UE19CS332 : Algorithms for Web and Information Retrieval
ASSIGNMENT - 3

Problem definition:

- **Build a search engine for any 3 corpora of your choice,**
- **Your Code should be able to: Search for the terms in the query – Create Postings list**
- **Fill the Inverted Index**
- **Retrieve the data from the dictionary – Query response time.**

TEAM MEMBERS :

NAME	SRN	SECTION
Priya Mohata	PES2UG19CS301	E
R Sharmila	PES2UG19CS309	E
Ritik	PES2UG19CS332	E

CORPUS – 3 : *NEWS HEADLINES ANALYSIS CORPUS*

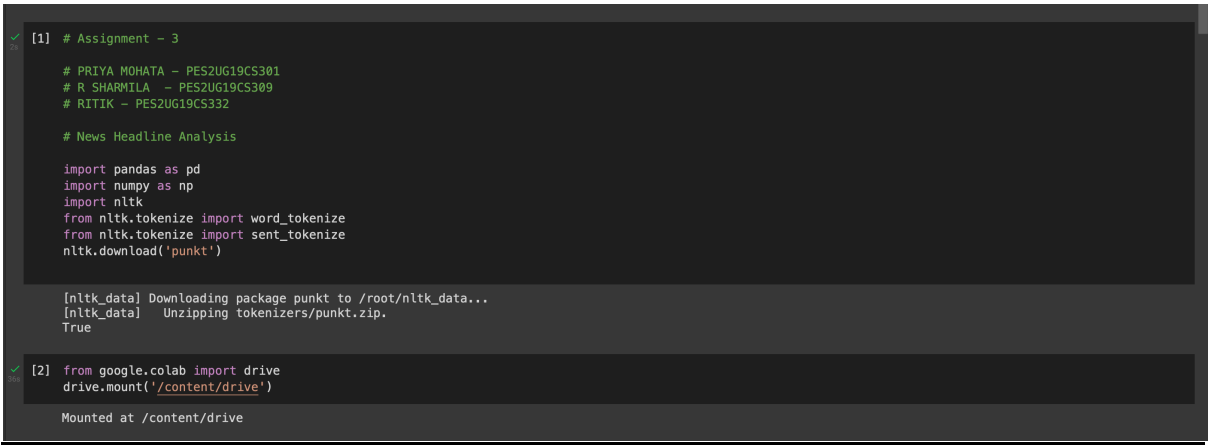
DATASET LOCATION : https://drive.google.com/file/d/1dL-jz_hwksiA4uhTiBPgWj_fUQB_QdEJ/view?usp=sharing

NOTEBOOK NAME : A3_P3.ipynb

Link to the notebook :
https://colab.research.google.com/drive/1_Up7II6U7diJtqLkW7P3KYeRnIEoUsnd?usp=sharing

STEPS :

Importing all libraries



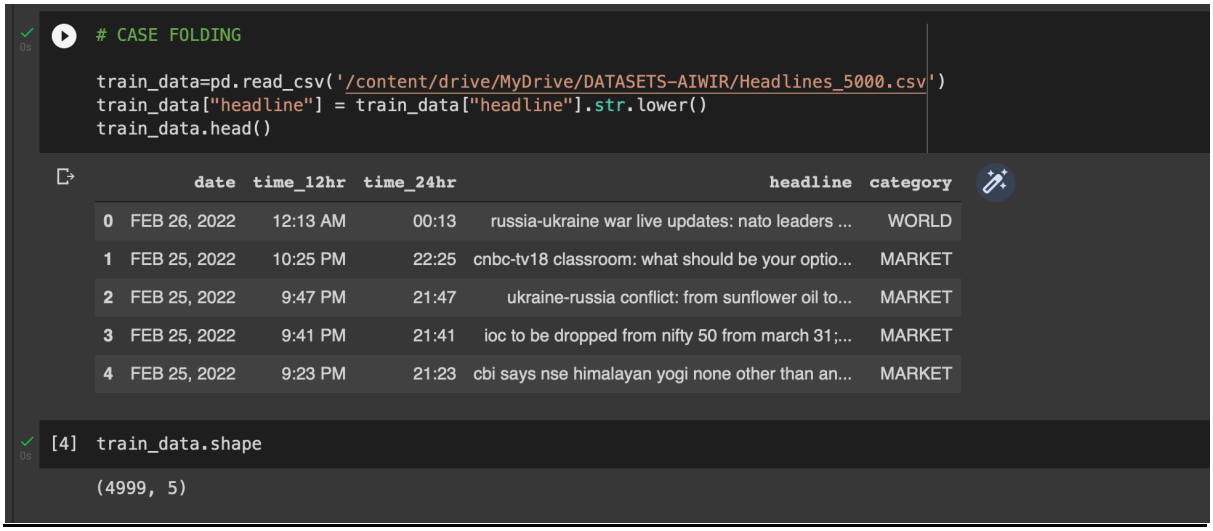
```
[1] # Assignment - 3

    # PRIYA MOHATA - PES2UG19CS301
    # R SHARMILA  - PES2UG19CS309
    # RITIK - PES2UG19CS332

    # News Headline Analysis

    import pandas as pd
    import numpy as np
    import nltk
    from nltk.tokenize import word_tokenize
    from nltk.tokenize import sent_tokenize
    nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
[2] from google.colab import drive
    drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Case folding



