REPORT

Files included: preprocessing.py, query.py, inverted_index.json, doc_id_doc_name.json, skips_vs_time_window_and_system.png, word_cloud.png. skips_vs_comparison_window_and_system.png, skips_vs_time_wonder_and_sick.png, skips_vs_comparison_wonder_and_sick.png

Assumptions

1. Header is assumed to be everything above the first blank line.
2. Stop words present in the NLTK corpus are considered.
3. Document IDs are provided
4. Document Frequency is not needed in this assignment. Hence not stored.

Pre-processing Steps

Removal of Header

(All the lines before the first blank line are removed)

↓

Removal of Punctuation marks, comma, etc

(They are removed through regular expression)

↓

Tokenization

(Tokens are formed using RegexpTokenizer and words length>2 are considered)

↓

Removal of Stop Words

(Stop words are removed using NLTK stop words)

↓

Normalization

(All token are converted into lower case)

↓

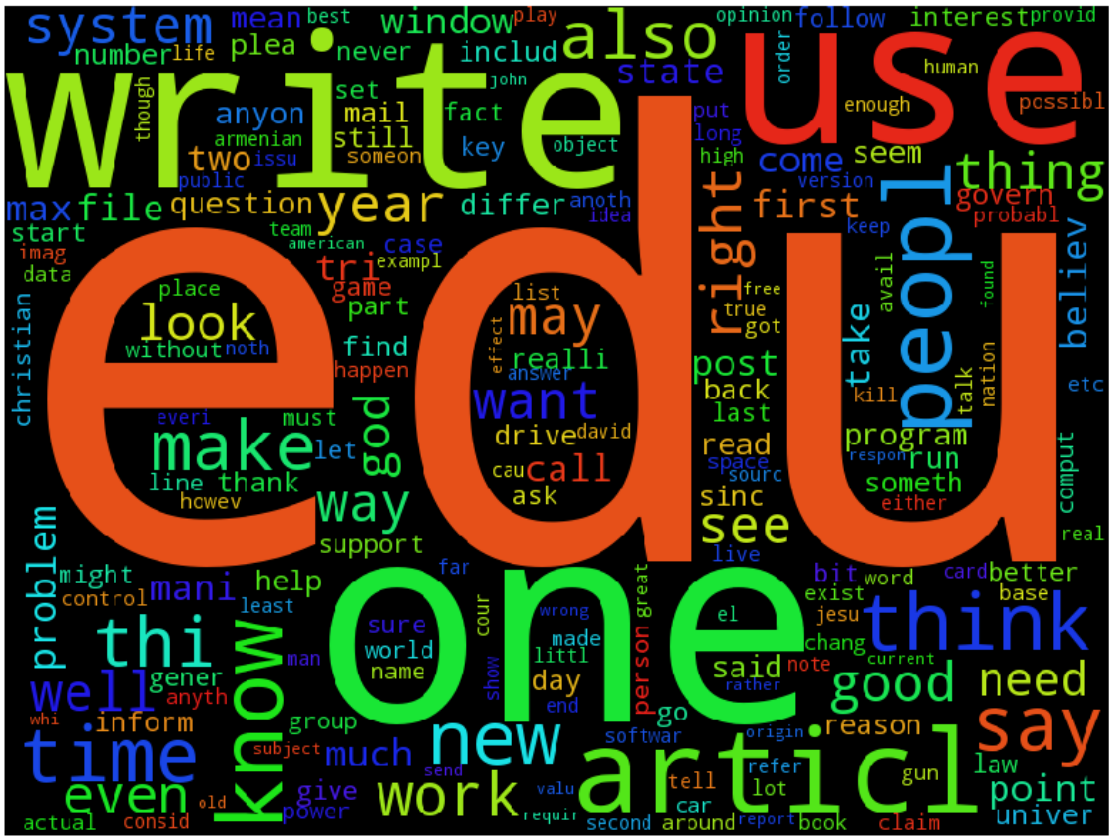
Stemming

(Stemming is performed using Porter algorithm to get the root word)

Number of Documents: 19997

Number of Tokens: 143700

Word Cloud



Word cloud is created from the words retrieved after the pre-processing steps.

Inverted Index

Inverted index is created using dictionary with token as key and a list of Document IDs as value and dumped into a json file.

