CMPE 493
Information Retrieval
Assignment 1

A Simple Document Retrieval System for Boolean Queries

Beyza Gul Gurbuz
2013400081

March 8, 2018

Data Preprocessing

The Preprocessing consists of four parts, namely, tokenization, case folding, stemming, and stopword removal.

Tokenization

To tokenize the dataset, first of all, all the documents has been taken from the dataset by looking their xml structures. (starts with <REUTERS and ends with </REUTERS>).Id, title, and body of each document has been taken. The title and body of each document have been splitted according to space character. After that, the punctuation marks on front and back of all the tokens have been stripped.

Case Folding

All tokens have been changed to lowercase versions of them.

Stemming

After tokenization and case folding, all the tokens have sent to Porter Stemmer.

Stopword Removal

Stop word removal has been handled during positional inverted index creation since although they are not wanted in dictionary, we still need their positions in positional inverted index.

- Number of tokens before stopword removal and stemming: 2702225
- Number of tokens after stopword removal and stemming: 2056322
- Number of terms before stopword removal, stemming, and case-folding:89296
- Number of terms after stopword removal, stemming, and case-folding:62134
- 20 most frequent terms before stopword removal, stemming, and case-folding:
  'the', 'of', 'to', 'and', 'said', 'in', 'a', 'mln', 'for', 'The', 'dlrs', 'it', 'on', 'pct', 'is', 'that', 'from', 'its', 'will', 'vs'

- 20 most frequent terms after stopword removal, stemming, and case-folding:
  'to', 'said', 'mln', 'dlr', 'reuter', 'pct', 'from', 'vs', 'at', 'year', 'wa', 'bank', 'billion', 'ha', 'compani', 'share', 'u.', 'ct', 'would', 'market'

Used Data Structures

For dictionary, a python dictionary whose keys are terms and values are the id's of terms has been used.
For positional index, a dictionary whose keys are the id's of terms has been used. The values of the positional index dictionary are also dictionaries whose key is all documents that contains the term and value is an array of the positions of the term in each document.

Appendix

Figure 1: Conjunction Query

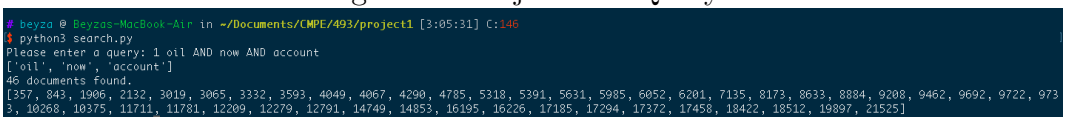

Figure 2: Phrase Query

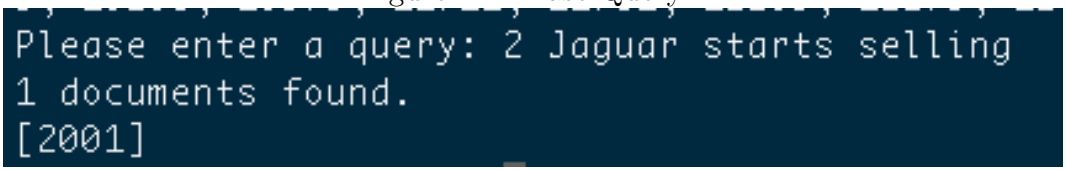

Figure 3: Proximity Query