CSE508 Winter 2024 Assignment 1 Report
Chaudhary Digvijay Daniel Singh
2020559

Question 1

For this section, as asked, I made a “preprocess” function that first turns the sentences into lower case. Then, I used the NLTK library to tokenize the words and remove the stopwords and also removed punctuations, blank space tokens. Then I read the files, preprocessed all the files using the “preprocess” function, and turned the tokenized lists to space separated strings so that it could be stored in text files and easily restored back by “split” when needed. The results are in the processed files folder.

Question 2

For this section, I made a Unigram Inverted Index by making a dictionary with the tokenized terms as keys and the documents (“1” for “file1.txt” and so on) it appears in as the data for that key. Then, turned it into a pickle file and loaded it back.

I implemented OR, OR_NOT, AND, AND_NOT functions using sets. Making use of Set properties.

For query answering, I preprocessed the Input Sequence, and also turned the Operations into a list, then I made a new list with the tokens of the Sequence in Even Places, and the Operations in the Odd places, turned this into a string gives us the Query. I then processed these queries by first taking the set for the 0th index Token as the initial set, then operating left to right finally getting our target set. This set only includes the suffix of the file names instead of the whole file, “1” for “file1.txt”, so that was converted back.

Results:

```
Input Number of Queries: 2
Input Sequence: perfect feedback sound
Comma seperated Operations: and, or
Query 1 : perfect AND feedback OR sound
```

```
Number of documents retrieved for query 1 : 183
Names of the documents retrieved for query 1 ; file512.txt, file513.txt,
file514.txt, file4.txt, file5.txt
```

```
Input Sequence: perfect feedback sound
Comma seperated Operations: and not, and or
Query 2 : perfect AND NOT feedback AND OR sound
Number of documents retrieved for query 2 : 836
Names of the documents retrieved for query 2 ; file1.txt, file2.txt,
file3.txt
```

Only showing some of the documents name retrieved here

Question 3

For this section, I made a Positional Index by making a dictionary with the tokenized terms as keys and the documents (“1” for “file1.txt” and so on) it appears in as the data for that key, they themselves also acting as dictionaries showcasing the indices where the token occurs in that file. Then, turned it into a pickle file and loaded it back.

For query handling, I first took the AND of sets of keys of the dictionary of Phrase tokens. This gives us the files in which all the tokens are present. Then for each file, I took the sets of position of each token, subtracted “i” from the “i”th token so that if the second and third token are in 26th and 27th place, they would coincide with the first token at 25th place. I took the AND of these sets, if this AND had any items then this meant that this particular file had these tokens in this exact order and was added to our answer.

Results:

```
Input Number of Queries: 1
Input Sequence: true perfect sound
Number of documents retrieved for query 1 : 1
Names of the documents retrieved for query 1 ; file310.txt
```