Web Mining Lab Assignment 3

Name: Om Ashish Mishra

Registration Number: 16BCE0789

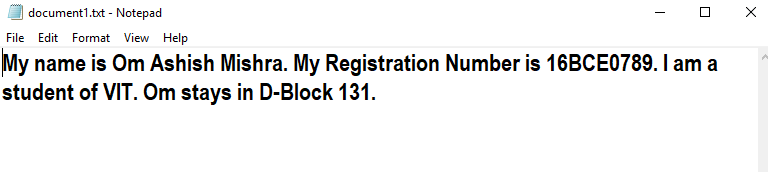
Slot: F2

# The Question:

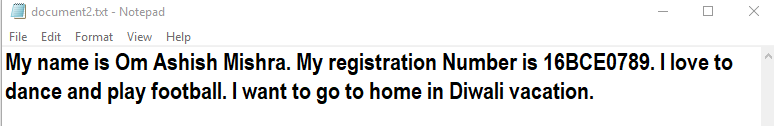
Write a program that collects all the words from a set of documents. Build an index from the words. Know what indexing is and Represent a document using the inverted index using python. Also implement a search for (multiple) terms from that index.

# The Answer:

The Document 1:



The Document 2:



## The Code(Using Regular Expression):

from os import system, name

import re

def process\_files(filenames):

file\_to\_terms = {}

for file in filenames:

pattern = re.compile('[\W\_]+')

file\_to\_terms[file] = open(file, 'r').read().lower();

file\_to\_terms[file] = pattern.sub(' ',file\_to\_terms[file])

re.sub(r'[\W\_]+','', file\_to\_terms[file])

file\_to\_terms[file] = file\_to\_terms[file].split()

return file\_to\_terms

def index\_one\_file(termlist):

fileIndex = {}

for index, word in enumerate(termlist):

if word in fileIndex.keys():

fileIndex[word].append(index)

else:

fileIndex[word] = [index]

return fileIndex

def make\_indices(termlists):

total = {}

for filename in termlists.keys():

total[filename] = index\_one\_file(termlists[filename])

return total

def fullIndex(regdex):

total\_index = {}

for filename in regdex.keys():

for word in regdex[filename].keys():

if word in total\_index.keys():

if filename in total\_index[word].keys():

total\_index[word][filename].extend(regdex[filename][word][:])

else:

total\_index[word][filename] = regdex[filename][word]

else:

total\_index[word] = {filename: regdex[filename][word]}

return total\_index

def one\_word\_query(word, invertedIndex):

pattern = re.compile('[\W\_]+')

word = pattern.sub(' ',word)

if word in invertedIndex.keys():

return [filename for filename in invertedIndex[word].values()]

else:

return []

def free\_text\_query(string,index):

pattern = re.compile('[\W\_]+')

string = pattern.sub(' ',string)

result = []

for word in string.split():

result += one\_word\_query(word,index)

return list(set(result))

def phrase\_query(string, invertedIndex):

pattern = re.compile('[\W\_]+')

string = pattern.sub(' ',string)

listOfLists, result = [],[]

for word in string.split():

listOfLists.append(free\_text\_query(word,invertedIndex))

setted = set(listOfLists[0]).intersection(\*listOfLists)

for filename in setted:

temp = []

for word in string.split():

temp.append(invertedIndex[word][filename][:])

for i in range(len(temp)):

for ind in range(len(temp[i])):

temp[i][ind] -= i

if set(temp[0]).intersection(\*temp):

result.append(filename)

print('\n temp : \n')

print(temp)

return result

filenames=['document1.txt','document2.txt']

termslist=process\_files(filenames)

print('\nterm list \n')

print(termslist)

print('\n\n')

print('\n\n')

totaldict=make\_indices(termslist)

print('total dictionary \n')

print(totaldict)

print('\n\n')

print('\n\n')

index=fullIndex(totaldict)

print('full index \n')

print(index)

print('\n\n')

#one\_word\_query('exceptions', index)

#query\_word=free\_text\_query('exceptions',index)

#print(query\_word)

system('cls')

print('\n\n')

print('\n\n')

#r=phrase\_query('python has exceptions handling',index)

#print (r)

