Boolean Retrieval System

Kushagra Singh 2019B1A70994H
Raj Srivastava 2019B1A71426H
Abhishek Jalan 2019B1A71547H

Design Document

Application architecture :

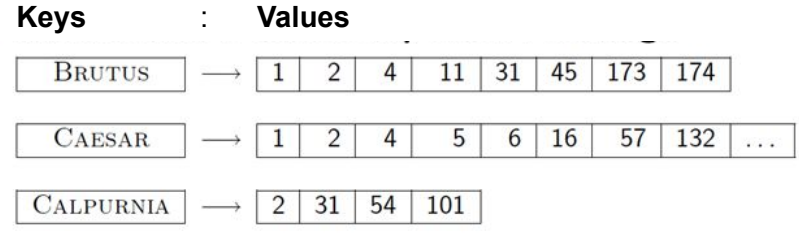
The application contains two python files:

1. **final.py**
   The file contains code to take in the database and runs the inverted index function and ends with running the query.

2. **functions.py**
   Contains all the functions to be used in the application

Central Data Structure:

Dictionary of key-value pairs
key as words (inclluding different permuterms)
values as set of documents that contain the key



Running times

1. Stopword Removal:

   If stopword removal takes place in a sentences we use O(n) complexity where n is the size of the sentence.

   But since here we are traversing through every file and in every file through every sentence. Each sentence takes O(n) complexity. Removing stopwords while creating the inverted index.
   The complexity becomes :
   O(f*m*n)
   Where f=number of files,m=number of lines in each file
   n=number of words in each line.

2. Stemming/Lemmatization:

   Stemming of a particular word takes O(1) time.
   But since here we are traversing through every file and in every file through every sentence. Each sentence takes O(n) complexity.
   The complexity becomes
   O(f*m*n)
   Where f=number of files,m=number of lines in each file
   n=number of words in each line.

3. Building Index:

   O(f*m*n)
   Where f=number of files,m=number of lines in each file
   n=number of words in each line.

4. Querying:

   O(qs)
   where qs= size of query

5. Search/Retrieval :

   O(n)
   Where n denotes the number of keys in the dictionary