UE19CS332 : Algorithms for Web and Information Retrieval
ASSIGNMENT

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 – 1 : *FINANCIAL SENTIMENT ANALYSIS CORPUS*

DATASET LOCATION : DATASET > DATA.CSV

NOTEBOOK NAME : A3_P1.ipynb

STEPS :

Importing all libraries

```
[1] # Assignment - 1

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

    # Financial Sentiment 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
```

Case folding

```
# CASE FOLDING

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

	Sentence	Sentiment
0	the geosolutions technology will leverage bene...	positive
1	$esi on lows, down $1.50 to $2.50 bk a real po...	negative
2	for the last quarter of 2010 , componenta 's n...	positive
3	according to the finnish-russian chamber of co...	neutral
4	the swedish buyout firm has sold its remaining...	neutral

+ Code + Text

```
[3] train_data.shape
```

```
(5842, 2)
```

Renaming Columns

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

[5]  train_data.columns

     Index(['text', 'Sentiment'], dtype='object')
```

Sentence Tokenization

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

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

     train_data.head()
```

	text	Sentiment	sent_token
0	the geosolutions technology will leverage bene...	positive	[the geosolutions technology will leverage ben...
1	$esi on lows, down $1.50 to $2.50 bk a real po...	negative	[$esi on lows, down $1.50 to $2.50 bk a real p...
2	for the last quarter of 2010 , componenta 's n...	positive	[for the last quarter of 2010 , componenta 's ...
3	according to the finnish-russian chamber of co...	neutral	[according to the finnish-russian chamber of c...
4	the swedish buyout firm has sold its remaining...	neutral	[the swedish buyout firm has sold its remainin...

Word Tokenization

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

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

[11] train_data.head()
```

	text	Sentiment	sent_token	word_token
0	the geosolutions technology will leverage bene...	positive	[the geosolutions technology will leverage ben...	[the, geosolutions, technology, will, leverage...
1	$esi on lows, down $1.50 to $2.50 bk a real po...	negative	[$esi on lows, down $1.50 to $2.50 bk a real p...	[$, esi, on, lows, ,, down, $, 1.50, to, $, 2....
2	for the last quarter of 2010 , componenta 's n...	positive	[for the last quarter of 2010 , componenta 's ...	[for, the, last, quarter, of, 2010, ,, compone...
3	according to the finnish-russian chamber of co...	neutral	[according to the finnish-russian chamber of c...	[according, to, the, finnish-russian, chamber,...
4	the swedish buyout firm has sold its remaining...	neutral	[the swedish buyout firm has sold its remainin...	[the, swedish, buyout, firm, has, sold, its, r...

Stop Words Removal

```
[12] # 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.

[13] 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

[14] train_data.head()
```

	text	Sentiment	sent_token	word_token	stop_words_removed
0	the geosolutions technology will leverage bene...	positive	[the geosolutions technology will leverage ben...	[the, geosolutions, technology, will, leverage...	[geosolutions, technology, leverage, benefon, ...
1	$esi on lows, down $1.50 to $2.50 bk a real po...	negative	[$esi on lows, down $1.50 to $2.50 bk a real p...	[$, esi, on, lows, ,, down, $, 1.50, to, $, 2....	[$, esi, lows, ,, $, 1.50, $, 2.50, bk, real, ...
2	for the last quarter of 2010 , componenta 's n...	positive	[for the last quarter of 2010 , componenta 's ...	[for, the, last, quarter, of, 2010, ,, compone...	[last, quarter, 2010, ,, componenta, 's, net, ...
3	according to the finnish-russian chamber of co...	neutral	[according to the finnish-russian chamber of c...	[according, to, the, finnish-russian, chamber,...	[according, finnish-russian, chamber, commerce...
4	the swedish buyout firm has sold its remaining...	neutral	[the swedish buyout firm has sold its remainin...	[the, swedish, buyout, firm, has, sold, its, r...	[swedish, buyout, firm, sold, remaining, 22.4,...

