Barın Özmen
2012400045

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 uniqe 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']

Barın Özmen
2012400045

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