Avishi Gupta (19155)
Manvi Goel (19472)

Assignment 1
Information Retrieval

README

Question 1

<u>Input</u>
Two-line input.
Enter words in the first line and operators in the second. Enter operators separated by ", " and in uppercase.

<u>Output</u>
1. The number of documents retrieved.
2. The minimum number of total comparisons done (if any).
3. The list of document names retrieved.

<u>Libraries Used</u>
1. The library *os* for handling the directory
2. The libraries *nltk* and *string* for preprocessing.

<u>Assumptions</u>
1. The drive is loaded, and an appropriate path is provided to the folder.
2. The type and number of operators should be valid. The number of operators should be one less than the total number of words. Available operators are: AND, OR, AND NOT, OR NOT.
3. Words are input in space-separated format.
4. Operators are input in uppercase and separated by ", ". Example: AND, OR.
5. The number of valid tokens should be greater than two. Valid tokens are those words that are present in the given dataset.
6. The minimum number of comparisons in NOT operator is the length of the list.

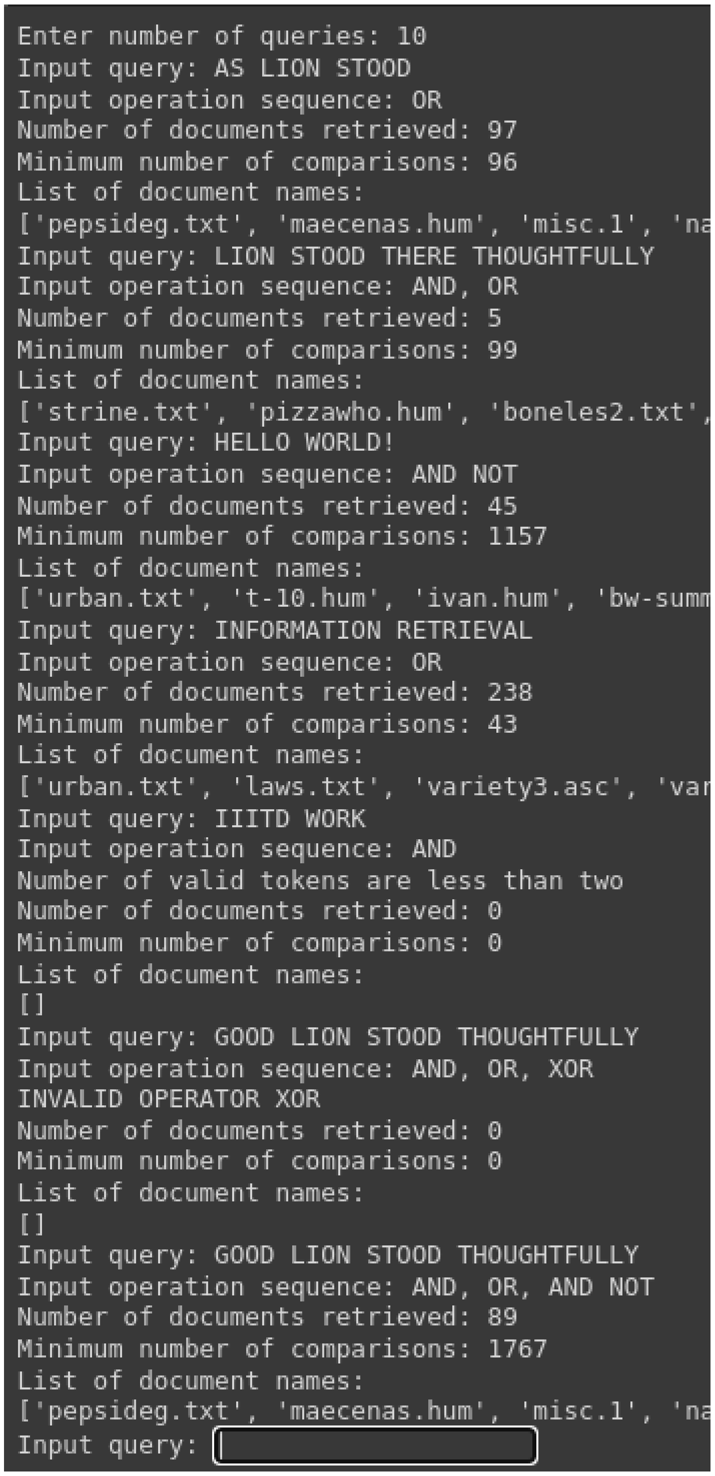
<u>Preprocessing Steps</u>
The preprocessing steps from the assignment are followed.
The following steps are done:
1. Converting the text to lowercase
2. Performing word tokenization
3. Removing stopwords
4. Removing punctuation marks
5. Removing the words with non-alphabet characters
6. Stripping the word of extra spaces.
7. Remove blank spaces.

