function pick_ranked_relevant_docs which takes the parameters alpha, beta and gamma as parameters. It returns the ranked relevant documents for a query on the scale 0-2.
    c. Implemented the function for Rocchio's algorithm which takes as input the parameters alpha (a), beta (b), gamma (c), pseudo. Pseudo is by default false and it can be made true in the case of pseudo relevance feedback.
    d. a, b, c are converted to float types because not doing that was resulting in error due to different data types
    e. For each document, its relevance is noted (judgment value 2 is taken to be relevant and the rest to be relevant) and it is added to the query vector after scaling it with the appropriate constant.
2. Relevance feedback (RF) and Pseudo Relevance feedback (PsRF):
    a. Average precision and NDCG values are computed from the evation functions used in Part II for the given values of alpha, beta and gamma.
    b. The mean of evaluation metrics are computed and stored in csv files.

B. Important words:

1. TF-IDF vectors and computing the most important words:
    a. Argument parser was written and comments were written for the file
    b. The TF-IDF vectors were recomputed for each of the top 10 relevant documents for each query. This part remains the same as in the previous part.
    c. The mapping of document frequency for all terms was loaded and the index to term mapping was loaded. This part was not pickled before so it had to be added to the previous part.
    d. For each query, the top 10 documents were taken and the average of their TF-IDF vectors was taken. The TF-IDF vectors were sorted and the top 5 terms were noted. A very interesting observation is that 'Coronavirus' occurs in most of the documents as the most frequent word.