

CmpE 493 – Project 1

i) Preprocessing part:

First, i construct an algortihm for tokenizing step. Only takes NewId field and title,body parts of the documents. Case-folding is easist step. For stemming, i used porter stemmer from the website and integreted into my code. Between these two, stopwords removed.

Answers of the questions are below. Yet, e and f part, i counted the words only 1 time per 1 new (i thought that it will be approximately same so i didn't change because it consumes time a lot). The algorithm works that way to answer queires. I forgot to change when i started the sorting algortihm and sorting takes about 45 min per each part. I commented out that part because of no effection query part. Only opened it to be able to sort, otherwise it slows down the program. To sum up, i counted the unique tokens (terms) occurance instead of tokens.

(a) How many tokens does the corpus contain before stopword removal and stemming?

- 2808571

(b) How many tokens does the corpus contain after stopword removal and stemming?

- 2191349

(c) How many terms (unique tokens) are there before stopword removal, stemming, and case-folding?

-73129

(d) How many terms (unique tokens) are there after stop word removal, stemming, and case-folding?

- 68219

(e) List the top 20 most frequent terms before stopword removal, stemming, and casefolding?

- ['reuter', 'of', 'the', 'said', 'to', 'and', 'in', 'a', 'for', 'it', 'mln', 'on', 'dlrs', 'its', 'is', 'from', 'by', 'at', 'with', 'will']

(f) List the top 20 most frequent terms after stopword removal, stemming, and case-folding?

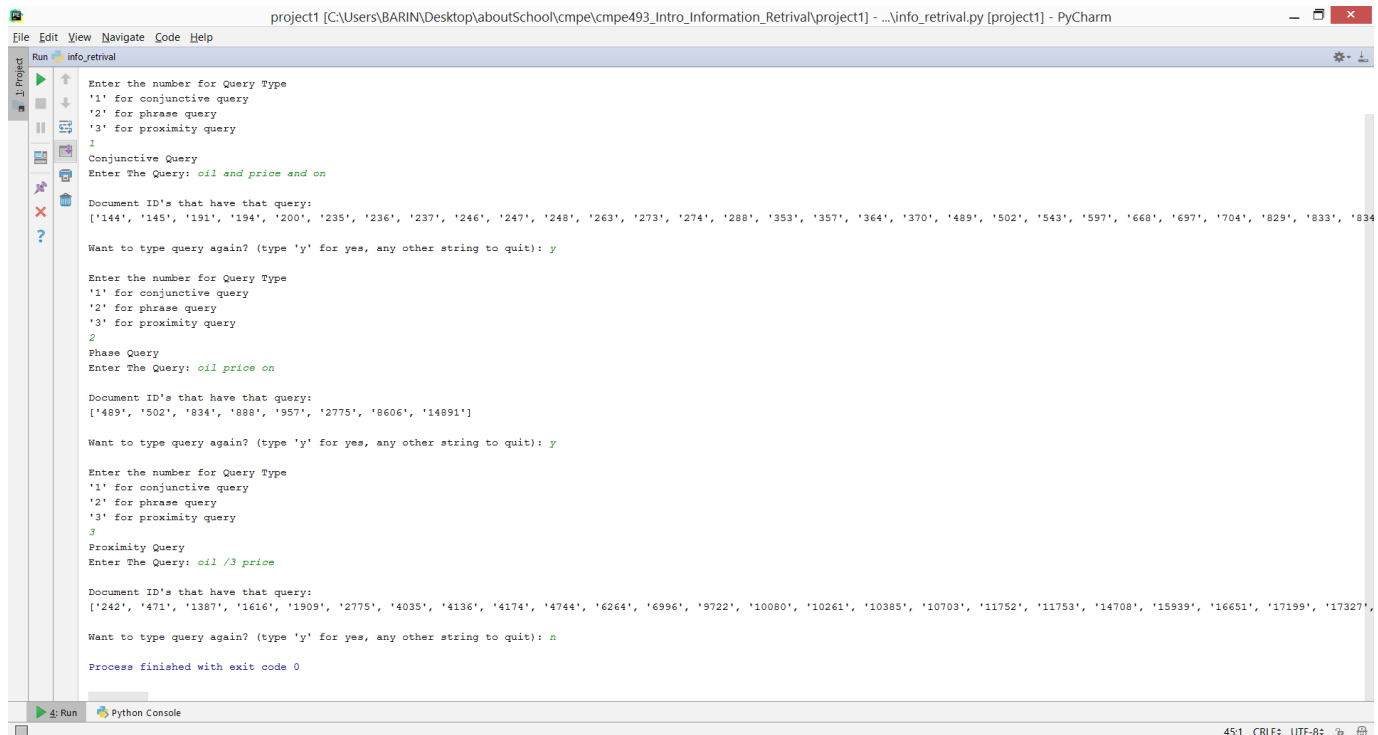
- ['reuter', 'said', 'to', 'on', 'mln', 'dlr', 'from', 'by', 'at', 'year', 'pct', 'that', '1', 'ha', 'compani', 'inc', '2', 'corp', 'not', '000']

ii) Dictionary and inverted index:

Dictionary consists hashmaps for every new id with starting as first element (HashMap[0] = newId). Others keys are their ids from 1 to number of the words that news have sorted by positions on that document. Values obtained from keys are tokens (strings).

Inverted Index is created from dictionary. It has only 1 hashmap that has the terms for strings as keys and news id for value in which document these terms occurs. Example: { 'barın' : [1, 10, 20, 30...] , ' ' : [] ... }

iii) Screenshot of running system



```
project1 [C:\Users\BARIN\Desktop\aboutSchool\cmpe\cmpe493_Intro_Information_Retrieval\project1] - ...info_retrival.py [project1] - PyCharm
File Edit View Navigate Code Help
Run info_retrival
Enter the number for Query Type
'1' for conjunctive query
'2' for phrase query
'3' for proximity query
1
Conjunctive Query
Enter The Query: oil and price and on
Document ID's that have that query:
['144', '145', '191', '194', '200', '235', '236', '237', '246', '247', '248', '263', '273', '274', '288', '353', '357', '364', '370', '489', '502', '543', '597', '668', '697', '704', '829', '833', '834']
Want to type query again? (type 'y' for yes, any other string to quit): y
Enter the number for Query Type
'1' for conjunctive query
'2' for phrase query
'3' for proximity query
2
Phrase Query
Enter The Query: oil price on
Document ID's that have that query:
['489', '502', '834', '888', '957', '2775', '8606', '14891']
Want to type query again? (type 'y' for yes, any other string to quit): y
Enter the number for Query Type
'1' for conjunctive query
'2' for phrase query
'3' for proximity query
3
Proximity Query
Enter The Query: oil /3 price
Document ID's that have that query:
['242', '471', '1387', '1616', '1909', '2775', '4035', '4136', '4174', '4744', '6264', '6996', '9722', '10080', '10261', '10385', '10703', '11752', '11753', '14708', '15939', '16651', '17199', '17327']
Want to type query again? (type 'y' for yes, any other string to quit): n
Process finished with exit code 0
Run Python Console
45:1 CRLF UTF-8
```