IR Assignment 1

Question-1

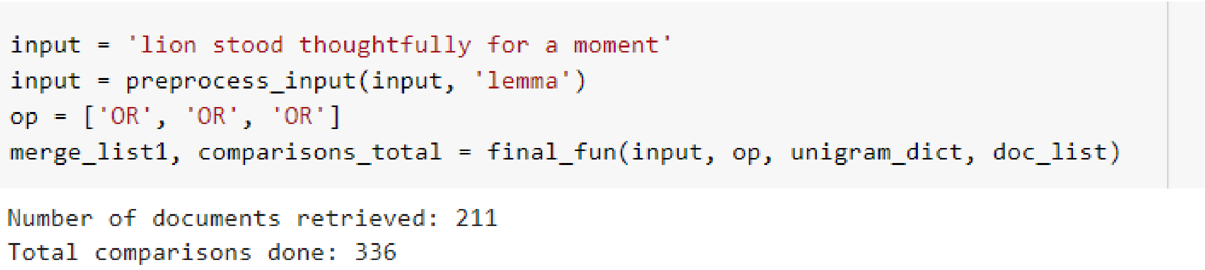
a) PreProcessing :

Pre-processing is done on the text contained in all the given text files.

1. **Converting to lower case :** the whole text contained into the text files is converted to lower case.
2. **Word Tokenization :** Tokenization refers to the breaking of the sentences into tokens, tokes here refers to the words contained in the sentence.
3. **Removing Stop words :** Stop words like ‘a’, ‘an’, ‘the’ are removed from the text using a defined function.
4. **Removing Punctuation marks :** After that punctuation marks are removed.
5. **Removing Blank space token :** Lastly blank spaces are filtered out.

All the above preprocessing is performed by defining a preprocessing function and a message “Preprocessing Complete” is printed in the output.

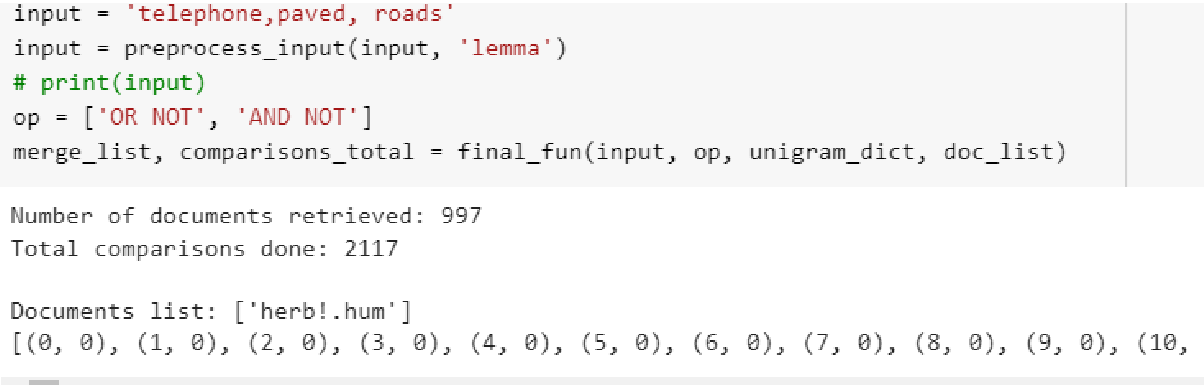
c) Output for the first query



```
input = 'lion stood thoughtfully for a moment'
input = preprocess_input(input, 'lemma')
op = ['OR', 'OR', 'OR']
merge_list1, comparisons_total = final_fun(input, op, unigram_dict, doc_list)
```

```
Number of documents retrieved: 211
Total comparisons done: 336
```