```
# CASE FOLDING

train_data=pd.read_csv('/content/drive/MyDrive/DATASETS-AIWIR/Headlines_5000.csv')
train_data["headline"] = train_data["headline"].str.lower()
train_data.head()
```

	date	time_12hr	time_24hr	headline	category
0	FEB 26, 2022	12:13 AM	00:13	russia-ukraine war live updates: nato leaders ...	WORLD
1	FEB 25, 2022	10:25 PM	22:25	cnbc-tv18 classroom: what should be your optio...	MARKET
2	FEB 25, 2022	9:47 PM	21:47	ukraine-russia conflict: from sunflower oil to...	MARKET
3	FEB 25, 2022	9:41 PM	21:41	ioc to be dropped from nifty 50 from march 31;...	MARKET
4	FEB 25, 2022	9:23 PM	21:23	cbi says nse himalayan yogi none other than an...	MARKET

```
[4] train_data.shape
```

```
(4999, 5)
```

Renaming Columns

```
[5]  train_data.rename(columns = {'headline':'text'}, inplace = True)

[6]  train_data.columns

     Index(['date', 'time_12hr', 'time_24hr', 'text', 'category'], dtype='object')
```

Sentence Tokenization

```
[7]  # SENTENCE TOKENIZATION
     df=train_data['text']
     l=list()
     for line in df:
       token=sent_tokenize(line)
       l.append(token)

[8]  df=train_data['text']
     train_data['sent_token']=l

[9]  train_data.head()
```

	date	time_12hr	time_24hr	text	category	sent_token
0	FEB 26, 2022	12:13 AM	00:13	russia-ukraine war live updates: nato leaders ...	WORLD	[russia-ukraine war live updates: nato leaders...
1	FEB 25, 2022	10:25 PM	22:25	cnbc-tv18 classroom: what should be your optio...	MARKET	[cnbc-tv18 classroom: what should be your opti...
2	FEB 25, 2022	9:47 PM	21:47	ukraine-russia conflict: from sunflower oil to...	MARKET	[ukraine-russia conflict: from sunflower oil t...
3	FEB 25, 2022	9:41 PM	21:41	ioc to be dropped from nifty 50 from march 31;...	MARKET	[ioc to be dropped from nifty 50 from march 31...
4	FEB 25, 2022	9:23 PM	21:23	cbi says nse himalayan yogi none other than an...	MARKET	[cbi says nse himalayan yogi none other than a...

Word Tokenization

```
[10] # WORD TOKENIZATION
     df=train_data['text']
     l1=list()
     for line in df:
       tokens=word_tokenize(line)
       l1.append(tokens)

[11] df=train_data['text']
     train_data['word_token']=l1

[12] train_data.head()
```

	date	time_12hr	time_24hr	text	category	sent_token	word_token
0	FEB 26, 2022	12:13 AM	00:13	russia-ukraine war live updates: nato leaders ...	WORLD	[russia-ukraine war live updates: nato leaders...	[russia-ukraine, war, live, updates, :, nato, ...
1	FEB 25, 2022	10:25 PM	22:25	cnbc-tv18 classroom: what should be your optio...	MARKET	[cnbc-tv18 classroom: what should be your opti...	[cnbc-tv18, classroom, :, what, should, be, yo...
2	FEB 25, 2022	9:47 PM	21:47	ukraine-russia conflict: from sunflower oil to...	MARKET	[ukraine-russia conflict: from sunflower oil t...	[ukraine-russia, conflict, :, from, sunflower,...
3	FEB 25, 2022	9:41 PM	21:41	ioc to be dropped from nifty 50 from march 31;...	MARKET	[ioc to be dropped from nifty 50 from march 31...	[ioc, to, be, dropped, from, nifty, 50, from, ...
4	FEB 25, 2022	9:23 PM	21:23	cbi says nse himalayan yogi none other than an...	MARKET	[cbi says nse himalayan yogi none other than a...	[cbi, says, nse, himalayan, yogi, none, other,...

Stop Words Removal

