CSE508 IR Assignment 1
Group 41

1. Assumption:

- In the Stories folder we excluded index.html files, SRE folder and FARNON folder so we have a total of 452 files of different extensions.
- We uploaded the stories folder on google drive and mounted into the google colab.
- DataSet Link
- Environment - Google Colab
- Language - Python

2.Preprocessing Step:

The preprocessing done on the textdata are as follows:

- **Convert Lower case:** All data is converted into lower case.
- **Replacing digits with their english form:** replacing digits of text data with their corresponding english word like '0' by 'one', '1' by '2' etc.
- **Remove Punctuation** means remove punctuation special character like "/!@#%^&*()_<>:"{}[];',./' from textdata.
- **Tokenization:** basically divides data in the form of tokens i.e into words.
- **Remove StopWords**,first we download the english stopword(and,the,or etc) from nltk library and remove from textdata.
- **Remove Single Character** we removed those words whose length are less than 1.
- **Apply Lemmatization** first we import word lemmatizer then break text data into tokens and after apply lemmatization on each word.

Also then we created a DataFrame with all the file names with index as a Document ID that will be used for further printing list of document names retrieved for given Query.

3.Methodology:

- Posting for preprocess textdata:
  - Posting list is a list which contains all unique words and all docid in which those unique words are present.
  - First perform preprocessing of each file in the stories dataset and do tokenizer steps.
  - Iterate all token and check if token is present in posting list then update the docid for that token otherwise make update posting list of new word of tokens.
  - Repeat the above process for all the files.

  Transpose of Posting DataFrame create shown with **43788 unique Word**.

shizophrenia	{451}
plurality	{451}
gollum	{451}
hehehehe	{451}
aftershock	{450}
...	...
man	{0, 2, 3, 5, 8, 11, 13, 14, 15, 16, 18, 20, 21...
brunet	{0}
rick	{0, 417, 130, 134, 301, 318}
one	{0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14...
contradiction	{0, 448, 34, 228, 169, 89}

43788 rows × 1 columns

- Input Query and Operations form users.
- PreProcessing Input text Query.
- Method 1 (Left To Right):
  - Every Time Iterate over preprocess Query text and store the all unique words doc id list in sorted order on word_docids.
  - We take two operand means word docId from left to right and perform operation left to right which is given by user input.

    - The operations are ‘AND’ means intersection of two list, ‘OR’ means union, ‘AND NOT’ means intersection of operand1 and not of operand2, ‘OR NOT’ means union of operand1 and not of operand2.
    - Return length of document match, no of comparison and name of document matched.

- Method 2 (Select word with Minimum DocID):
    - Every Time Iterate over Preprocess Query text word with Minimum size of DocIds in posting List and select its left word and perform Query AND, OR, AND NOT, OR NOT accordingly same mention above and if left word does not exist then select right word.

4.Result:

- Given 1st Query Result:

```
Input Query:  lion stood thoughtfully for a moment
Input operation sequence:  ['OR', 'OR', 'OR']
Number of documents matched:  263
No. of comparisons required:  670
['nitepeek.sto', 'timem.hac', 'blind.txt', 'tree.txt', 'b
```

- Given 2nd Query Result:

```
Input Query:  telephone,paved, roads
Input operation sequence:  ['OR NOT', 'AND NOT']
Number of documents matched:  331
No. of comparisons required:  870
['contrad1.hum', 'timem.hac', 'cameloto.hum', 'beyond.hum',
```

-