d) Output for the second query

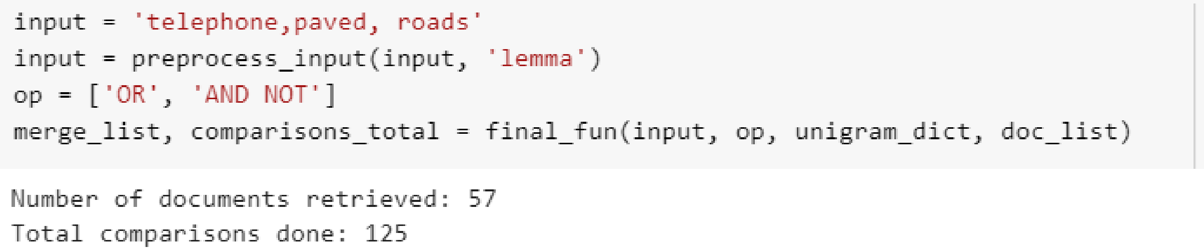


```
input = 'telephone,paved, roads'
input = preprocess_input(input, 'lemma')
# print(input)
op = ['OR NOT', 'AND NOT']
merge_list, comparisons_total = final_fun(input, op, unigram_dict, doc_list)
```

```
Number of documents retrieved: 997
Total comparisons done: 2117

Documents list: ['herb!.hum']
[(0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (5, 0), (6, 0), (7, 0), (8, 0), (9, 0), (10,
```

e) Output for another query

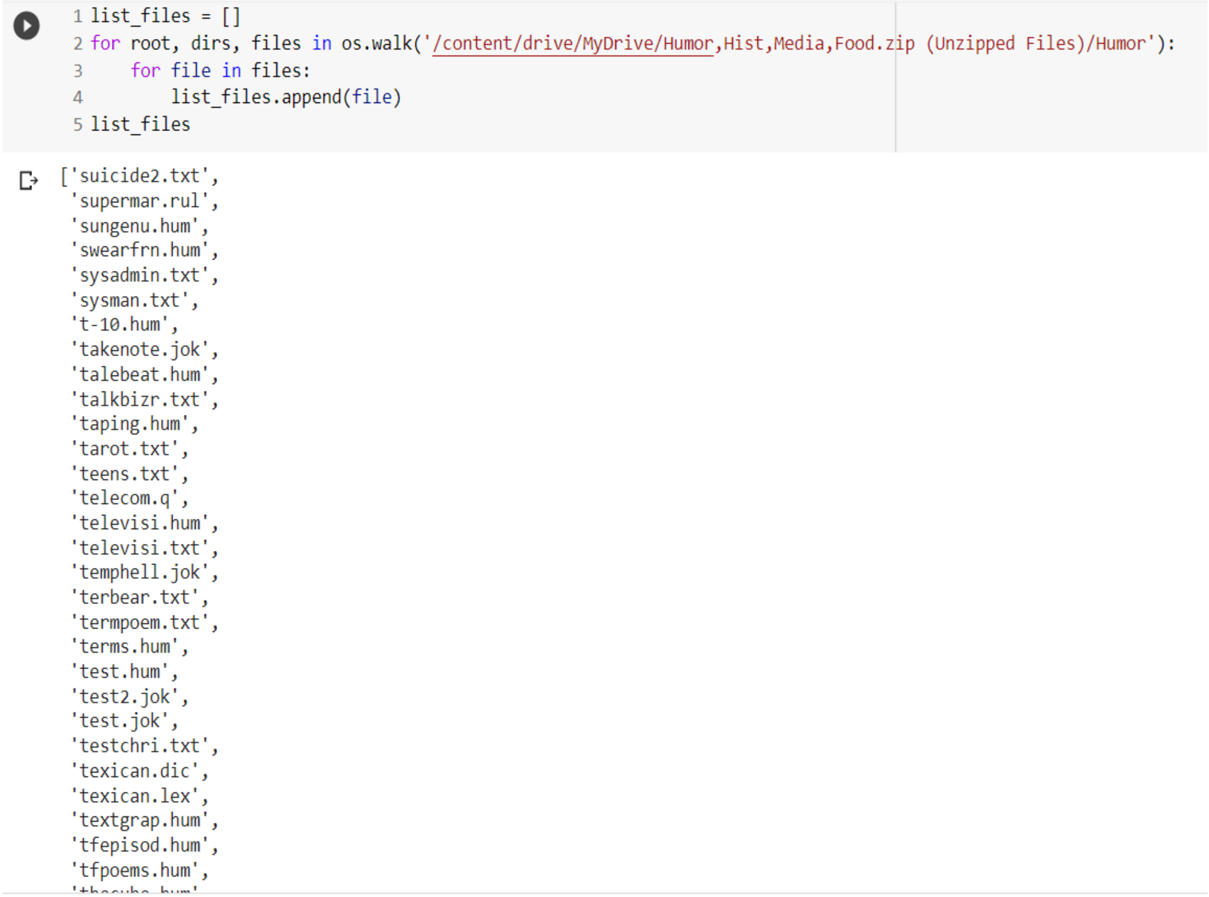


```
input = 'telephone,paved, roads'
input = preprocess_input(input, 'lemma')
op = ['OR', 'AND NOT']
merge_list, comparisons_total = final_fun(input, op, unigram_dict, doc_list)
```

```
Number of documents retrieved: 57
Total comparisons done: 125
```