3s completed at 4:27 PM

Stemming

```
[15] # 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])

     [['geosolut', 'technolog', 'leverag', 'benefon', 'gp', 'solut', 'provid', 'locat', 'base', 'search', 'technolog', 'commun', 'platform', 'locat', 'relev', 'mu

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

	text	Sentiment	sent_token	word_token	stop_words_removed	stemmed_words
0	the geosolutions technology will leverage bene...	positive	[the geosolutions technology will leverage ben...	[the, geosolutions, technology, will, leverage...	[geosolutions, technology, leverage, benefon, ...	[geosolut, technolog, leverag, benefon, gp, so...
1	$esi on lows, down $1.50 to $2.50 bk a real po...	negative	[$esi on lows, down $1.50 to $2.50 bk a real p...	[$, esi, on, lows, ,, down, $, 1.50, to, $, 2....	[$, esi, lows, ,, $, 1.50, $, 2.50, bk, real, ...	[esi, low, bk, real, possibl]
2	for the last quarter of 2010 , componenta 's n...	positive	[for the last quarter of 2010 , componenta 's ...	[for, the, last, quarter, of, 2010, ,, compone...	[last, quarter, 2010, ,, componenta, 's, net, ...	[last, quarter, 2010, componenta, net, sale, d...
3	according to the finnish-russian chamber of co...	neutral	[according to the finnish-russian chamber of c...	[according, to, the, finnish-russian, chamber,...	[according, finnish-russian, chamber, commerce...	[accord, chamber, commerc, major, construct, c...
4	the swedish buyout firm has sold its remaining...	neutral	[the swedish buyout firm has sold its remainin...	[the, swedish, buyout, firm, has, sold, its, r...	[swedish, buyout, firm, sold, remaining, 22.4,...	[swedish, buyout, firm, sold, remain, percent,...

Lemmatization

```
# 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.
[['geosolutions', 'technology', 'leverage', 'benefon', 'gps', 'solution', 'providing', 'location', 'based', 'search', 'technology', 'community', 'platform',
```

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

	text	Sentiment	sent_token	word_token	stop_words_removed	stemmed_words	lemmatized_words
0	the geosolutions technology will leverage bene...	positive	[the geosolutions technology will leverage ben...	[the, geosolutions, technology, will, leverage...	[geosolutions, technology, leverage, benefon, ...	[geosolut, technolog, leverag, benefon, gp, so...	[geosolutions, technology, leverage, benefon, ...
1	$esi on lows, down $1.50 to $2.50 bk a real po...	negative	[$esi on lows, down $1.50 to $2.50 bk a real p...	[$, esi, on, lows, ,, down, $, 1.50, to, $, 2....	[$, esi, lows, ,, $, 1.50, $, 2.50, bk, real, ...	[esi, low, bk, real, possibl]	[esi, low, bk, real, possibility]
2	for the last quarter of 2010 , componenta 's n...	positive	[for the last quarter of 2010 , componenta 's ...	[for, the, last, quarter, of, 2010, ,, compone...	[last, quarter, 2010, ,, componenta, 's, net, ...	[last, quarter, 2010, componenta, net, sale, d...	[last, quarter, 2010, componenta, net, sale, d...
3	according to the finnish-russian chamber of co...	neutral	[according to the finnish-russian chamber of c...	[according, to, the, finnish-russian, chamber,...	[according, finnish-russian, chamber, commerce...	[accord, chamber, commerc, major, construct, c...	[according, chamber, commerce, major, construc...
4	the swedish buyout firm has sold its remaining...	neutral	[the swedish buyout firm has sold its remainin...	[the, swedish, buyout, firm, has, sold, its, r...	[swedish, buyout, firm, sold, remaining, 22.4,...	[swedish, buyout, firm, sold, remain, percent,...	[swedish, buyout, firm, sold, remaining, perce...

Building Inverted Index

```
# GENERATING INVERTED
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

final_train1=sorted(final_train.items())
```

```
geosolut : [0, 412]
technolog : [0, 59, 109, 137, 300, 412, 427, 436, 473, 558, 657, 672, 679, 682, 712, 720, 757, 758, 823, 849, 941, 990, 1018, 1105, 1124, 1215, 1329, 1462, 
leverag : [0, 858, 3958]
benefon : [0, 300, 1053, 1312, 3808, 4158, 4993, 5604]
gp : [0, 300, 412, 3967, 4158]
solut : [0, 62, 73, 82, 85, 105, 171, 199, 201, 214, 319, 328, 341, 348, 366, 459, 496, 527, 530, 546, 607, 629, 642, 679, 687, 706, 739, 740, 773, 787, 817,
provid : [0, 73, 82, 99, 135, 180, 191, 287, 321, 327, 357, 408, 412, 420, 436, 459, 506, 546, 607, 679, 687, 773, 820, 821, 832, 953, 970, 985, 986, 1104, 
locat : [0, 68, 187, 300, 533, 824, 983, 1454, 1977, 2038, 2137, 2164, 2230, 2284, 2807, 3006, 3131, 3153, 3248, 3393, 3605, 3625, 3680, 3808, 3828, 3966, 4
base : [0, 22, 198, 199, 253, 348, 352, 354, 391, 443, 524, 629, 777, 918, 981, 990, 1021, 1079, 1101, 1123, 1131, 1140, 1209, 1490, 1515, 1568, 1638, 1686,
search : [0, 2573, 3812, 4623, 4642, 5507]
commun : [0, 7, 135, 254, 287, 305, 390, 413, 427, 506, 608, 648, 696, 740, 815, 919, 930, 986, 1051, 1160, 1434, 1620, 1696, 1763, 1859, 1888, 2076, 2129, 
platform : [0, 1088, 1568, 2230, 2559, 3645, 3877, 4069, 4082, 4158, 4186, 4276, 4905]
relev : [0, 3421, 4025, 4549, 5528]
multimedia : [0, 277, 4263]
```