```
[13] # STOP WORDS REMOVAL
     import nltk
     nltk.download('stopwords')
     from nltk.corpus import stopwords
     stoplist= stopwords.words('english')

     [nltk_data] Downloading package stopwords to /root/nltk_data...
     [nltk_data]   Unzipping corpora/stopwords.zip.

     stoplist=set(stoplist)
     l2=list()
     for i in l1:
       output = [w for w in i if not w in stoplist]
       l2.append(output)
     train_data['stop_words_removed']=l2

[15] train_data.head()
```

	date	time_12hr	time_24hr	text	category	sent_token	word_token	stop_words_removed
0	FEB 26, 2022	12:13 AM	00:13	russia-ukraine war live updates: nato leaders ...	WORLD	[russia-ukraine war live updates: nato leaders...	[russia-ukraine, war, live, updates, :, nato, ...	[russia-ukraine, war, live, updates, :, nato, ...
1	FEB 25, 2022	10:25 PM	22:25	cnbc-tv18 classroom: what should be your optio...	MARKET	[cnbc-tv18 classroom: what should be your opti...	[cnbc-tv18, classroom, :, what, should, be, yo...	[cnbc-tv18, classroom, :, options, trading, ex...
2	FEB 25, 2022	9:47 PM	21:47	ukraine-russia conflict: from sunflower oil to...	MARKET	[ukraine-russia conflict: from sunflower oil t...	[ukraine-russia, conflict, :, from, sunflower,...	[ukraine-russia, conflict, :, sunflower, oil, ...
3	FEB 25, 2022	9:41 PM	21:41	ioc to be dropped from nifty 50 from march 31;...	MARKET	[ioc to be dropped from nifty 50 from march 31...	[ioc, to, be, dropped, from, nifty, 50, from, ...	[ioc, dropped, nifty, 50, march, 31, ;, apollo...
4	FEB 25, 2022	9:23 PM	21:23	cbi says nse himalayan yogi none other than an...	MARKET	[cbi says nse himalayan yogi none other than a...	[cbi, says, nse, himalayan, yogi, none, other,...	[cbi, says, nse, himalayan, yogi, none, anand,...

3s completed at 5:11 PM

Stemming

```
# STEMMING
from nltk.stem import WordNetLemmatizer
from nltk.stem import PorterStemmer

# Stemming :
final_train_stem_list=[]
ps = PorterStemmer()
for line in train_data['stop_words_removed']:
  Stem_words=[]
  for i in line:
    rootWord = ps.stem(i)
    Stem_words.append(rootWord)
  Stem_words= [word for word in Stem_words if word.isalnum()]
  final_train_stem_list.append(Stem_words)

print(final_train_stem_list[0:5])

[['war', 'live', 'updat', 'nato', 'leader', 'meet', 'reassur', 'alli', 'near', 'russia', 'ukrain', 'air', 'india', 'oper', '3', 'flight', 'bucharest', 'evacu

[17] train_data['stemmed_words']=final_train_stem_list
     train_data.head()
```

```
train_data['stemmed_words']=final_train_stem_list
train_data.head()
```

	date	time_12hr	time_24hr	text	category	sent_token	word_token	stop_words_removed	stemmed_words
0	FEB 26, 2022	12:13 AM	00:13	russia-ukraine war live updates: nato leaders ...	WORLD	[russia-ukraine war live updates: nato leaders...	[russia-ukraine, war, live, updates, :, nato, ...	[russia-ukraine, war, live, updates, :, nato, ...	[war, live, updat, nato, leader, meet, reassur...
1	FEB 25, 2022	10:25 PM	22:25	cnbc-tv18 classroom: what should be your optio...	MARKET	[cnbc-tv18 classroom: what should be your opti...	[cnbc-tv18, classroom, :, what, should, be, yo...	[cnbc-tv18, classroom, :, options, trading, ex...	[classroom, option, trade, exit, strategi, exp...
2	FEB 25, 2022	9:47 PM	21:47	ukraine-russia conflict: from sunflower oil to...	MARKET	[ukraine-russia conflict: from sunflower oil t...	[ukraine-russia, conflict, :, from, sunflower,...	[ukraine-russia, conflict, :, sunflower, oil, ...	[conflict, sunflow, oil, beer, consum, good, c...
3	FEB 25, 2022	9:41 PM	21:41	ioc to be dropped from nifty 50 from march 31;...	MARKET	[ioc to be dropped from nifty 50 from march 31...	[ioc, to, be, dropped, from, nifty, 50, from, ...	[ioc, dropped, nifty, 50, march, 31, ;, apollo...	[ioc, drop, nifti, 50, march, 31, apollo, hosp...
4	FEB 25, 2022	9:23 PM	21:23	cbi says nse himalayan yogi none other than an...	MARKET	[cbi says nse himalayan yogi none other than a...	[cbi, says, nse, himalayan, yogi, none, other,...	[cbi, says, nse, himalayan, yogi, none, anand,...	[cbi, say, nse, himalayan, yogi, none, anand, ...

Lemmatization

