Demo for Project 3 of COMP 479 by Silas Kalinowski ID: 40256077

The program can be started by running the main.py file in a Python environment. For the program to work all modules and all input files need to be in the local directory. Also, the program uses NLTK, which needs to be downloaded. The output files are saved in the local directory and are named in the following manner:

index.txt for the index containing the idf, terms and postings lists with tf

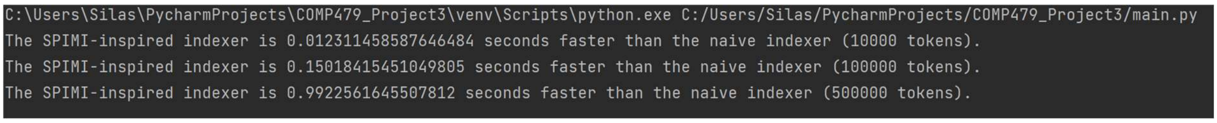
index_doc.txt for the index that stores the length of every document

subcorpus_10000.txt contains all the list of tokens for the subcorpus.

Running the main file starts the naïve indexer and the SPIMI indexer and prints the time taken for different subcorpi, then the whole corpus is indexed and lastly the queries are started. The user can select the type of query he likes to use and then terms in the console which he wants to search for or type t to run all tests and sample queries. The query starts again after every output and can be stopped by typing stop.