<u>Methodology</u>
1. Process the raw data from each file and generate tokens

2. Use a dictionary to store the frequency, document IDs in which the token is present.
3. Process the input query.
4. Check if all the words from the input query are available in the dictionary. Check if the given inputs and operators are valid.
5. Apply operators on the list of documents for subsequent "tokens" and add the total number of comparisons for all operations.

Return the documents and number of comparisons that satisfy the above condition.

Sample Outputs for ques 1



```
Enter number of queries: 10
Input query: AS LION STOOD
Input operation sequence: OR
Number of documents retrieved: 97
Minimum number of comparisons: 96
List of document names:
['pepsideg.txt', 'maecenas.hum', 'misc.1', 'na
Input query: LION STOOD THERE THOUGHTFULLY
Input operation sequence: AND, OR
Number of documents retrieved: 5
Minimum number of comparisons: 99
List of document names:
['strine.txt', 'pizzawho.hum', 'boneles2.txt',
Input query: HELLO WORLD!
Input operation sequence: AND NOT
Number of documents retrieved: 45
Minimum number of comparisons: 1157
List of document names:
['urban.txt', 't-10.hum', 'ivan.hum', 'bw-summ
Input query: INFORMATION RETRIEVAL
Input operation sequence: OR
Number of documents retrieved: 238
Minimum number of comparisons: 43
List of document names:
['urban.txt', 'laws.txt', 'variety3.asc', 'var
Input query: IIITD WORK
Input operation sequence: AND
Number of valid tokens are less than two
Number of documents retrieved: 0
Minimum number of comparisons: 0
List of document names:
[]
Input query: GOOD LION STOOD THOUGHTFULLY
Input operation sequence: AND, OR, XOR
INVALID OPERATOR XOR
Number of documents retrieved: 0
Minimum number of comparisons: 0
List of document names:
[]
Input query: GOOD LION STOOD THOUGHTFULLY
Input operation sequence: AND, OR, AND NOT
Number of documents retrieved: 89
Minimum number of comparisons: 1767
List of document names:
['pepsideg.txt', 'maecenas.hum', 'misc.1', 'na
Input query:
```

Question 2

Input
The input query phrase.

Output
Number of documents with the phrase and list of their names

Libraries Used
1. The library *os* for handling the directory
2. The libraries *nltk* and *string* for preprocessing.

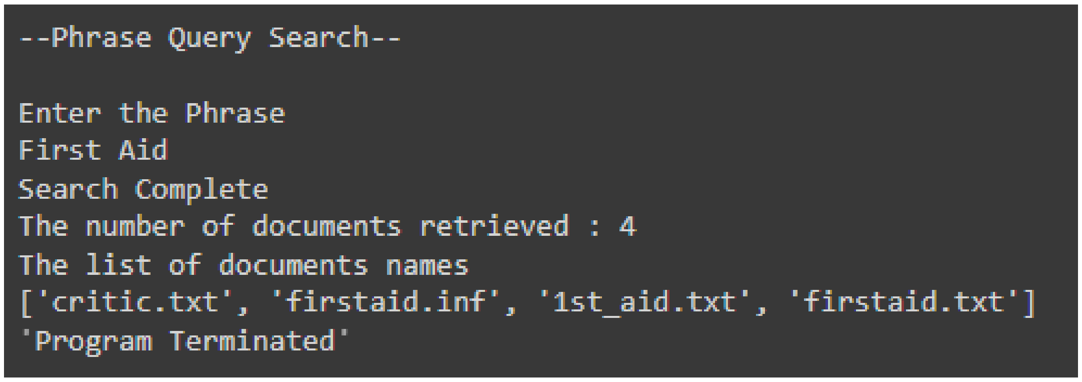
Assumptions
1. The drive is loaded, and an appropriate path is provided to the folder.
2. Each document is reported only once, even if the phrase occurs multiple times.
3. All words of the input query must be present in the dictionary for the query to be processed.
4. Words should be present in the input query after preprocessing.
5. Words should be space-separated.