## The Output:

runfile('C:/Users/OM/(OM)/5Fifth Semester/relab3.py', wdir='C:/Users/OM/(OM)/5Fifth Semester')

term list

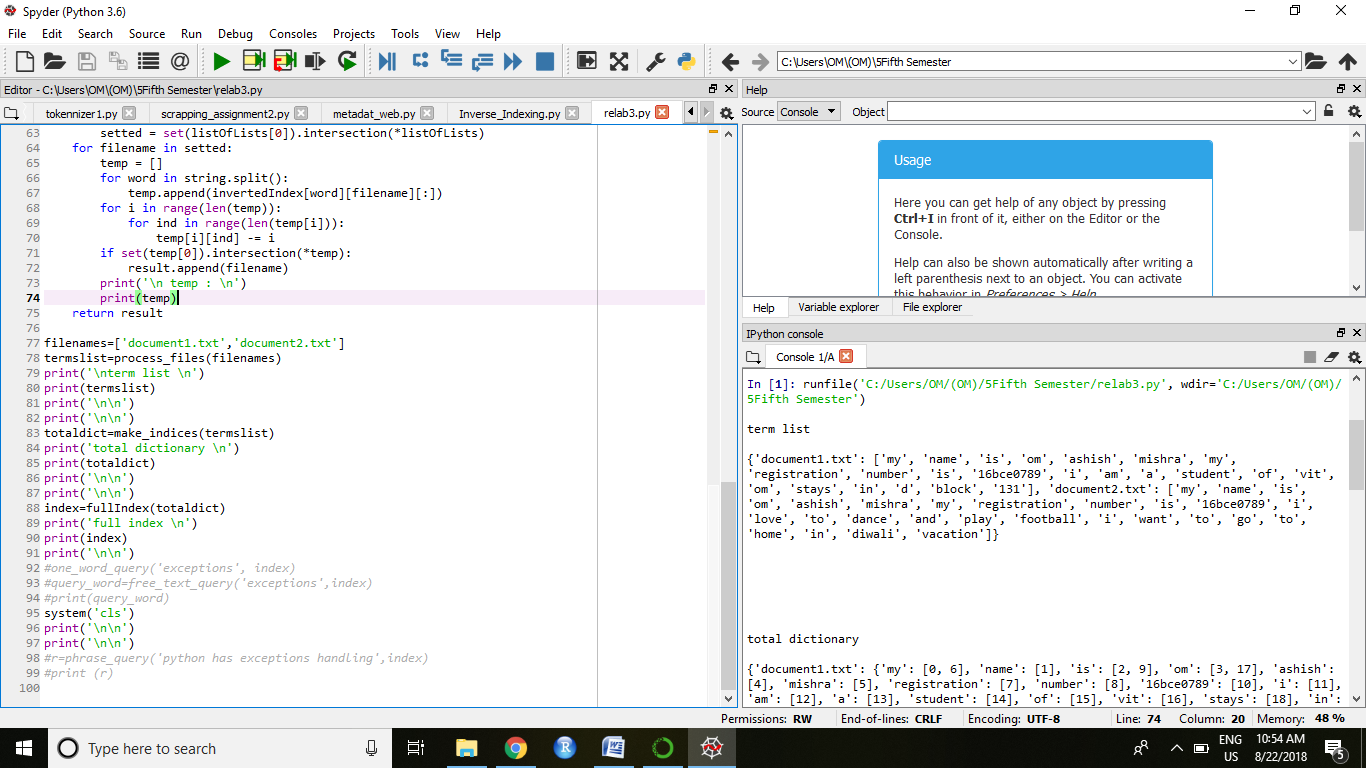
{'document1.txt': ['my', 'name', 'is', 'om', 'ashish', 'mishra', 'my', 'registration', 'number', 'is', '16bce0789', 'i', 'am', 'a', 'student', 'of', 'vit', 'om', 'stays', 'in', 'd', 'block', '131'], 'document2.txt': ['my', 'name', 'is', 'om', 'ashish', 'mishra', 'my', 'registration', 'number', 'is', '16bce0789', 'i', 'love', 'to', 'dance', 'and', 'play', 'football', 'i', 'want', 'to', 'go', 'to', 'home', 'in', 'diwali', 'vacation']}

