CMPE 493 INFORMATION RETRIEVAL ASSIGNMENT 1 : BOOLEAN SEARCH ENGINE

In this assignment we built a boolean search engine. Inverted index was used as the search dictionary. AND, OR and NOT operators for singular words are usable in the system.

Index building module is working with:

python3 index build.py ./reuters21578/ ./stopwords.txt

As the index is built, Query processing module is working with:

python3 query process.py

Assignment consists of 3 main steps:

- 1- Data Preprocessing
- 2- Inverted Index Building
- 3- Query Processing and Boolean Search

1- Data Preprocessing

Reuters data was read via a HTML file processing library, beautifulSoup. As the documents are read, they are trimmed from punctionations according to the string.punctuation list and cleaned from stop words specified by the user. Latin-1 encoding was used for the documents. Multiprocessing which creates a process for each document was used to speed up the data reading at data preprocessing step.

2- Inverted Index Building

In this step the index of files which is linked as doc to words in doc is transformed into word to docs containing the word.

For the inverted index, I used built-in dictionary in which keys represent words while values are basic list structures for postings list. I prefered dictionary (hashmap structure in python) for inverted index because it uses a hash function to link key and values and it provides an O(1) time complexity to store and retrieve a particular object. [1] I prefered a basic array structure for postings list as it has O(1) complexity for adding new elements. It has O(n) complexity for searching items but as the postings lists of words are not large, I thought it won't create a problem during query search. [2]

3- Query Processing

At the last step of the assignment, a query processing and a boolean search engine connected to the inverted index built in the first step is implemented. Conjunction, disjunction and negation operations for singular words are open to use.

```
Last login: Thu Nov 18 21:48:08 on ttys000
| maral@192 ~ % cd /Users/maral/Desktop/Assignment1 | maral@192 Assignment1 % python3 index_build.py ./reuters21578/ ./stopwords.txt - Index is created in 4.194551944732666 seconds - maral@192 Assignment1 % |
```

Figure 1 (Inverted index building operation takes 4.19 seconds)

Creating an inverted index takes 4.19 seconds.

```
Last login: Thu Nov 18 21:48:08 on ttys000
|maral@192 ~ % cd /Users/maral/Desktop/Assignment1
|maral@192 Assignment1 % python3 index_build.py ./reuters21578/ ./stopwords.txt
 price AND oil is found at:
[127, 144, 191, 194, 213, 236, 246, 263, 357, 471, 489, 502, 543, 597, 829, 834, 843, 873, 885, 952, 1026, 1349, 1370, 1387, 1711, 1848, 18
75, 1909, 1909, 2045, 2061, 2068, 2074, 2121, 2132, 2228, 2251, 2383, 266, 2775, 2828, 2833, 2975, 2998, 3024, 3065, 3174, 3181, 3189, 324
9, 3303, 3342, 3389, 3430, 3452, 3455, 3490, 3535, 3563, 3571, 3593, 3798, 3869, 3985, 4005, 4017, 4035, 4061, 4174, 4214, 4232, 4453, 4474
, 4481, 4546, 4564, 4576, 4584, 4634, 4662, 4679, 4713, 4744, 4835, 4878, 5037, 5061, 5145, 5167, 5179, 5184, 5244, 5255, 5268, 5270, 5273,
5318, 5323, 5389, 5569, 5631, 5761, 5769, 5787, 5851, 5936, 6023, 6086, 6121, 6177, 6201, 6208, 6371, 6413, 6656, 6876, 6954, 6994, 6994,
7174, 7200, 7408, 7639, 7643, 7731, 7937, 8815, 8041, 8095, 8131, 8134, 8173, 8209, 8210, 8478, 8606, 8615, 8630, 8820, 8960, 8964, 9031, 9
077, 9149, 9156, 9213, 9392, 9436, 9462, 9485, 9639, 9650, 9691, 9706, 9732, 9733, 9761, 9763, 9799, 9853, 9933, 9947, 10015, 10078, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081, 10081,
      price AND oil is found at:
```