Another dictionary is also maintained to map the Document ID with the corresponding file.

Query

1. X OR Y
   merge_union() function is used to return the list of all Document IDs.

2. X and Y
   merge_intersect() function is used.

3. X and not Y
   merge_and_not() function is used.

4. X or not Y
   merge_or_not() function is used.

Skip pointer algorithm

Query: wonder and sick

Query: wonder and sick

Length of first word posting list: 1309

Length of second word posting list: 237

Time for Intersect Using Merge Algortihm : 0.000877121109625

Number of documents retrieved using merge algorithm: 28

Number of skips for first posting list: 12

Number of skips for second posting list: 5

Time taken: 0.00194001216551

Comparisons performed: 1514

Number of documents retrieved: 28

Number of skips for first posting list: 18

Number of skips for second posting list: 7

Time taken: 0.00175990593743

Comparisons performed: 1514

Number of documents retrieved: 28

Number of skips for first posting list: 36

Number of skips for second posting list: 15

Time taken: 0.00156054306808

Comparisons performed: 1412

Number of documents retrieved: 28

Number of skips for first posting list: 72

Number of skips for second posting list: 30

Time taken: 0.00147143390586

Comparisons performed: 1237

Number of documents retrieved: 28

Number of skips for first posting list: 108

Number of skips for second posting list: 45

Time taken: 0.00129359316452

Comparisons performed: 1098

Number of documents retrieved: 28

Number of skips for first posting list: 144

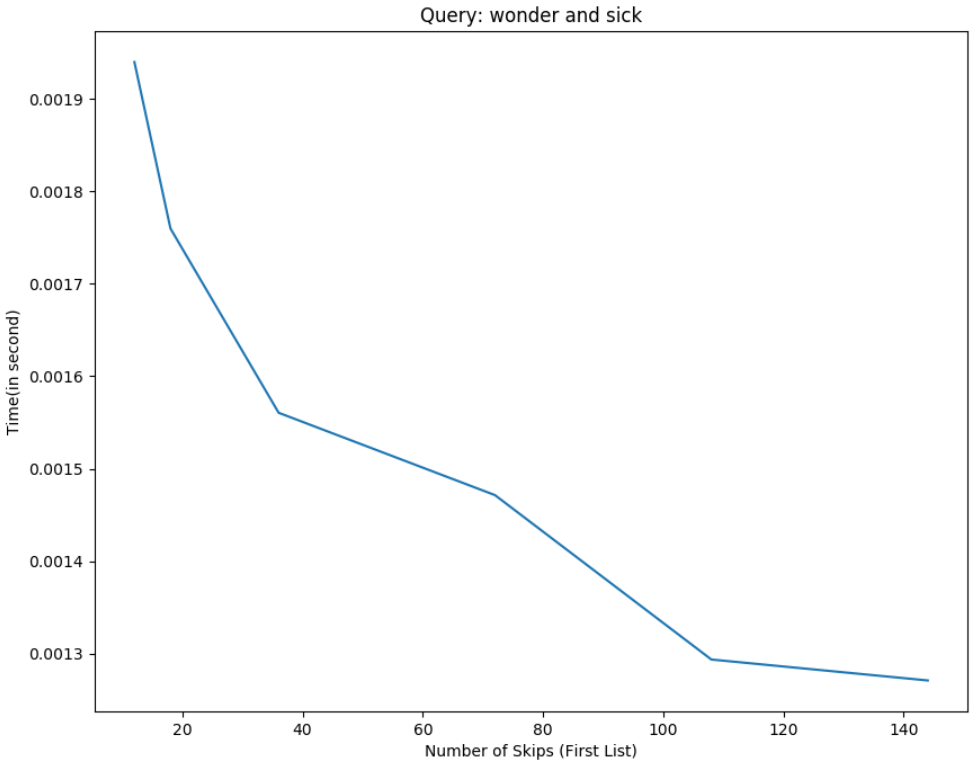
Number of skips for second posting list: 60

