

Cmpe 493 Introduction to Information Retrieval, Fall 2021
Assignment 2 - A Simple Search System for Phrase and Free Text Queries
Zuhal Didem Aytaç - 2018400045

1. Data Preprocessing

I created a file preprocess.py. This code handles the initial processing of data. It reads the reutx-xxx.sgm files and the stopwords.txt file. It extracts the necessary content from the sgm files. Then with the helper functions, it performs case-fold, punctuation removal and stop-word removal operations on the extracted texts. It keeps the normalized, tokenized text using a dictionary, with new_id as the key.

This preprocess module is not run by itself, it is called by the indexing module.

2. The Inverted Index

The indexize module calls the preprocess module described in part 1. The preprocess module returns a dictionary (*normalized*) with new_id as key and list of tokens as value. The indexize module traverses that dictionary (*normalized*) and builds two new dictionaries. One is *inverted_intex* with token as keys and list of new_ids as value. The other is *document_index* with new_id as keys and {token:frequency} dictionary as value.

After that, the *build_frequency_index* function is called with *inverted_intex* dictionary. This function builds the dictionary *index_with_frequencies*. It stores for each token, the document frequency and the term frequencies and positions for each news article containing the token. The dictionary is as follows:

```
{ token:
  {
    'df': document frequency,
    'documents': { new_id: { 'tf': term frequency, 'positions': list of positions } }
  }
}
```

Finally, the *index_with_frequencies* and *document_index* dictionaries are dumped to json files (*index.json* and *documents.json*).

3. The Indexing Module

```
didemaytac@Zuhal-MacBook-Pro src % python3 indexize.py
```

4. Query Processing

Type python3 query_processor.py, then type the query and press enter. The output will be shown at the command line. If no results are found, nothing is printed.

Free text query old crop cocoa (only first and last 5 results are screenshotted):

```
didemaytac@Zuhal-MacBook-Pro src % python3 query_processor.py  
old crop cocoa  
10491: 0.2760332448513201  
10471: 0.26636939976928303  
19358: 0.239975282285636  
18221: 0.23793650255963059  
17733: 0.23347756270392622
```

...

```
7405: 0.026159547113957238  
1579: 0.02595574756033258  
12011: 0.025263604871086247  
5214: 0.020380987112288165  
6657: 0.020054895075638304
```

Other free text query examples (lira) (bogazici) :

```
didemaytac@Zuhal-MacBook-Pro src % python3 query_processor.py  
lira  
10636: 0.2961901203472848  
10340: 0.24422363514913387  
3518: 0.15610216003944227  
11823: 0.13302104186803607  
6392: 0.12856239494073188  
17293: 0.12708228448405548  
17300: 0.12708228448405548  
15534: 0.1175496882394384  
7105: 0.1141008857417545  
14664: 0.08528480018917496  
4809: 0.08374865477083687  
bogazici  
█
```

Phrase query examples (“old crop cocoa”, “bogazici university”, “turkish lira”, “lira”):

```
didemaytac@Zuhal-MacBook-Pro src % python3 query_processor.py  
"old crop cocoa"  
1  
"bogazici university"  
"turkish lira"  
7105  
"lira"  
3518  
4809  
6392  
7105  
10340  
10636  
11823  
14664  
15534  
17293  
17300  
█
```