Figure 2 (A Conjunction Operation for 'price AND oil)

```
Please enter a query(''E'' for exit):petroleum OR oil OR gas petroleum OR oil OR gas petroleum OR oil OR gas is found at: [68, 84, 91, 140, 145, 176, 277, 394, 379, 372, 507, 542, 544, 570, 622, 708, 739, 799, 837, 862, 918, 927, 963, 1024, 1066, 1084, 1297, 13 01, 1316, 1335, 1343, 1379, 1387, 1456, 1466, 1561, 1520, 1558, 1619, 1650, 1658, 1661, 1708, 1709, 1711, 1751, 1756, 1824, 1856, 1891, 190 6, 1948, 2007, 2045, 2061, 2303, 2394, 2415, 2477, 2479, 2480, 2483, 2511, 2515, 2522, 2528, 2685, 2373, 72738, 2745, 2758, 2789, 2818, 2833, 2835, 2897, 2923, 2927, 2998, 3015, 3024, 3080, 3145, 3169, 3174, 3210, 3249, 3252, 2528, 2685, 2373, 72738, 2745, 2758, 2789, 2789, 2781, 3752, 3758, 3763, 3779, 3787, 3789, 3849, 3843, 3846, 3814, 3169, 3174, 3210, 3249, 3252, 3264, 3310, 3344, 3490, 3505, 3592, 3714, 3736, 3747, 3752, 3758, 3768, 3779, 3787, 3789, 3849, 3843, 3846, 3864, 3872, 3925, 3980, 491, 4017, 4027, 4049, 4162, 4209, 4222, 4290, 4317, 4340, 4353, 4467, 4475, 4507, 4509, 4509, 4509, 4509, 4509, 4609, 4601, 4604, 4661, 4730, 4734, 4755, 4834, 4835, 4908, 4930, 4936, 4951, 4902, 4963, 4961, 5037, 5944, 5061, 5150, 5187, 5210, 5236, 5295, 5342, 5425, 5440, 5472, 5538, 5544, 5552, 5675, 5739, 5755, 5776, 5793, 5796, 581, 4888, 6936, 5949, 5945, 6086, 6656, 6660, 6685, 6688, 6692, 6712, 6722, 6740, 6742, 6746, 6760, 6836, 6876, 6888, 6995, 6954, 7807, 7997, 7156, 7157, 7209, 7287, 7291, 7397, 7397, 7397, 7480, 7420, 7220, 7548, 7611, 7639, 7663, 7617, 7702, 7749, 7749, 7749, 7799, 7831, 7897, 7897, 7897, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 7891, 789
    Please enter a query(''E'' for exit):petroleum OR oil OR gas
```

Figure 3 (A disjunction operation for 'petroleum OR oil OR gas')

Maral Dicle MARAL 18.11.2021

price And 0:11 NOT vegetable

price

Figure 4 (A Conjunction and Negation operation for 'price AND oil NOT vegetable')

Please enter a query(''E'' for exit):petroleum OR oil NOT price petroleum OR oil NOT price is found at:
[2, 140, 166, 211, 237, 273, 340, 370, 391, 918, 927, 939, 988, 1004, 1808, 1336, 1343, 1456, 1463, 1558, 1616, 1660, 1825, 1856, 1874, 1878, 1948, 2159, 2187, 2210, 2221, 223
1, 2262, 2394, 2435, 2495, 2820, 2834, 2838, 2923, 2973, 3808, 3918, 3948, 3218, 3318, 3328, 334, 3908, 3908, 4839, 4129, 4201, 4315, 4367, 4454, 4563, 4510, 5118, 5119, 5119, 5116, 5136, 3203, 5203, 5208, 5203, 5386, 5578, 5560, 5675, 5712, 5789, 5796, 5985, 6080, 4610, 4577, 4608, 4604, 4604, 4604, 4607, 460

Figure 5 (Disjunction and Negation operation for 'petroleum OR oil NOT price')

Search operations generally take around 0.12 seconds for each search.

REFERENCES

- [1]"Built-in Types Python 3.10.0 documentation", *Docs.python.org*, 2021. [Online]. Available: https://docs.python.org/3/library/stdtypes.html#typesmapping. [Accessed: 18- Nov- 2021].
- [2]"The complexity of simple algorithms and data structures in JS", *freeCodeCamp.org*, 2021. [Online]. Available: https://www.freecodecamp.org/news/the-complexity-of-simple-algorithms-and-data-structures-in-javascript-11e25b29de1e/. [Accessed: 18-Nov- 2021].