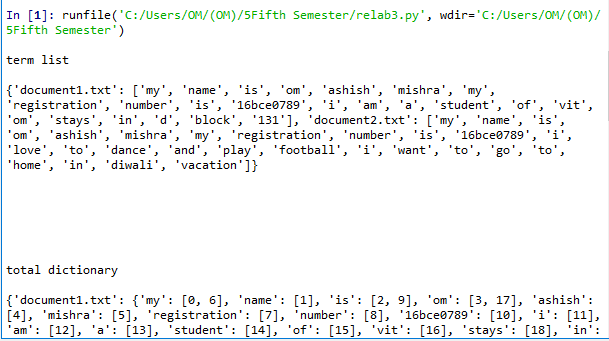
total dictionary

{'document1.txt': {'my': [0, 6], 'name': [1], 'is': [2, 9], 'om': [3, 17], 'ashish': [4], 'mishra': [5], 'registration': [7], 'number': [8], '16bce0789': [10], 'i': [11], 'am': [12], 'a': [13], 'student': [14], 'of': [15], 'vit': [16], 'stays': [18], 'in': [19], 'd': [20], 'block': [21], '131': [22]}, 'document2.txt': {'my': [0, 6], 'name': [1], 'is': [2, 9], 'om': [3], 'ashish': [4], 'mishra': [5], 'registration': [7], 'number': [8], '16bce0789': [10], 'i': [11, 18], 'love': [12], 'to': [13, 20, 22], 'dance': [14], 'and': [15], 'play': [16], 'football': [17], 'want': [19], 'go': [21], 'home': [23], 'in': [24], 'diwali': [25], 'vacation': [26]}}

full index

{'my': {'document1.txt': [0, 6], 'document2.txt': [0, 6]}, 'name': {'document1.txt': [1], 'document2.txt': [1]}, 'is': {'document1.txt': [2, 9], 'document2.txt': [2, 9]}, 'om': {'document1.txt': [3, 17], 'document2.txt': [3]}, 'ashish': {'document1.txt': [4], 'document2.txt': [4]}, 'mishra': {'document1.txt': [5], 'document2.txt': [5]}, 'registration': {'document1.txt': [7], 'document2.txt': [7]}, 'number': {'document1.txt': [8], 'document2.txt': [8]}, '16bce0789': {'document1.txt': [10], 'document2.txt': [10]}, 'i': {'document1.txt': [11], 'document2.txt': [11, 18]}, 'am': {'document1.txt': [12]}, 'a': {'document1.txt': [13]}, 'student': {'document1.txt': [14]}, 'of': {'document1.txt': [15]}, 'vit': {'document1.txt': [16]}, 'stays': {'document1.txt': [18]}, 'in': {'document1.txt': [19], 'document2.txt': [24]}, 'd': {'document1.txt': [20]}, 'block': {'document1.txt': [21]}, '131': {'document1.txt': [22]}, 'love': {'document2.txt': [12]}, 'to': {'document2.txt': [13, 20, 22]}, 'dance': {'document2.txt': [14]}, 'and': {'document2.txt': [15]}, 'play': {'document2.txt': [16]}, 'football': {'document2.txt': [17]}, 'want': {'document2.txt': [19]}, 'go': {'document2.txt': [21]}, 'home': {'document2.txt': [23]}, 'diwali': {'document2.txt': [25]}, 'vacation': {'document2.txt': [26]}}





## The Code(The Glob Package used):

from pprint import pprint as pp

from glob import glob

try: reduce

except: from functools import reduce

try: raw\_input

except: raw\_input = input

def parsetexts(fileglob='document\*.txt'):

texts, words = {}, set()

for txtfile in glob(fileglob):

with open(txtfile, 'r') as f:

txt = f.read().split()

words |= set(txt)

texts[txtfile.split('\\')[-1]] = txt

return texts, words

def termsearch(terms): # Searches simple inverted index

return reduce(set.intersection,(invindex[term] for term in terms),set(texts.keys()))

texts, words = parsetexts()

print('\nTexts')

pp(texts)

print('\nWords')

pp(sorted(words))

invindex = {word:set(txt for txt, wrds in texts.items() if word in wrds)for word in words}

print('\nInverted Index')

pp({k:sorted(v) for k,v in invindex.items()})

terms = ["what", "is", "it"]

print('\nTerm Search for: ' + repr(terms))

pp(sorted(termsearch(terms)))