```
[18] # LEMMATIZATION

     import nltk
     nltk.download('wordnet')
     final_train_lemma_word = []
     wordnet_lemmatizer = WordNetLemmatizer()
     for line in train_data['stop_words_removed']:
       lemma_word=[];
       for w in line:
           word1 = wordnet_lemmatizer.lemmatize(w, pos = "n")
           word2 = wordnet_lemmatizer.lemmatize(word1, pos = "v")
           word3 = wordnet_lemmatizer.lemmatize(word2, pos = ("a"))
           lemma_word.append(word1)
       lemma_word= [word for word in lemma_word if word.isalnum()]
       final_train_lemma_word.append(lemma_word)

     print(final_train_lemma_word[0:5])
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
[['war', 'live', 'update', 'nato', 'leader', 'meet', 'reassure', 'ally', 'near', 'russia', 'ukraine', 'air', 'india', 'operate', '3', 'flight', 'bucharest',
```

```
[19] train_data['lemmatized_words']=final_train_lemma_word
     train_data.head()
```

[19]

	date	time_12hr	time_24hr	text	category	sent_token	word_token	stop_words_removed	stemmed_words	lemmatized_words
0	FEB 26, 2022	12:13 AM	00:13	russia-ukraine war live updates: nato leaders ...	WORLD	[russia-ukraine war live updates: nato leaders...	[russia-ukraine, war, live, updates, :, nato, ...	[russia-ukraine, war, live, updates, :, nato, ...	[war, live, updat, nato, leader, meet, reassur...	[war, live, update, nato, leader, meet, reassu...
1	FEB 25, 2022	10:25 PM	22:25	cnbc-tv18 classroom: what should be your optio...	MARKET	[cnbc-tv18 classroom: what should be your opti...	[cnbc-tv18, classroom, :, what, should, be, yo...	[cnbc-tv18, classroom, :, options, trading, ex...	[classroom, option, trade, exit, strategi, exp...	[classroom, option, trading, exit, strategy, e...
2	FEB 25, 2022	9:47 PM	21:47	ukraine-russia conflict: from sunflower oil to...	MARKET	[ukraine-russia conflict: from sunflower oil t...	[ukraine-russia, conflict, :, from, sunflower,...	[ukraine-russia, conflict, :, sunflower, oil, ...	[conflict, sunflow, oil, beer, consum, good, c...	[conflict, sunflower, oil, beer, consumer, goo...
3	FEB 25, 2022	9:41 PM	21:41	ioc to be dropped from nifty 50 from march 31;...	MARKET	[ioc to be dropped from nifty 50 from march 31...	[ioc, to, be, dropped, from, nifty, 50, from, ...	[ioc, dropped, nifty, 50, march, 31, :, apollo...	[ioc, drop, nifti, 50, march, 31, apollo, hosp...	[ioc, dropped, nifty, 50, march, 31, apollo, h...
4	FEB 25, 2022	9:23 PM	21:23	cbi says nse himalayan yogi none other than an...	MARKET	[cbi says nse himalayan yogi none other than a...	[cbi, says, nse, himalayan, yogi, none, other,...	[cbi, says, nse, himalayan, yogi, none, anand,...	[cbi, say, nse, himalayan, yogi, none, anand, ...	[cbi, say, nse, himalayan, yogi, none, anand, ...

Building Inverted Index

```
# GENERATING INVERTED INDEX
def generate_inverted_index(data: list):
    inv_idx_dict = {}
    for index, doc_text in enumerate(data):
        for word in doc_text:
            if word not in inv_idx_dict.keys():
                inv_idx_dict[word] = [index]
            elif index not in inv_idx_dict[word]:
                inv_idx_dict[word].append(index)
    return inv_idx_dict

final_train=generate_inverted_index(final_train_stem_list)

j=0
for i in final_train:
  print(i,":",final_train[i])
  if j==20:
    break;
  j=j+1
```