Question-2

Firstly we tried printing all the text files



```
list_files = []
for root, dirs, files in os.walk('/content/drive/MyDrive/Humor,Hist,Media,Food.zip (Unzipped Files)/Humor'):
    for file in files:
        list_files.append(file)
list_files
```

```
['suicide2.txt',
 'supermar.rul',
 'sungenu.hum',
 'swearfrn.hum',
 'sysadmin.txt',
 'sysman.txt',
 't-10.hum',
 'takenote.jok',
 'talebeat.hum',
 'talkbizr.txt',
 'taping.hum',
 'tarot.txt',
 'teens.txt',
 'telecom.q',
 'televisi.hum',
 'televisi.txt',
 'temphell.jok',
 'terbear.txt',
 'termpoem.txt',
 'terms.hum',
 'test.hum',
 'test2.jok',
 'test.jok',
 'testchri.txt',
 'texican.dic',
 'texican.lex',
 'textgrap.hum',
 'tfepisod.hum',
 'tfpoems.hum',
 'thecube.hum'
```

b) PreProcessing :

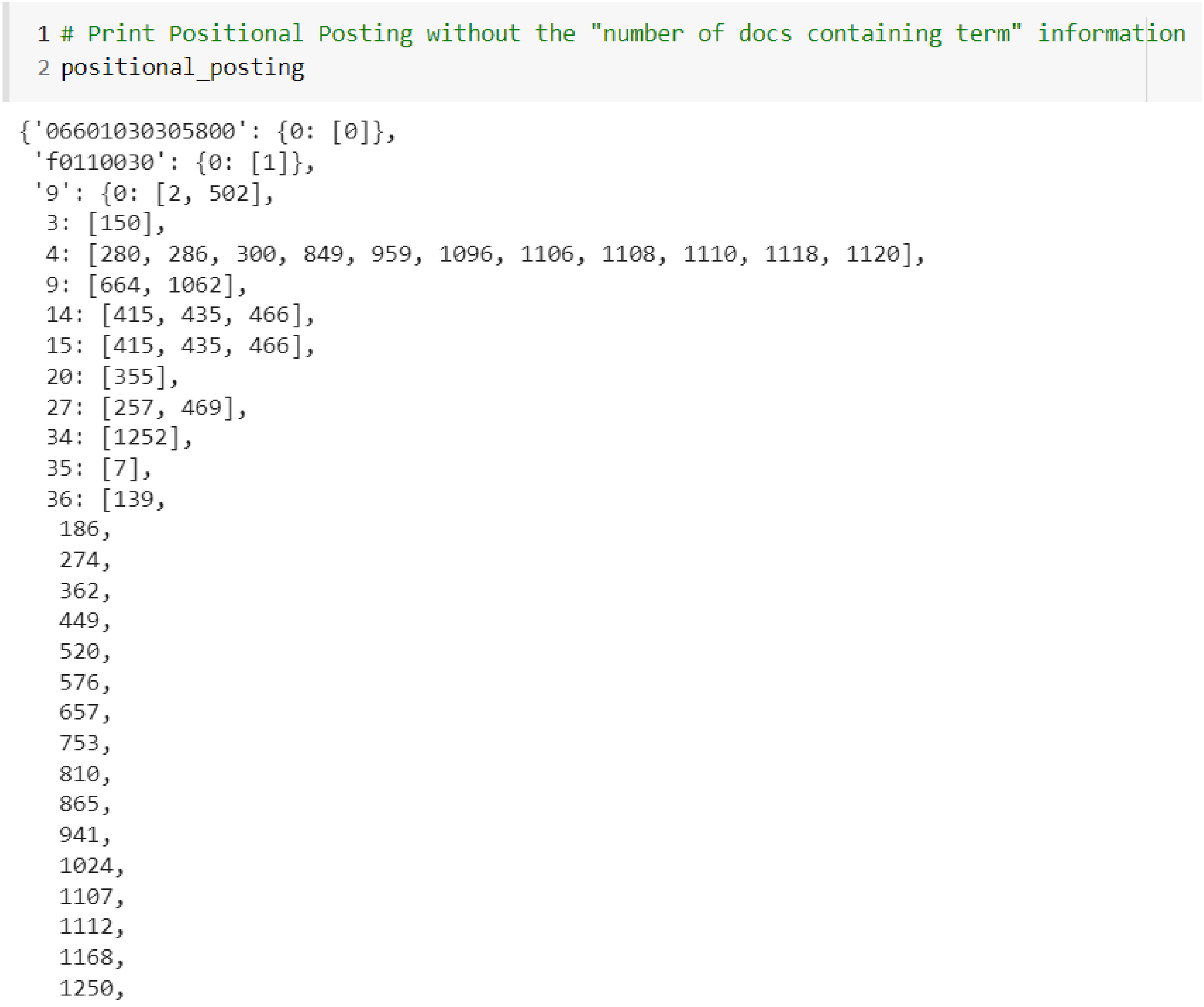
Pre-processing is done on the text contained in all the given text files.

6. **Converting to lower case :** the whole text contained into the text files is converted to lower case.
7. **Word Tokenization :** Tokenization refers to the breaking of the sentences into tokens, tokes here refers to the words contained in the sentence.
8. **Removing Stop words :** Stop words like ‘a’, ‘an’, ‘the’ are removed from the text using a defined function.
9. **Removing Punctuation marks :** After that punctuation marks are removed.
10. **Removing Blanck space token :** Lastly blank spaces are filtered out.

All the above preprocessing is performed by defining a preprocessing function and a message “Preprocessing Complete” is printed in the output.

b) Positional indexes :

Now positional indexes of word in a document are found and printed them as output.