## The Output:

runfile('C:/Users/OM/(OM)/5Fifth Semester/Inverse\_Indexing.py', wdir='C:/Users/OM/(OM)/5Fifth Semester')

Texts

{'document1.txt': ['My',

'name',

'is',

'Om',

'Ashish',

'Mishra.',

'My',

'Registration',

'Number',

'is',

'16BCE0789.',

'I',

'am',

'a',

'student',

'of',

'VIT.',

'Om',

'stays',

'in',

'D-Block',

'131.'],

'document2.txt': ['My',

'name',

'is',

'Om',

'Ashish',

'Mishra.',

'My',

'registration',

'Number',

'is',

'16BCE0789.',

'I',

'love',

'to',

'dance',

'and',

'play',

'football.',

'I',

'want',

'to',

'go',

'to',

'home',

'in',

'Diwali',

'vacation.']}

Words

['131.',

'16BCE0789.',

'Ashish',

'D-Block',

'Diwali',

'I',

'Mishra.',

'My',

'Number',

'Om',

'Registration',

'VIT.',

'a',

'am',

'and',

'dance',

'football.',

'go',

'home',

'in',

'is',

'love',

'name',

'of',

'play',

'registration',

'stays',

'student',

'to',

'vacation.',

'want']

Inverted Index

{'131.': ['document1.txt'],

'16BCE0789.': ['document1.txt', 'document2.txt'],

'Ashish': ['document1.txt', 'document2.txt'],

'D-Block': ['document1.txt'],

'Diwali': ['document2.txt'],

'I': ['document1.txt', 'document2.txt'],

'Mishra.': ['document1.txt', 'document2.txt'],

'My': ['document1.txt', 'document2.txt'],

'Number': ['document1.txt', 'document2.txt'],

'Om': ['document1.txt', 'document2.txt'],

'Registration': ['document1.txt'],

'VIT.': ['document1.txt'],

'a': ['document1.txt'],

'am': ['document1.txt'],

'and': ['document2.txt'],

'dance': ['document2.txt'],

'football.': ['document2.txt'],

'go': ['document2.txt'],

'home': ['document2.txt'],

'in': ['document1.txt', 'document2.txt'],

'is': ['document1.txt', 'document2.txt'],

'love': ['document2.txt'],

'name': ['document1.txt', 'document2.txt'],

'of': ['document1.txt'],

'play': ['document2.txt'],

'registration': ['document2.txt'],

'stays': ['document1.txt'],

'student': ['document1.txt'],

'to': ['document2.txt'],

'vacation.': ['document2.txt'],

'want': ['document2.txt']}

Term Search for: ['what', 'is', 'it']

Traceback (most recent call last):

File "<ipython-input-2-b00121154629>", line 1, in <module>

runfile('C:/Users/OM/(OM)/5Fifth Semester/Inverse\_Indexing.py', wdir='C:/Users/OM/(OM)/5Fifth Semester')

File "C:\Users\OM\Anaconda3\lib\site-packages\spyder\_kernels\customize\spydercustomize.py", line 678, in runfile

execfile(filename, namespace)

File "C:\Users\OM\Anaconda3\lib\site-packages\spyder\_kernels\customize\spydercustomize.py", line 106, in execfile

exec(compile(f.read(), filename, 'exec'), namespace)

File "C:/Users/OM/(OM)/5Fifth Semester/Inverse\_Indexing.py", line 30, in <module>

pp(sorted(termsearch(terms)))

File "C:/Users/OM/(OM)/5Fifth Semester/Inverse\_Indexing.py", line 18, in termsearch

return reduce(set.intersection,(invindex[term] for term in terms),set(texts.keys()))

File "C:/Users/OM/(OM)/5Fifth Semester/Inverse\_Indexing.py", line 18, in <genexpr>

return reduce(set.intersection,(invindex[term] for term in terms),set(texts.keys()))

KeyError: 'what'

