Assignment -2

ReadMe

Ans 1)

To retrieve the documents containing the query, following are the steps taken:-

1. Files in the stories folder are retrieved, cleaned, lowered, tokenized, stopwords removed, and lemmatized.
2. Positional Index is made, in which dataFrame is created, for every column is a word(term) and 0th row is the number of documents containing the word/term, and 1st row is a dictionary in which keys are document id and values are positional index of word in the document.
3. The query is also cleaned exactly in the same way as with documents.
4. Documents are retrieved where tokenized query word's index in the documents are present consecutively.
5. Finally from the document id retrieved, the name of the document containing the query is printed.

Since cleaning will remove all digits both from queries and documents , thus queries containing digits will not be matched accurately. And the same would be the case for special characters.

Ans 2)

JACCARD :

Document dictionary is taken as input and sent to a function named jaccard_on_documents().

Each document is sent to a function named jaccard_similarity(), where lists are converted to sets and their intersection is divided by their union and documents are sorted in descending order of their jaccard score and shown as output.

TFIDF :

Dataset description : Postings list created in Part1 is used . Using posting list dataframe document frequency and term frequency can is determined for each entry.

Firstly a vocab of all words is created and a vocabulary list for query. Using these two data structures all five notions are calculated. As an output to these notions a vector of vocab length is obtained. For the query a TFIDF vector is obtained and similarly for each document vectors obtained are stored and hence a matrix is obtained for the TFIDF of documents which has the size of Number of documents * Vocabulary length. After obtaining these Vectors and matrices we calculate the score and display the top 5 documents for each weighing scheme.

COSINE :

Document dictionary is taken as input and sent to a function named jaccard_on_documents().

Each document is sent to a function named jaccard_similarity(), where lists are converted to sets and their intersection is divided by their union and documents are sorted in descending order of their jaccard score and shown as output.

Ans 3)

Methodology –

Dataset Description – The dataset used is Microsoft Learning to Rank dataset. In this, queries and urls are represented by IDs. The dataset contains feature vectors extracted from query-url pairs along with the relevance judgement label. The relevant judgements can take values from 0 to 4. 0 represents irrelevant and 4 represents perfectly relevant. The dataset has 32516648 query-url pairs. The sample query-url pair is shown as-

```
0 qid:4 1:3 2:0 3:2 4:0 5:3 6:1 7:0 8:0.666667 9:0.....134:0 135:0 136:0
```

The first column represents the relevance label, second column is query id and following columns are features. Only those query-url pairs are taken where qid is 4. Such pairs are 103. The relevance judgement labels are selected as relevance scores.

To calculate the maximum Discounted Cumulative Gain, query-url pair is arranged in decreasing order of Relevance. Such pairs are written in a queryUrlPairFile.txt file. Then calculated Normalized Discounted Cumulative Gain for the whole dataset and for rank = 50.

Then calculated precision and recall considering non-zero relevance judgement values to be relevant pairs and precision-recall curve is plotted.

Assumptions –

- The query-url pairs are ranked according to feature id 75 which corresponds to sum of tf * idf for the whole document.
- For the query-url pair where relevance judgement values is non-zero are considered as relevant.