```
war : [0, 18, 47, 54, 579, 3141]
live : [0, 2103, 3765, 4736]
updat : [0, 367, 1058, 1610, 1679, 1736, 1774, 1873, 2465, 2803, 2813, 3059, 3575, 3629, 3670, 4183, 4276]
nato : [0, 54, 322]
leader : [0, 1918, 3857, 4561, 4986]
meet : [0, 217, 221, 262, 429, 492, 699, 806, 837, 894, 960, 1057, 1112, 1118, 1670, 1762, 1941, 2332, 2410, 2500, 2576, 2735, 2828, 2958, 3009, 3302, 3438,
reassur : [0]
alli : [0, 1727, 2889]
near : [0, 39, 137, 153, 160, 294, 303, 318, 398, 568, 581, 593, 604, 637, 649, 816, 864, 914, 976, 1010, 1249, 1286, 1312, 1675, 1958, 2008, 2093, 2130, 21
russia : [0, 48, 51, 60, 62, 65, 68, 102, 183, 310, 322, 355, 369, 395, 2692]
ukrain : [0, 6, 7, 11, 48, 51, 60, 62, 63, 65, 68, 78, 86, 99, 102, 123, 148, 160, 175, 177, 183, 215, 229, 262, 303, 326, 355, 362, 369, 395, 406, 423, 461,
air : [0, 7, 988, 989, 1003, 1640, 2258, 2599, 3835, 4484, 4724, 4845]
india : [0, 7, 11, 53, 59, 73, 94, 111, 228, 242, 255, 259, 267, 273, 301, 324, 326, 328, 348, 392, 427, 429, 441, 442, 455, 498, 576, 612, 613, 657, 684, 7
oper : [0, 51, 78, 1315, 1406, 1575, 1733, 2782, 3071, 3085, 3261, 3270, 3283, 3377, 3596, 3834, 4485, 4637]
3 : [0, 26, 81, 83, 219, 254, 343, 378, 428, 463, 466, 487, 590, 592, 614, 616, 627, 665, 682, 726, 762, 771, 774, 779, 800, 801, 814, 826, 847, 867, 881, 9
flight : [0, 7, 991, 1999]
bucharest : [0]
evacu : [0, 7]
```

✓ 3s completed at 5:11 PM

```
evacu : [0, 7]
indian : [0, 7, 45, 46, 47, 127, 148, 164, 232, 251, 285, 308, 442, 548, 663, 707, 886, 934, 938, 1006, 1090, 1286, 1546, 1548, 15
classroom : [1, 469, 517]
option : [1, 469, 473, 517, 1670, 1758, 2213, 2701, 2783, 3127, 3554, 3793, 4064, 4516, 4638]
```

```
[22] # SORTING THE INDEX BASED ON TERMS
     final_train1=sorted(final_train.items())
     final_train1
```

```
     2988,
     3006,
     3387,
     3589,
     3969,
     4034,
     4038,
     4058,
     4113]),
    ('circl', [2233]),
    ('circuit',
     [119,
      1086,
      1115,
      1192,
      1362,
      1488,
      1620,
      1711,
      1848,
      2662,
      2908,
      2956,
      3055,
      3251,
      3927,
      4260,
      4322]),
    ('circul', [3073, 3684]),
    ('circular', [3125, 3646]),
    ('cite', [3399, 4244]),
```

Adding the module for timing the query response

```
!pip install ipython-autotime
%load_ext autotime
```

```
Collecting ipython-autotime
  Downloading ipython_autotime-0.3.1-py2.py3-none-any.whl (6.8 kB)
Requirement already satisfied: ipython in /usr/local/lib/python3.7/dist-packages (from ipython-autotime) (5.5.0)
Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (4.8.0)
Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (2.6.1)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (57.4.0)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (0.7.5)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (5.1.1)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (4.4.2)
Requirement already satisfied: prompt-toolkit<2.0.0,>=1.0.4 in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (1.0.18)
Requirement already satisfied: simplegeneric>0.8 in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (0.8.1)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from prompt-toolkit<2.0.0,>=1.0.4->ipython->ipython-autotime) (1.15.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packages (from prompt-toolkit<2.0.0,>=1.0.4->ipython->ipython-autotime) (0.2.5)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.7/dist-packages (from pexpect->ipython->ipython-autotime) (0.7.0)
Installing collected packages: ipython-autotime
Successfully installed ipython-autotime-0.3.1
time: 229 µs (started: 2022-03-27 11:38:06 +00:00)
```

Building Positional Index