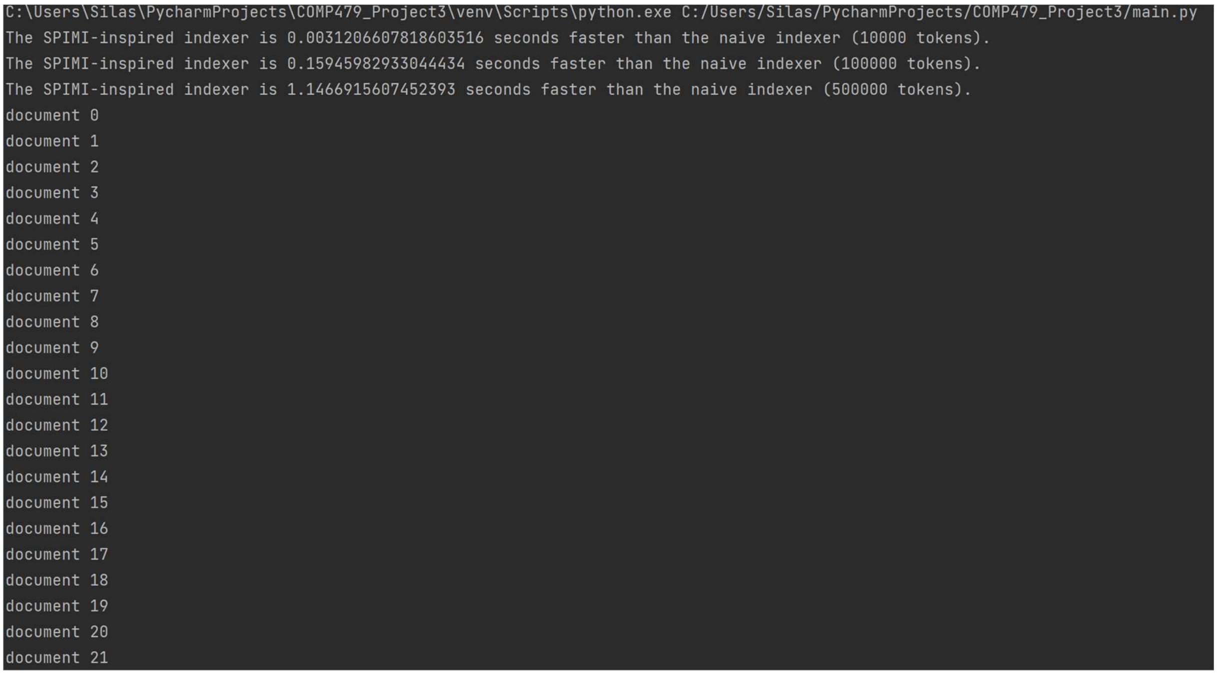
Subproject 1

When running the main.py file the comparison between SPIMI inspired indexer and naïve indexer is automatically started and the results are printed.



```
C:\Users\Silas\PycharmProjects\COMP479_Project3\venv\Scripts\python.exe C:/Users/Silas/PycharmProjects/COMP479_Project3/main.py
The SPIMI-inspired indexer is 0.012311458587646484 seconds faster than the naive indexer (10000 tokens).
The SPIMI-inspired indexer is 0.15018415451049805 seconds faster than the naive indexer (100000 tokens).
The SPIMI-inspired indexer is 0.9922561645507812 seconds faster than the naive indexer (500000 tokens).
```

After that the whole index with all the parameters for BM25 is created.



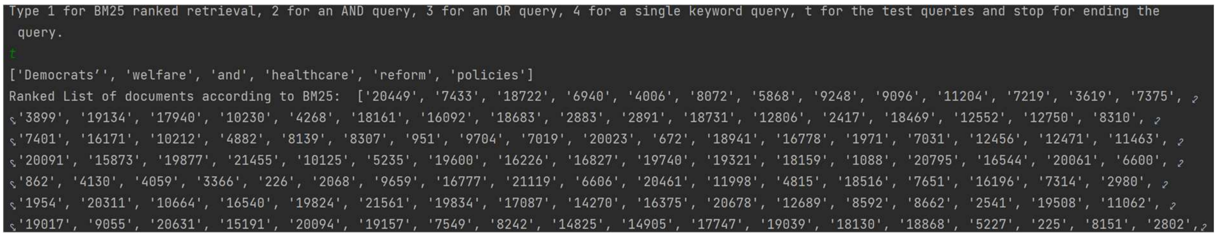
```
C:\Users\Silas\PycharmProjects\COMP479_Project3\venv\Scripts\python.exe C:/Users/Silas/PycharmProjects/COMP479_Project3/main.py
The SPIMI-inspired indexer is 0.0031206607818603516 seconds faster than the naive indexer (10000 tokens).
The SPIMI-inspired indexer is 0.15945982933044434 seconds faster than the naive indexer (100000 tokens).
The SPIMI-inspired indexer is 1.1466915607452393 seconds faster than the naive indexer (500000 tokens).
document 0
document 1
document 2
document 3
document 4
document 5
document 6
document 7
document 8
document 9
document 10
document 11
document 12
document 13
document 14
document 15
document 16
document 17
document 18
document 19
document 20
document 21
```

Then the query input for subproject 2 is started as seen below.

Subproject 2

The user just must follow the print in the console, the query starts automatically after a search again and stops when typing stop.

The user can start the sample and test queries by typing t.