Time taken: 0.00127093829269

Comparisons performed: 1027

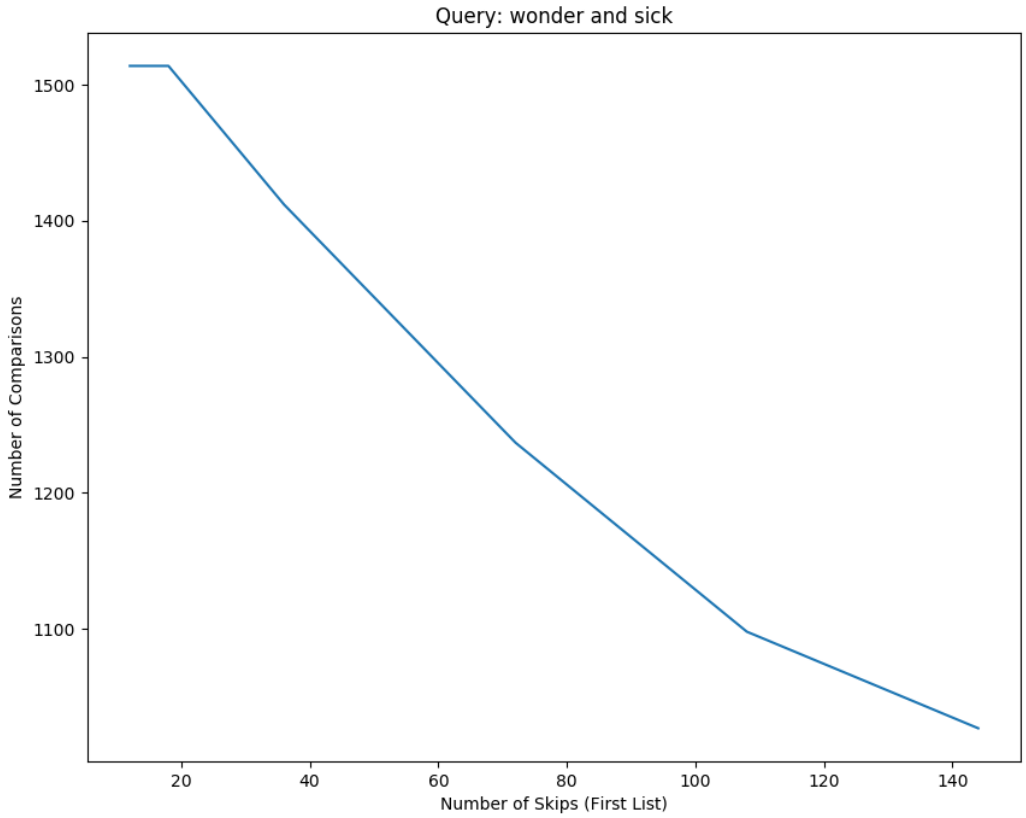
Number of documents retrieved: 28

Time to execute the skip pointer algorithm for the Query: wonder and sick is shown in the graph below.



We can infer from the above graph that the time to execute the AND query decreases with the increase in number of skips.

Number of comparisons required to execute the skip pointer algorithm for the Query: wonder and sick is shown in the graph below



We can infer that the number of comparisons also reduces with the increase in number of skips.