Preprocessing Steps
The preprocessing steps from the assignment are followed.
The following steps are done:
1. Converting the text to lowercase
2. Performing word tokenization
3. Removing stopwords
4. Removing punctuation marks
5. Removing the words with non-alphabet characters
6. Stripping the word of extra spaces.
7. Remove blank spaces.

Methodology
1. Process the raw data from each file and generate tokens
2. Use a dictionary to store the frequency, document IDs in which the token is present, and the positions of each token in each document.
3. Process the input query.
4. Check if all the words from the input query are available in the dictionary.
5. For all the positions where the first word is present, check whether the subsequent tokens are present in their respective order.
6. Return the documents which satisfy the above condition.

Sample Outputs

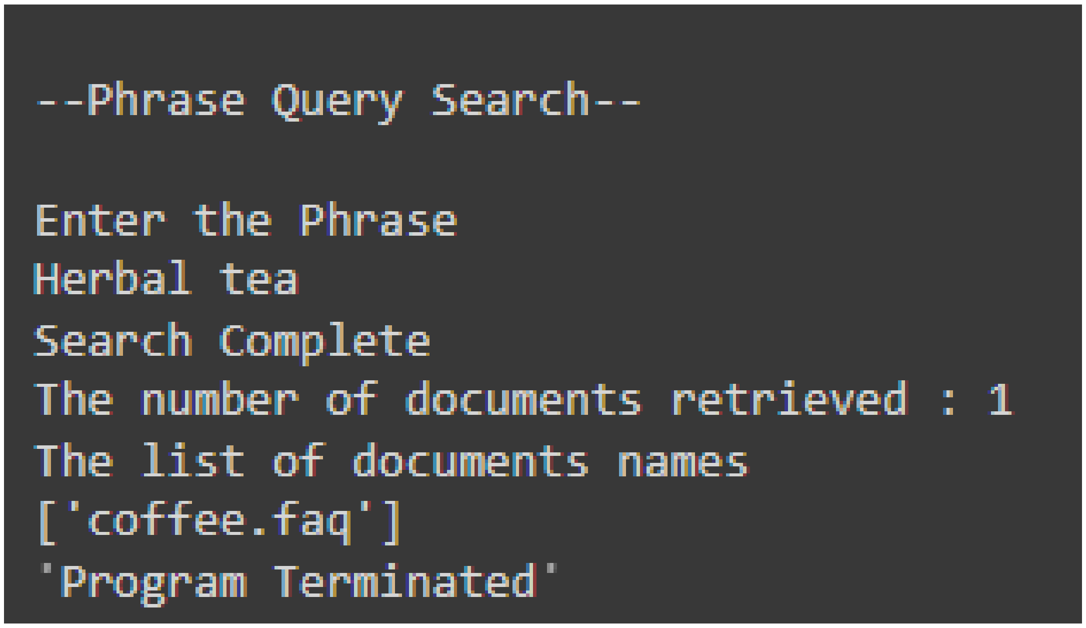
1. Input: *First Aid*



```
--Phrase Query Search--

Enter the Phrase
First Aid
Search Complete
The number of documents retrieved : 4
The list of documents names
['critic.txt', 'firstaid.inf', '1st_aid.txt', 'firstaid.txt']
'Program Terminated'
```

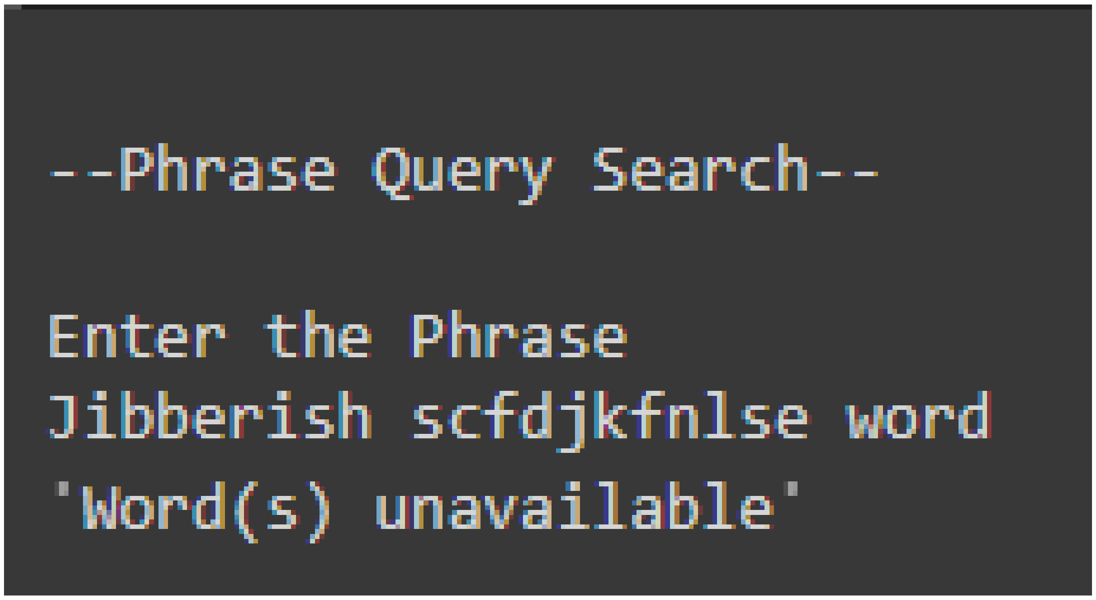
2. Input: *Herbal Tea*



```
--Phrase Query Search--

Enter the Phrase
Herbal tea
Search Complete
The number of documents retrieved : 1
The list of documents names
['coffee.faq']
'Program Terminated'
```