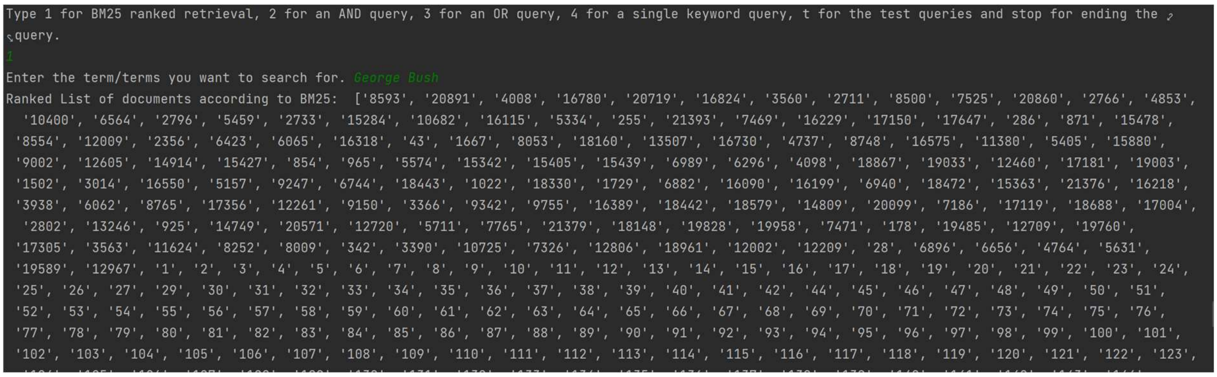
```
Type 1 for BM25 ranked retrieval, 2 for an AND query, 3 for an OR query, 4 for a single keyword query, t for the test queries and stop for ending the
 query.
t
['Democrats'', 'welfare', 'and', 'healthcare', 'reform', 'policies']
Ranked List of documents according to BM25:  ['20449', '7433', '18722', '6940', '4006', '8072', '5868', '9248', '9096', '11204', '7219', '3619', '7375',
'3899', '19134', '17940', '10230', '4268', '18161', '16092', '18683', '2883', '2891', '18731', '12806', '2417', '18469', '12552', '12750', '8310',
'7401', '16171', '10212', '4882', '8139', '8307', '951', '9704', '7019', '20023', '672', '18941', '16778', '1971', '7031', '12456', '12471', '11463',
'20091', '15873', '19877', '21455', '10125', '5235', '19600', '16226', '16827', '19740', '19321', '18159', '1088', '20795', '16544', '20061', '6600',
'862', '4130', '4059', '3366', '226', '2068', '9659', '16777', '21119', '6606', '20461', '11998', '4815', '18516', '7651', '16196', '7314', '2980',
'1954', '20311', '10664', '16540', '19824', '21561', '19834', '17087', '14270', '16375', '20678', '12689', '8592', '8662', '2541', '19508', '11062',
'19017', '9055', '20631', '15191', '20094', '19157', '7549', '8242', '14825', '14905', '17747', '19039', '18130', '18868', '5227', '225', '8151', '2802',
```

The results are stored in txt files.

When the user types 1 the input is processed by using the BM25 formula and the full list of ranked documents is returned.

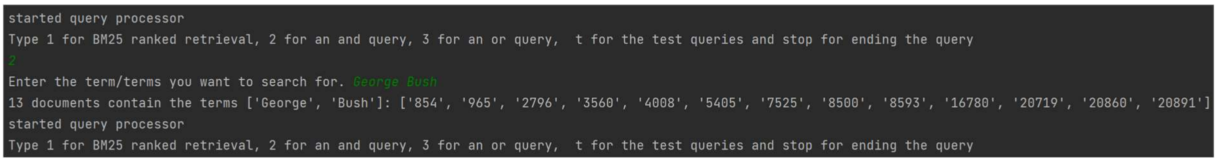
The user can search for terms in the different queries and can stop the query by typing stop.

Otherwise, the query is starts again after every output.



