INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

Computer Science and Engineering
INFORMATION RETRIEVAL (CS60092)
Assignment - 3 (Autumn 2022)

Group: 9
Altaf Ahmad
18MA20005

Report

1 Parts Completed

I was responsible for completing part 3A(Relevance Feedback) of the Term Project. For this purpose, I first refactored the code from part 2 for calculating the TF-IDF vectorization of the documents and the queries using lnc.ltc scheme. After this, the query vector needed to be modified using Rocchio's algorithm with the data from the gold standard documents (qrels.csv). Using this, I extracted all the relevant documents, which had a gold standard judgment score = 2, and marked the rest of the documents as non-relevant. Then, for each of the query vectors, I modified them using the Rocchio's algorithm as follows:

$$q_m = \alpha q_0 + \beta \frac{1}{|D_R|} \sum_{d_j \in D_R} d_j - \gamma \frac{1}{|D_{NR}|} \sum_{d_j \in D_{NR}} d_j$$

Where q_m is the modified vector, q_0 is the original vector, D_R and D_{NR} are the sets of relevant and non-relevant documents respectively. $\alpha, \beta,$ & γ are the hyper-parameters whose values are given as :

Index	α	β	γ
0	1	1	0.5
1	0.5	0.5	0.5
2	1	0.5	0

Now, using the modified queries, I was able to find the cosine similarity in the documents and then rank them based on their scores. After this ranking, using the MAP@20 and NDCG@20 Evaluation scoring method from 2B. I also helped a little bit in debugging some parts of the evaluation.
Similarly, I was able to modify the queries from the Pseudo Relevance feedback as well, using the top 10 ranked documents as relevant and the set of non-relevant documents as **null**.

2 Challenges Faced

The major challenge was memory management as computing the TF-IDF vectorization of more than 50,000 documents needed a lot of memory to run and we tried running it directly on 8 GB RAM, which wasn't working. So we had to run it on a more powerful machine. We also used garbage collection as effectively as possible to drop all the data frames, dictionaries, and arrays that were not used.
Another problem was that there were too many words in the vocabulary so, for many terms the values of the TF-IDF vector for these words were 0 for all the documents. Therefore, we removed all these terms from the dictionary.
Other than that there were some minor issues while calculating the NDCG scores, there were some queries for which no documents matched the gold standard documents which led to NaN values in the DCG. Due to this there was some edge cases too where division by 0 was going on which had to be taken care of.

3 Design

The TF-IDF vectors were stored in dictionaries with their cord-ids as keys so that they can be easily fetched, identified, and isolated when required. This helped in looking them up in average constant time. The vector operations like updating the query using Rocchio's algorithm and finding the cosine similarity were done using NumPy arrays instead of iterating over them which improved the time.