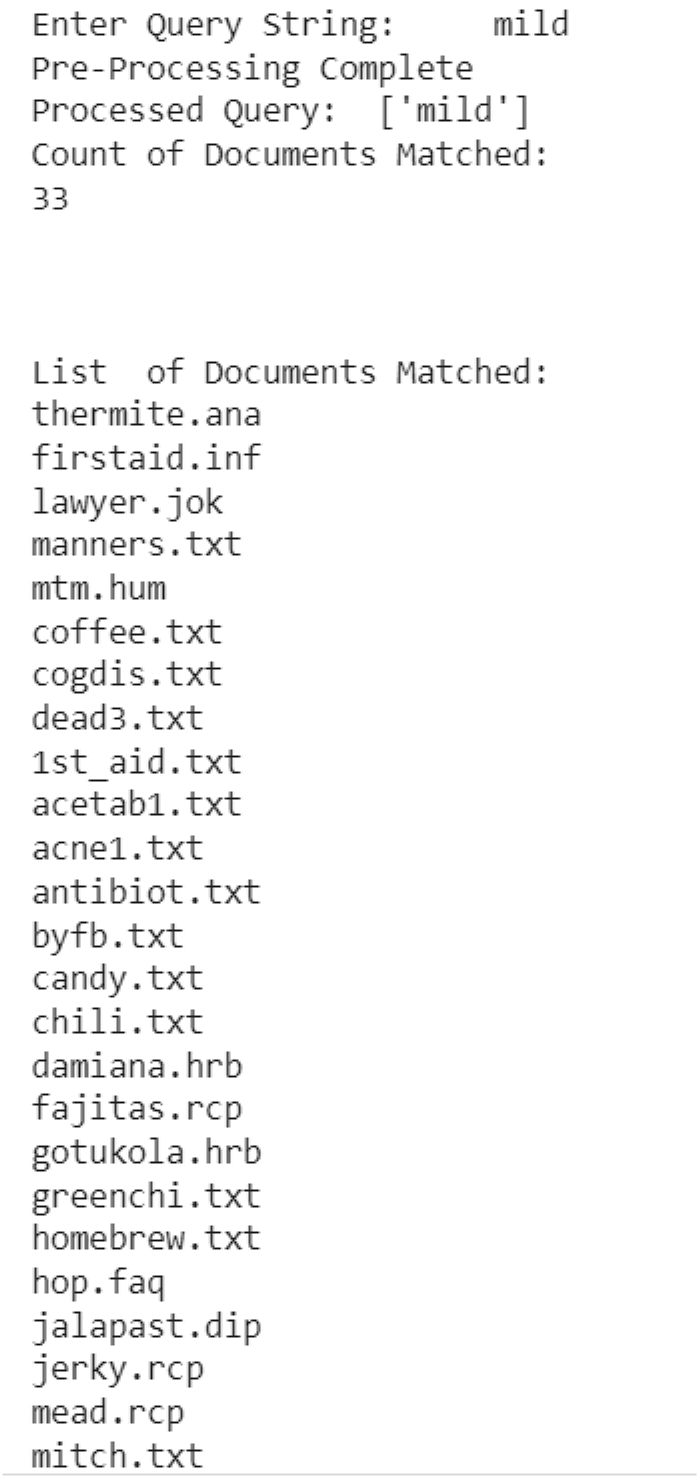


```
# Print Positional Posting without the "number of docs containing term" information
positional_posting
```

```
{'06601030305800': {0: [0]},
 'f0110030': {0: [1]},
 '9': {0: [2, 502],
  3: [150],
  4: [280, 286, 300, 849, 959, 1096, 1106, 1108, 1110, 1118, 1120],
  9: [664, 1062],
  14: [415, 435, 466],
  15: [415, 435, 466],
  20: [355],
  27: [257, 469],
  34: [1252],
  35: [7],
  36: [139,
   186,
   274,
   362,
   449,
   520,
   576,
   657,
   753,
   810,
   865,
   941,
   1024,
   1107,
   1112,
   1168,
   1250,
```

c) Processing the queries by asking the user to enter the query as User Interface.
Queries are assumend to be of length less than or equal to 5.
It will retrieve the number of documents those contain that query and will print the list of documents matched with the entered query.

Query 1 : Below is the first query entered that is ‘mild’
Total 33 documents got matched with the query. And the list of the documents is pritnted below.