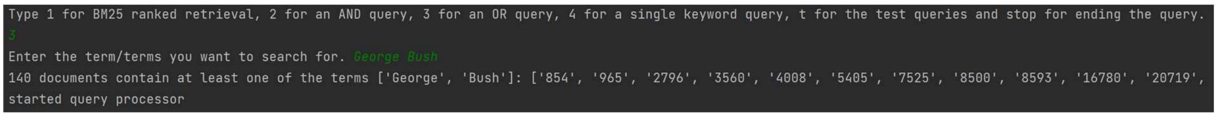
```
Type 1 for BM25 ranked retrieval, 2 for an AND query, 3 for an OR query, 4 for a single keyword query, t for the test queries and stop for ending the
query.
1
Enter the term/terms you want to search for. George Bush
Ranked List of documents according to BM25:  ['8593', '20891', '4008', '16780', '20719', '16824', '3560', '2711', '8500', '7525', '20860', '2766', '4853',
  '10400', '6564', '2796', '5459', '2733', '15284', '10682', '16115', '5334', '255', '21393', '7469', '16229', '17150', '17647', '286', '871', '15478',
 '8554', '12009', '2356', '6423', '6065', '16318', '43', '1667', '8053', '18160', '13507', '16730', '4737', '8748', '16575', '11380', '5405', '15880',
 '9002', '12605', '14914', '15427', '854', '965', '5574', '15342', '15405', '15439', '6989', '6296', '4098', '18867', '19033', '12460', '17181', '19003',
 '1502', '3014', '16550', '5157', '9247', '6744', '18443', '1022', '18330', '1729', '6882', '16090', '16199', '6940', '18472', '15363', '21376', '16218',
 '3938', '6062', '8765', '17356', '12261', '9150', '3366', '9342', '9755', '16389', '18442', '18579', '14809', '20099', '7186', '17119', '18688', '17004',
  '2802', '13246', '925', '14749', '20571', '12720', '5711', '7765', '21379', '18148', '19828', '19958', '7471', '178', '19485', '12709', '19760',
 '17305', '3563', '11624', '8252', '8009', '342', '3390', '10725', '7326', '12806', '18961', '12002', '12209', '28', '6896', '6656', '4764', '5631',
 '19589', '12967', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', '21', '22', '23', '24',
 '25', '26', '27', '29', '30', '31', '32', '33', '34', '35', '36', '37', '38', '39', '40', '41', '42', '44', '45', '46', '47', '48', '49', '50', '51',
 '52', '53', '54', '55', '56', '57', '58', '59', '60', '61', '62', '63', '64', '65', '66', '67', '68', '69', '70', '71', '72', '73', '74', '75', '76',
 '77', '78', '79', '80', '81', '82', '83', '84', '85', '86', '87', '88', '89', '90', '91', '92', '93', '94', '95', '96', '97', '98', '99', '100', '101',
 '102', '103', '104', '105', '106', '107', '108', '109', '110', '111', '112', '113', '114', '115', '116', '117', '118', '119', '120', '121', '122', '123',
```

When the user types 2 the input is processed as an AND query. Only documents that contain all the terms in the input are returned. The terms should be separated by a space.



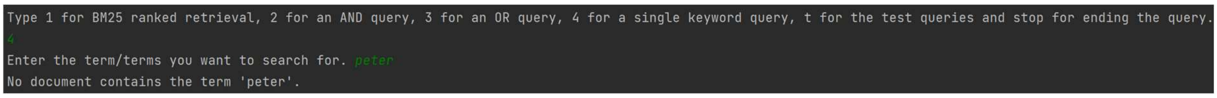
```
started query processor
Type 1 for BM25 ranked retrieval, 2 for an and query, 3 for an or query,  t for the test queries and stop for ending the query
2
Enter the term/terms you want to search for. George Bush
13 documents contain the terms ['George', 'Bush']: ['854', '965', '2796', '3560', '4008', '5405', '7525', '8500', '8593', '16780', '20719', '20860', '20891']
started query processor
Type 1 for BM25 ranked retrieval, 2 for an and query, 3 for an or query,  t for the test queries and stop for ending the query
```

When the user types 3 the input is processed as an OR query. Only documents that contain at least one of the terms in the input are returned. The documents are ranked by the how many terms of the input they contain. The terms should be separated by a space.



```
Type 1 for BM25 ranked retrieval, 2 for an AND query, 3 for an OR query, 4 for a single keyword query, t for the test queries and stop for ending the query.
3
Enter the term/terms you want to search for. George Bush
140 documents contain at least one of the terms ['George', 'Bush']: ['854', '965', '2796', '3560', '4008', '5405', '7525', '8500', '8593', '16780', '20719',
started query processor
```

When the user types 4 the input is processed as a single keyword query.



```
Type 1 for BM25 ranked retrieval, 2 for an AND query, 3 for an OR query, 4 for a single keyword query, t for the test queries and stop for ending the query.
4
Enter the term/terms you want to search for. peter
No document contains the term 'peter'.
```

I certify that this submission is my original work and meets the Faculty's Expectations of Originality.

S Malinowski

Signature ID: 40256077 Date 2022-11-07