```
# GENERATING POSITIONAL INDEX
pos_index = {}
file_map = {}
def generate_positional_index(data:list):
  fileno=0
  lineno=-1
  for line in data:
    lineno+=1;
    for pos, term in enumerate(line):
      if term in pos_index:
          pos_index[term][0] = pos_index[term][0] + 1
          if fileno in pos_index[term][1]:
            pos_index[term][1][lineno].append(pos)
          else:
            pos_index[term][1][lineno] = [pos]
      else:
          pos_index[term] = []
          pos_index[term].append(1)
          pos_index[term].append({})
          pos_index[term][1][lineno] = [pos]
      fileno += 1
  return pos_index

final=generate_positional_index(final_train_stem_list)
count=0
for i in final:
  count=count+1;
  if count<=20:
    print(i,final[i])
  else:
    break;
```

```
war [6, {0: [0], 18: [0], 47: [0], 54: [5], 579: [2], 3141: [3]}]
live [4, {0: [1], 2103: [4], 3765: [7], 4736: [10]}]
updat [17, {0: [2], 367: [3], 1058: [8], 1610: [10], 1679: [9], 1736: [3], 1774: [8], 1873: [3], 2465: [4], 2803: [3], 2813: [1], 3059: [3], 3575: [1], 3629:
nato [3, {0: [3], 54: [3], 322: [7]}]
```

```
[34] nato [3, {0: [3], 54: [3], 322: [7]}]
     leader [5, {0: [4], 1918: [3], 3857: [7], 4561: [9], 4986: [4]}]
     meet [38, {0: [5], 217: [10], 221: [4], 262: [6], 429: [7], 492: [4], 699: [3], 806: [9], 837: [7], 894: [3], 960: [10], 1057: [6], 1112: [1], 1118: [13], 1
     reassur [1, {0: [6]}]
     alli [3, {0: [7], 1727: [1], 2889: [6]}]
     near [56, {0: [8], 39: [4], 137: [2], 153: [2], 160: [2], 294: [3], 303: [2], 318: [7], 398: [6], 568: [6], 581: [9], 593: [10], 604: [2], 637: [7], 649: [3
     russia [15, {0: [9], 48: [7], 51: [4], 60: [8], 62: [6], 65: [6], 68: [7], 102: [3], 183: [6], 310: [2], 322: [9], 355: [5], 369: [3], 395: [9], 2692: [8]}]
     ukrain [38, {0: [10], 6: [4], 7: [9], 11: [7], 48: [9], 51: [7], 60: [10], 62: [8], 63: [6], 65: [8], 68: [9], 78: [8], 86: [6], 99: [3], 102: [4], 123: [0],
     air [12, {0: [11], 7: [3], 988: [3], 989: [8], 1003: [4], 1640: [9], 2258: [7], 2599: [8], 3835: [11], 4484: [10], 4724: [7], 4845: [10]}]
     india [250, {0: [12], 7: [4], 11: [9], 53: [0], 59: [3], 73: [9], 94: [6], 111: [2], 228: [5], 242: [4], 255: [0], 259: [3], 267: [4], 273: [4], 301: [4], 3
     oper [19, {0: [13], 51: [6], 78: [7], 1315: [6], 1406: [8], 1575: [1], 1733: [6], 2782: [10], 3071: [2], 3085: [3], 3261: [3], 3270: [9], 3283: [3], 3377: [
     3 [150, {0: [14], 26: [8], 81: [7], 83: [5], 219: [3], 254: [9], 343: [6], 378: [5], 428: [5], 463: [2], 466: [7], 487: [10], 590: [10], 592: [1], 614: [12],
     flight [4, {0: [15], 7: [5], 991: [12], 1999: [6]}]
     bucharest [1, {0: [16]}]
     evacu [2, {0: [17], 7: [6]}]
     indian [95, {0: [18], 7: [7], 45: [3], 46: [3], 47: [1], 127: [0], 148: [0], 164: [3], 232: [4], 251: [11], 285: [1], 308: [8], 442: [7], 548: [7], 663: [4],
     classroom [3, {1: [0], 469: [0], 517: [0]}]
     time: 95.3 ms (started: 2022-03-27 11:40:36 +00:00)
```

```
# SORTING THE POSITIONAL INDEX BASED ON TERMS
final_train1=sorted(final.items())
final_train1
```