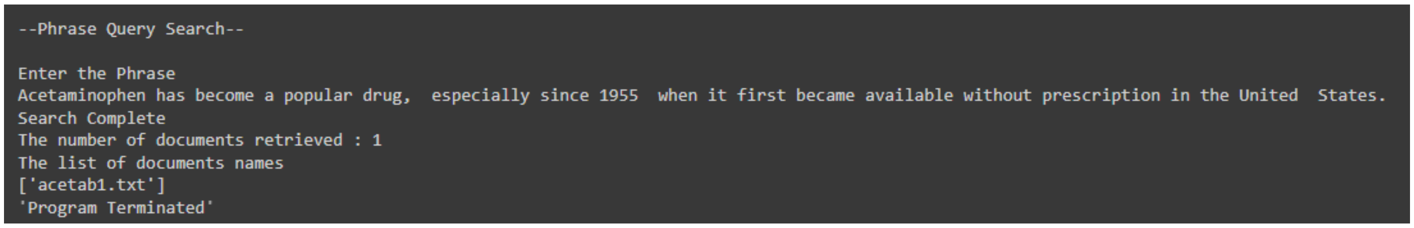
3. Input: *Jibberish scfdjkfnlse word*



```
--Phrase Query Search--

Enter the Phrase
Jibberish scfdjkfnlse word
'Word(s) unavailable'
```

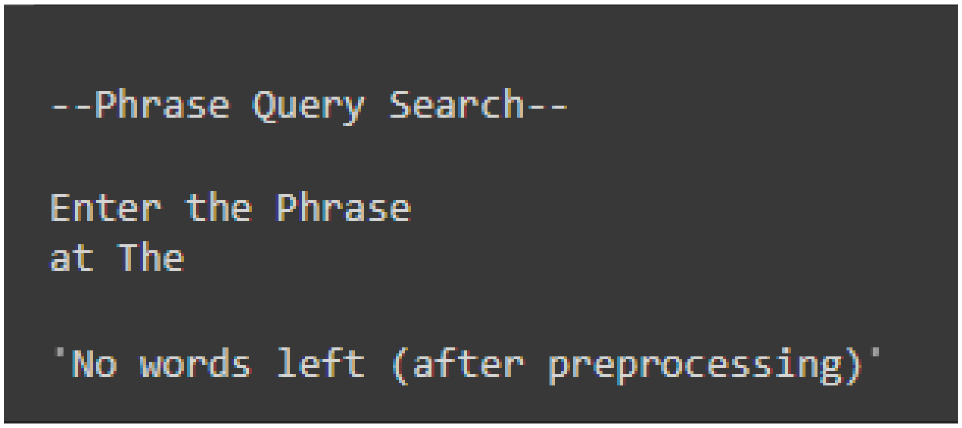
4. Input: *Acetaminophen has become a popular drug, especially since 1955, when it first became available without prescription in the United States.*



```
--Phrase Query Search--

Enter the Phrase
Acetaminophen has become a popular drug,  especially since 1955  when it first became available without prescription in the United  States.
Search Complete
The number of documents retrieved : 1
The list of documents names
['acetab1.txt']
'Program Terminated'
```

5. Input: *at the*



```
--Phrase Query Search--

Enter the Phrase
at The

'No words left (after preprocessing)'
```