Another Query: window and system

Query: window and system

Length of first word posting list: 1671

Length of second word posting list: 2779

Time for Intersect Using Merge Algortihm : 0.00357644905262

Number of documents retrieved using merge algorithm: 466

Number of skips for first posting list: 13

Number of skips for second posting list: 17

Time taken: 0.00828337619487

Comparisons performed: 3982

Number of documents retrieved: 466

Number of skips for first posting list: 20

Number of skips for second posting list: 26

Time taken: 0.00620554689431

Comparisons performed: 3982

Number of documents retrieved: 466

Number of skips for first posting list: 40

Number of skips for second posting list: 52

Time taken: 0.00557649662778

Comparisons performed: 3931

Number of documents retrieved: 466

Number of skips for first posting list: 80

Number of skips for second posting list: 104

Time taken: 0.00559197745679

Comparisons performed: 3763

Number of documents retrieved: 466

Number of skips for first posting list: 120

Number of skips for second posting list: 156

Time taken: 0.00535976502351

Comparisons performed: 3622

Number of documents retrieved: 466

Number of skips for first posting list: 160

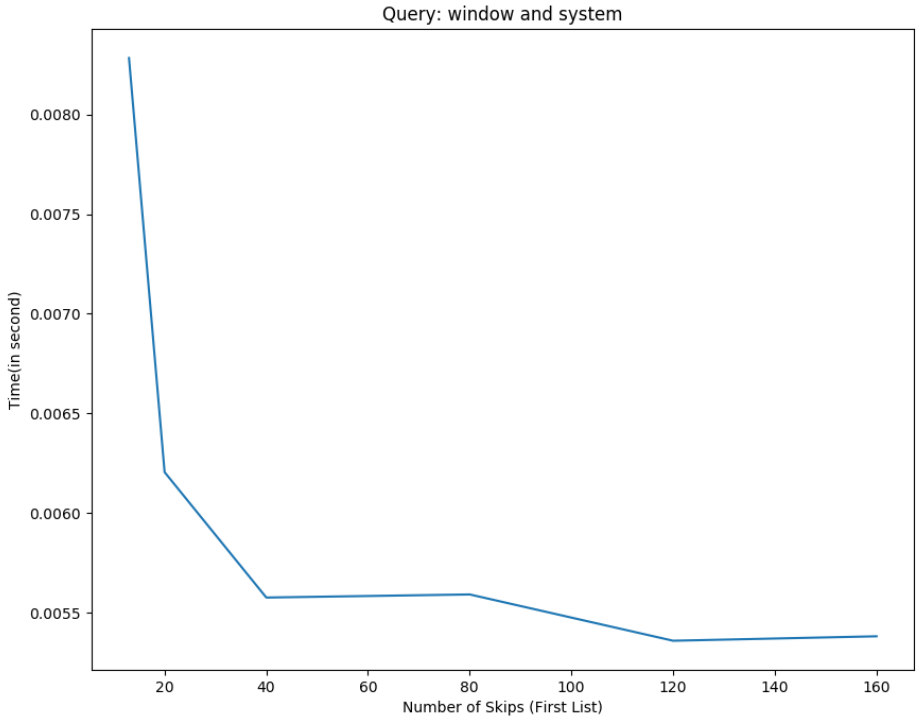
Number of skips for second posting list: 208

Time taken: 0.00538204231361

Comparisons performed: 3455

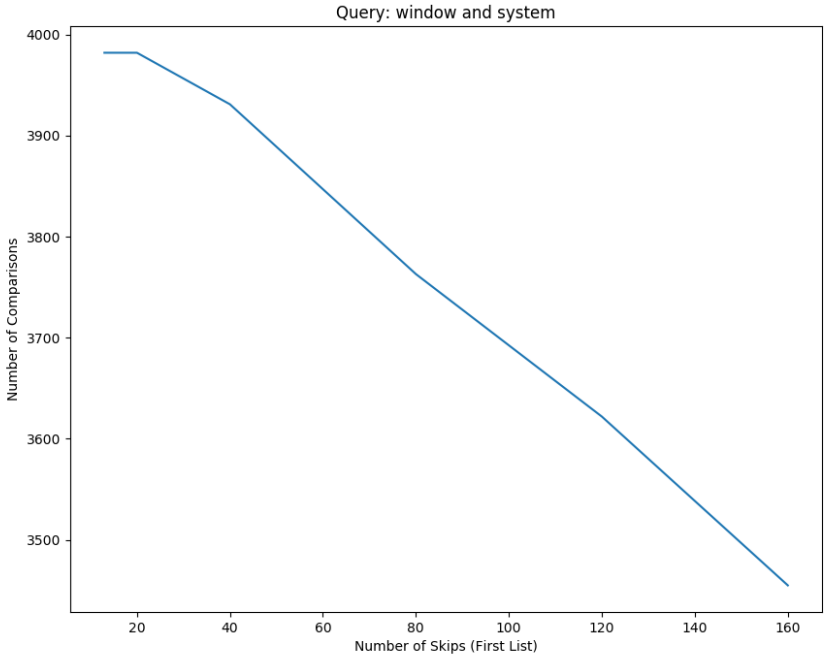
Number of documents retrieved: 466

Time to execute the skip pointer algorithm for the Query: window and system is shown in the graph below:



We can infer from the above graph that the time to execute the AND query decreases with the increase in number of skips.

Number of comparisons required to execute the skip pointer algorithm for the Query: window and system is shown in the graph below:



We can infer that the number of comparisons also reduces with the increase in number of skips.

When the number of skips are large, then the time to execute AND query and number of comparisons required reduces.

References

Pseudo Code for Merge and Skip pointer algorithm is taken from nlp.stanford.edu.