ASSIGNMENT-03

How to Run the Code?

1. Extract the files of 20_newsgroups and file.txt and IR-assignment-3-data.txt.
2. Must have NLTK.
3. Open Jupyter Notebook and make sure that files are available in the same folder as jupyter notebook and then start running the code from the beginning.

Assumptions:

1. Query terms are not repeated.
2. All queries words spellings are correct.
3. No word in present according to mobile dictionary.

Q1.

Preprocessing Steps:

Normalization:

- converting all text to the same case (upper or lower).
- removing punctuations

Stop Words:

- We may omit very common words such as the, a, to, of, be from unigram inverted index not from positional index.

Stemming:

- Used porter stemmer to stem the words. It is faster than lemmatization and does a good enough job of stemming related words to the same stem.

Tokenization:

- Cut character sequence into word tokens.

Num2Words:

- Convert number to words.

Single Character:

- Remove a character of length 1.

Removal of Header.

Methodology:

1. *Loaded documents using i.e 20_newsgroup folder and file.txt.*
2. *Maintained a list named* **g** *containing the static scores.*
3. *Sorted static scores in a dictionary named* **static_score**.
4. *Preprocessing of data and query is done using* **preprocess_data** *and* **preprocess_query** *methods respectively.*
5. *A dictionary is created* **dic** *for storing the paths of each file.*
6. *Made a dictionary named* **text_dic** *containing docId, tf and static score for each word in a corpus.*
7. *Pickle a text_dic and stored in Assignment3_Q1.pkl.*

8. **tf_dic** *is maintained having tf values for each term in a corpus.*
9. **GetMaxFlow** *and* **GetMinFlow** *methods are used for computing maximum and minimum length posting list,term and it's length.*
10. **OptimalR()** *method is used for finding optimal 'r' documents.*
11. *Another dictionary is made* **text_dic_new** *for storing high and low list which are made on the basis of r.*
12. *Then other 2 methods are used for r by entring r at run time or by computing r for dynamically for each term.*
13. *User enter the query. Then is query is preprocessed and for every query term high and low list is fetched and union of all query term is taken out using* ***union*** *method.*
14. *Then user enter the value of k and top k documents are fetched then net score of those documents are computed.*

Q2.

No Preprocessing is done.

Methodology:

1. *Loaded a file named IR-assignment-3-data.txt.*
2. *All the below work are done only for* **qid:4**.
3. *Created a list storing releveance score named* **relevance_score**.
4. *Then find the number of documents having relevance score 0,1,2,3,4 in rel_score_0,* **rel_score_1**, **rel_score_2**, **rel_score_3**, **rel_score_4**.
5. *Sort the relevace score and stored the values in* **rel_sorted** *list.*
6. *Created a file storing maxDCG named* **maxdcg_content.txt**.
7. *Find number of such files that are possible for maxDCG.*
8. *Computed DCG, IDCG at 50 and at whole dataset.*
9. *Created a list storing URL named* **feature_75**.
10. *Then precision and recall is computed.*
11. *Plotted a PR curve.*

References:

- *https://stackoverflow.com/*