REPORT
ASSIGNMENT-`4

- **Data Preprocessing Steps:**

- Stop Word Removal
- Lemmatization
- Removal of words of length less than 3.
- Removing empty lines
- Removing multiple spaces between words
- Removing Special Characters
- Lower case conversion

- **Methodology For Question-1:**

- Firstly, I read all the documents from the corpus (mentioned folders) and constructed the posting lists for every term in the vocabulary of the corpus. The posting list contains the term frequency of that particular term in every document.

- I built a document-frequency table which contains the document frequency of every term in the vocab.**I used the log-normal variate of TF (tf = 1+$\log_{10}$(tf)) and inverse variation of DF (df = $\log_{10}$(N/df)) where N is the total number of documents considered for the corpus.** Then i had computed tf-idf values by multiplying corresponding term frequencies by document frequencies.

- When the query is given by the end-user,the tf-idf vector is built for it by first pre-processing the query and then considering the term-frequencies of terms in the given query and document frequencies of the term in the corpus. (both are used with their variates).

- Now for every doc the tf-idf vector is built by considering only the words in the query by preserving the order of words. So, now the size of every vector equals the size of the query vector.

- Now, we compute the cosine similarity between the query vector and all the docs and store the corresponding results in the dictionary, with key being “folder-name”_”doc-name” and value being its corresponding cosine-similarity.

- Now according to the K value entered by the user we retrieve top-k docs from the dictionary with their corresponding cosine-similarity scores.

$$\textbf{cosine-sim(A,B)} = \frac{\textbf{Dot-product(A,B)}}{|A| * |B|}$$; A and B are vectors of the same size.

- **Question-2:**

- **Assumptions:**

- I assumed that if an end-user enters multiple queries, the number of times (iterations) he embeds feedback into the model is equal for all the given queries or else does not embed the feedback at all for all the given queries.

- The docs which are considered as relevant in previous iteration are neither considered as relevant **(** instead we search for other relevant docs in the top-k which are not considered as relevant till then, to push them up in the order **)** nor considered as irrelevant.

- **Methodology:**

- In the continuation to above question we prompt the end-user if he wants to embed the feedback into the model. If yes, we also prompt him to enter the number of times he wants to embed the feedback and also the name of the directory from whose documents needed to be considered as relevant from the obtained top-k results. Then we separate the list of relevant docs and irrelevant docs. The docs which are considered as relevant in the previous iterations are not considered as relevant in the upcoming iterations.

- Now, we compute the centroids of vectors of the relevant-docs and irrelevant docs that we have marked in the above step. According to the Rocchio algorithm, the new query vector will now be,**(alpha)*(old_query_vector)*+ (beta)*(centroid of relevant docs) - (gamma)*(centroid of irrelevant docs).** The old_query_vector keeps updating every iteration.

- Then we again compute cosine-sim between the obtained new query vector and all the document vectors and store the results in a dictionary as done previously and print top-k.

- After retrieving the results at the end of every iteration, I calculated the **average precision**, precision at different points of recall and **plotted the precision recall curve**. I also **plotted the T-SNE for the query vector** after every iteration in comparison with the query vector without any embedding of feedback.

- I also calculated the **MAP (Mean Average Precision)** for all the given queries at the end of every iteration.

$$\textbf{Average precision} = \frac{\textbf{(sum of precisions after every doc retrieval)}}{\textbf{(Total No, of relevant docs)}}$$

$$\textbf{Mean Average Precision (MAP)} = \frac{\textbf{Sum of Average Precisions at iteration i for given queries}}{\textbf{(Total No. of Queries)}}$$

MAP VALUES FOR THE GIVEN THREE QUERIES:

==

Query 1: Pretty good opinions on biochemistry machines
Relevant set 1: Documents inside folder sci.med

Query 2: Scientific tools for preserving rights and body
Relevant set 2: Documents inside folder talk.politics.misc

Query 3: Frequently asked questions on State-of-the-art visualisation tools
Relevant set 3: Documents inside folder sci.med

==

MAP WITH EMBEDDING OF FEEDBACK FOR GIVEN 3 QUERIES AFTER ITERATION-1 IS 0.3814436372167576

==

MAP WITH EMBEDDING OF FEEDBACK FOR GIVEN 3 QUERIES AFTER ITERATION-2 IS 0.38929241481125615

==

MAP WITH EMBEDDING OF FEEDBACK FOR GIVEN 3 QUERIES AFTER ITERATION-3 IS 0.42072973658068485

==

MAP WITH EMBEDDING OF FEEDBACK FOR GIVEN 3 QUERIES AFTER ITERATION-4 IS 0.4496606578200983

(other plots and results mentioned in Analysis file)