```
     3387: [1],
     3589: [1],
     3969: [0],
     4034: [8],
     4038: [9],
     4058: [0],
     4113: [8]}]),
 ('circl', [1, {2233: [8]}]),
 ('circuit',
  [18,
   {119: [5],
    1086: [5],
    1115: [7],
    1192: [6],
    1362: [7],
    1488: [6],
    1620: [5],
```

Performing Boolean Queries

```
[24] # Boolean Query
     # AND
     def and_query(l1, l2):
         p1 = 0
         p2 = 0
         result = list()
         while p1 < len(l1) and p2 < len(l2):
             if l1[p1] == l2[p2]:
                 result.append(l1[p1])
                 p1 += 1
                 p2 += 1
             elif l1[p1] > l2[p2]:
                 p2 += 1
             else:
                 p1 += 1
         return result
```

```
time: 10.8 ms (started: 2022-03-27 11:38:08 +00:00)
```

```
[25] def or_query(l1,l2):
       result=list();
       p1=0
       p2=0
       while p1 < len(l1) and p2 < len(l2):
         if l1[p1] == l2[p2]:
           result.append(l1[p1])
           p1 += 1
           p2 += 1
         elif l1[p1] > l2[p2]:
           result.append(l2[p2])
           p2 += 1
         else:
           result.append(l1[p1])
           p1 += 1
       while(p1 < len(l1)):
             result.append(l1[p1])
             p1 += 1
       while p2 < len(l2):
             result.append(l2[p2])
             p2 += 1
       return result
```

```
time: 13.9 ms (started: 2022-03-27 11:38:12 +00:00)
```

```
# PERFORMING THE BOOLEAN QUERY
print("Enter the first input word : ")
input1=input()
print("Enter the second input word : ")
input2=input()
```

```
[29] Enter the first input word :
     india
     Enter the second input word :
     russia
     time: 9.51 s (started: 2022-03-27 11:39:45 +00:00)

[30] l1=final_train[input1]
     l2=final_train[input2]
     print("posting list for",input1 ,l1)
     print("posting list for",input2,l2)
     print("Resultant list: ",and_query(l1,l2))

     posting list for india [0, 7, 11, 53, 59, 73, 94, 111, 228, 242, 255, 259, 267, 273, 301, 324, 326, 328, 348, 392, 427, 429, 441, 442, 455, 498, 576, 612, 6
     posting list for russia [0, 48, 51, 60, 62, 65, 68, 102, 183, 310, 322, 355, 369, 395, 2692]
     Resultant list:  [0, 2692]
     time: 7.71 ms (started: 2022-03-27 11:39:57 +00:00)

[31] print("Resultant list: ",or_query(l1,l2))
     print("Length of posting list for",input1 ,len(l1))
     print("Length of posting list for",input2,len(l2))
     print("Length of and list: ",len(and_query(l1,l2)))
     print("Resultant list :",or_query(l1,l2))
     print("Length of the OR list:",len(or_query(l1,l2)))

     Resultant list:  [0, 7, 11, 48, 51, 53, 59, 60, 62, 65, 68, 73, 94, 102, 111, 183, 228, 242, 255, 259, 267, 273, 301, 310, 322, 324, 326, 328, 348, 355, 369,
     Length of posting list for india 248
     Length of posting list for russia 15
     Length of and list:  2
     Resultant list : [0, 7, 11, 48, 51, 53, 59, 60, 62, 65, 68, 73, 94, 102, 111, 183, 228, 242, 255, 259, 267, 273, 301, 310, 322, 324, 326, 328, 348, 355, 369,
     Length of the OR list: 261
     time: 6.7 ms (started: 2022-03-27 11:40:04 +00:00)
```

```
[32] print("Enter the third input word : ")
     input3=input()
     print("Enter the fourth input word : ")
     input4=input()
     l3=final_train[input3]
     l4=final_train[input4]
     resultant=or_query(or_query(and_query(l1,l4),l3),l2)
     print("Result:",resultant)
     print("Result length:",len(resultant))

     Enter the third input word :
     meet
     Enter the fourth input word :
     option
     Result: [0, 48, 51, 60, 62, 65, 68, 102, 183, 217, 221, 262, 310, 322, 355, 369, 395, 429, 492, 699, 806, 837, 894, 960, 1057, 1112, 1118, 1670, 1762, 1941,
     Result length: 52
     time: 6.63 s (started: 2022-03-27 11:40:16 +00:00)

[33] resultant=and_query(or_query(l1,l3),or_query(l2,l4))
     print("Result:",resultant)
     print("Result length:",len(resultant))

