IR ASSIGNMENT 1

ANAND SHARMA (MT19059)

READ ME FILE

Question (1):-

PREPROCESSING

(1) used os library of python and listdir to traverse all 20 subdirectory and files under each subdirectory
(2)open all 19997 files and converted into tokens
(3)used nltk for preprocessing of files data
(4)Used RegexpTokenizer ffrom nltk.tokenize to convert each word into tokens
(5)removed all numbers from files
(6)converted all tokens word to lower case

same all steps are applied to the input boolean string for retrieval of documents

ASSUMPTIONS

(1) Not removed the stopwords .user can give input of query including stopwords
(2)I have assumed that user have to give query input in proper format such as exp operator exp and just like a boolean query
(3)User cannot give boolean query words which are not present in a 20000 files
(all query exp has to be present in files used as dataset)
(4)Not performed stemming or lemmatization on tokens and datset

Methodology

(1)Build an Unigram inverted Index using dictionary where key is the query word and value is the list of doc id in which query word is present
(2)first operated on all not operator present in query and save their doc id in list
(3)then do the optimization part which is first perform and operation on small retrieved list when compared to large retreived list so i have sorted the list based on size of the list and append on correct position
(4)Then performed all and operations to the query and finally all OR opearation which is our final result of the documents retrieved
(5)I have optimized the no of comparisions in AND and OR operator which is O(m+n)of list of size m and n
(6)Output :-no of docs retrieved , doc id of retrieved docs and total no of comparisions

QUESTION 2

PREPROCESSING

same as all steps mentioned in question1
Dataset used is only comp.graphics and rec.motorcycle

ASSUMPTIONS

(1) Not removed the stopwords .user can give input of query including stopwords
(2)Not performed stemming or lemmatization on tokens and datset
(3)Input query is phrase query with max size =5

Methodology

(1)Build an inverted Index using dictionary which stores all the tokens and their document id and also the occurence position of tokens in that document
(2) Retrived doc id and position of tokens in that doc id and then checked all the query words in that docid and pos+k
(3)if all input phrase query is matched then print that docid