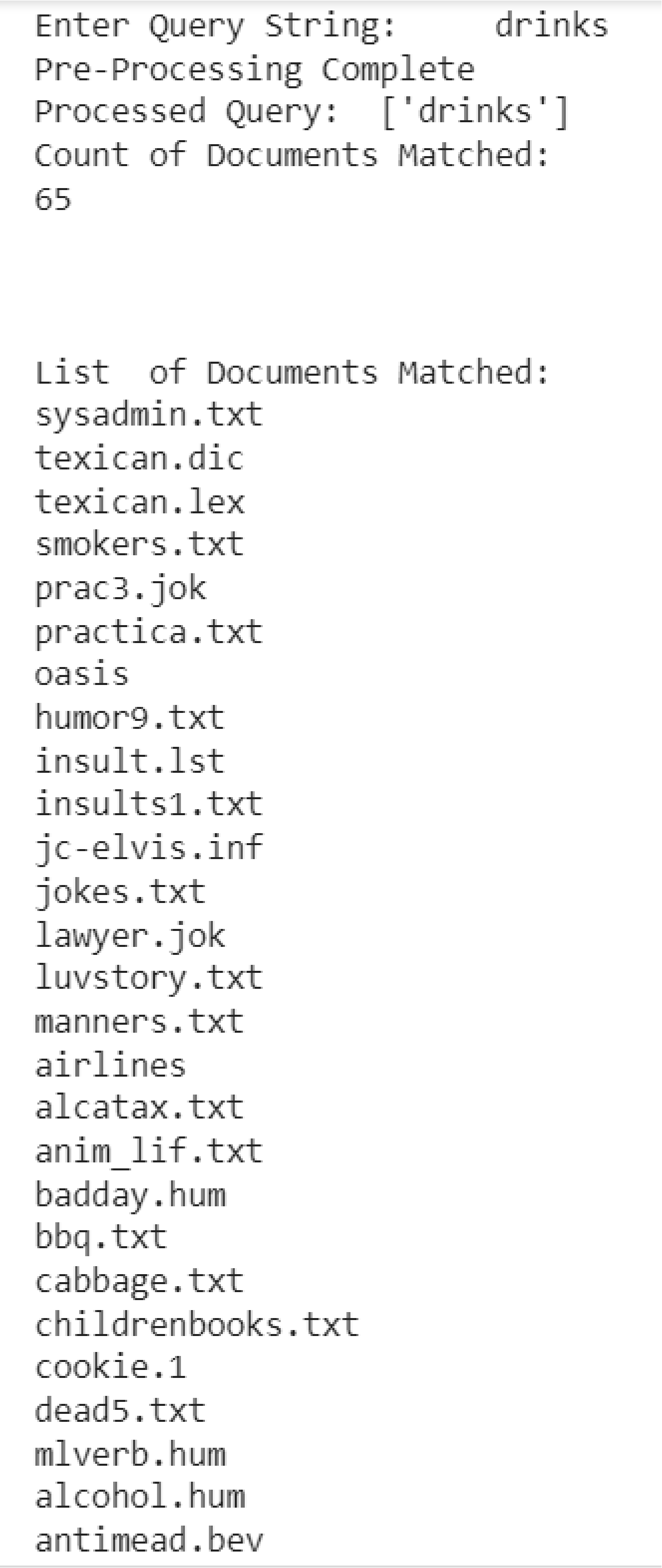
```
Enter Query String:     mild
Pre-Processing Complete
Processed Query:  ['mild']
Count of Documents Matched:
33



List  of Documents Matched:
thermite.ana
firstaid.inf
lawyer.jok
manners.txt
mtm.hum
coffee.txt
cogdis.txt
dead3.txt
1st_aid.txt
acetab1.txt
acne1.txt
antibiot.txt
byfb.txt
candy.txt
chili.txt
damiana.hrb
fajitas.rcp
gotukola.hrb
greenchi.txt
homebrew.txt
hop.faq
jalapast.dip
jerky.rcp
mead.rcp
mitch.txt
```

Query 2: Another query ‘drinks’ is entered to cheek whether it is there in the documents.

Total of 65 documents retrieved which match the entered query and the list of the documents.



```
Enter Query String:     drinks
Pre-Processing Complete
Processed Query:  ['drinks']
Count of Documents Matched:
65



List  of Documents Matched:
sysadmin.txt
texican.dic
texican.lex
smokers.txt
prac3.jok
practica.txt
oasis
humor9.txt
insult.lst
insults1.txt
jc-elvis.inf
jokes.txt
lawyer.jok
luvstory.txt
manners.txt
airlines
alcatax.txt
anim_lif.txt
badday.hum
bbq.txt
cabbage.txt
childrenbooks.txt
cookie.1
dead5.txt
mlverb.hum
alcohol.hum
antimead.bev
```