     Result: [0, 1670, 2692]
     Result length: 3
     time: 6.42 ms (started: 2022-03-27 11:40:26 +00:00)
```

Performing Phrase Query on Inverted Index

```
     nato [3, {0: [3], 54: [3], 322: [7]}]
[34] leader [5, {0: [4], 1918: [3], 3857: [7], 4561: [9], 4986: [4]}]
     meet [38, {0: [5], 217: [10], 221: [4], 262: [6], 429: [7], 492: [4], 699: [3], 806: [9], 837: [7], 894: [3], 960: [10], 1057: [6], 1112: [1], 1118: [13], 1
     reassur [1, {0: [6]}]
     alli [3, {0: [7], 1727: [1], 2889: [6]}]
     near [56, {0: [8], 39: [4], 137: [2], 153: [2], 160: [2], 294: [3], 303: [2], 318: [7], 398: [6], 568: [6], 581: [9], 593: [10], 604: [2], 637: [7], 649: [3
     russia [15, {0: [9], 48: [7], 51: [4], 60: [8], 62: [6], 65: [6], 68: [7], 102: [3], 183: [6], 310: [2], 322: [9], 355: [5], 369: [3], 395: [9], 2692: [8]}]
     ukrain [38, {0: [10], 6: [4], 7: [9], 11: [7], 48: [9], 51: [7], 60: [10], 62: [8], 63: [6], 65: [8], 68: [9], 78: [8], 86: [6], 99: [3], 102: [4], 123: [0],
     air [12, {0: [11], 7: [3], 988: [3], 989: [8], 1003: [4], 1640: [9], 2258: [7], 2599: [8], 3835: [11], 4484: [10], 4724: [7], 4845: [10]}]
     india [250, {0: [12], 7: [4], 11: [9], 53: [0], 59: [3], 73: [9], 94: [6], 111: [2], 228: [5], 242: [4], 255: [0], 259: [3], 267: [4], 273: [4], 301: [4], 3
     oper [19, {0: [13], 51: [6], 78: [7], 1315: [6], 1406: [8], 1575: [1], 1733: [6], 2782: [10], 3071: [2], 3085: [3], 3261: [3], 3270: [9], 3283: [3], 3377: [
     3 [150, {0: [14], 26: [8], 81: [7], 83: [5], 219: [3], 254: [9], 343: [6], 378: [5], 428: [5], 463: [2], 466: [7], 487: [10], 590: [10], 592: [1], 614: [12],
     flight [4, {0: [15], 7: [5], 991: [12], 1999: [6]}]
     bucharest [1, {0: [16]}]
     evacu [2, {0: [17], 7: [6]}]
     indian [95, {0: [18], 7: [7], 45: [3], 46: [3], 47: [1], 127: [0], 148: [0], 164: [3], 232: [4], 251: [11], 285: [1], 308: [8], 442: [7], 548: [7], 663: [4],
     classroom [3, {1: [0], 469: [0], 517: [0]}]
     time: 95.3 ms (started: 2022-03-27 11:40:36 +00:00)

     # SORTING THE POSITIONAL INDEX BASED ON TERMS
     final_train1=sorted(final.items())
     final_train1

       3387: [1],
       3589: [1],
       3969: [0],
       4034: [8],
       4038: [9],
       4058: [0],
       4113: [8]}]),
      ('circl', [1, {2233: [8]}]),
      ('circuit',
       [18,
        {119: [5],
         1086: [5],
         1115: [7],
         1192: [6],
         1362: [7],
         1488: [6],
         1620: [5],
```

Performing Phrase Query on Positional Index

```
[38] # Phrase query on positional index :
     def fetch_list(d):
         l=list();
         d1=d[1];
         for i in d1:
           l.append(i)
         return l;
     def post_phrase_query(phr):
       query=phr.split()
       for i in range(0,len(query)-1,2):
         l1=fetch_list(final[query[i]])
         l2=fetch_list(final[query[i+1]])
         result=and_query(l1,l2)
       print(result)
```

```
time: 8.94 ms (started: 2022-03-27 11:41:11 +00:00)
```

```
print("Enter your query")
q=input()
post_phrase_query(q)
```

```
Enter your query
air india russia
[0, 7, 988, 1003, 1640, 4724, 4845]
time: 11.9 s (started: 2022-03-27 11:41:14 +00:00)
```

```
[40] print("Enter your query")
     q=input()
     post_phrase_query(q)
```

```
Enter your query
india indian
[0, 7, 442, 2251]
time: 4.42 s (started: 2022-03-27 11:41:30 +00:00)
```