```
content : [0, 390, 527, 1078, 1606, 1790, 1843, 1935, 2146, 2545, 2553, 2784, 3260, 3393, 3736, 3961, 3986, 4173, 4412, 4534, 4670, 4989, 5156, 5566]
new : [0, 14, 82, 126, 144, 253, 272, 276, 287, 310, 339, 353, 354, 375, 395, 406, 412, 415, 434, 443, 464, 495, 508, 529, 582, 592, 596, 613, 639, 642, 644,
power : [0, 117, 606, 612, 715, 887, 1420, 1486, 1638, 1722, 1746, 1795, 1827, 2236, 2298, 2379, 2415, 2603, 2652, 2691, 2735, 2828, 2891, 2903, 3144, 3336,
commerci : [0, 82, 142, 300, 327, 363, 1138, 1431, 1704, 1832, 1842, 2015, 2040, 2059, 2096, 2385, 2753, 2825, 3192, 3197, 3230, 3834, 4017, 4674, 4864, 490
model : [0, 40, 167, 253, 297, 336, 396, 672, 786, 1044, 1128, 1271, 1291, 1298, 1616, 1649, 1760, 1885, 2060, 2311, 2647, 2691, 3007, 3038, 3145, 3258, 327
esi : [1]
low : [1, 173, 310, 382, 383, 416, 550, 600, 757, 769, 1029, 1107, 1462, 1465, 1552, 1602, 1843, 1961, 2066, 2357, 2398, 2425, 2449, 2614, 3200, 3289, 3303,
time: 363 ms (started: 2022-03-27 10:53:38 +00:00)
```

Sorting the index based on terms

```
# SORTING INDEX BASED ON TERMS
final_train1
```

```
   4847,
   5016,
   5136,
   5147,
   5249,
   5271,
   5329,
   5420,
   5545,
   5772]),
 ('backburn', [5164]),
 ('backdrop', [4464, 4773]),
 ('backhaul', [1380]),
 ('backlog', [307, 4446, 4753]),
 ('backup', [3393, 4873, 5047]),
 ('bad', [777, 1002, 1098, 2432, 2904, 3533, 4301, 5802]),
 ('badli', [200, 2726]),
 ('bae', [345, 542, 2514, 2916]),
 ('baer', [521]),
 ('bag', [3574, 4150, 4282, 4531, 5217, 5586]),
 ('bagdona', [5389]),
 ('bahia', [849, 1404]),
 ('bahr', [5752]),
 ('bahrain', [3996]),
 ('bailout', [5409]),
 ('baird', [1383, 3559]),
 ('bake', [3116]),
 ('baker', [5320]),
 ('bakeri', [1688, 5315]),
 ('bakman', [3082]),
 ('bakosch', [5527]),
 ('balanc',
  [554,
   759,
   1228,
   1708,
   1863,
   1910,
   2379,
   2580,
```

Adding the module for timing the query response

```
[24] !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: decorator in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (0.7.5)
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: setuptools>=18.5 in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (57.4.0)
Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (2.6.1)
Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (4.8.0)
Requirement already satisfied: simplegeneric>0.8 in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (0.8.1)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (5.1.1)
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: 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: 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: 169 µs (started: 2022-03-27 10:50:59 +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;
final1=sorted(final.items())
```

```
geosolut [2, {0: [0], 412: [0]}]
technolog [127, {0: [10], 59: [5], 109: [3], 137: [1], 300: [16], 412: [8], 427: [7], 436: [19], 473: [2], 558: [18], 657: [7], 672: [21], 679: [8], 682: [1
leverag [3, {0: [2], 858: [13], 3958: [6]}]
benefon [9, {0: [3], 300: [1], 1053: [4], 1312: [3], 3808: [6], 4158: [0], 4993: [0], 5604: [20]}]
gp [6, {0: [4], 300: [6], 412: [7], 3967: [0], 4158: [10]}]
solut [155, {0: [5], 62: [14], 73: [1], 82: [4], 85: [3], 105: [8], 171: [0], 199: [2], 201: [6], 214: [4], 319: [8], 328: [11], 341: [3], 348: [6], 366: [3
provid [142, {0: [6], 73: [2], 82: [5], 99: [1], 135: [8], 180: [2], 191: [8], 287: [4], 321: [0], 327: [3], 357: [3], 408: [3], 412: [2], 420: [7], 436: [1
locat [40, {0: [13], 68: [8], 187: [3], 300: [15], 533: [3], 824: [13], 983: [0], 1454: [10], 1977: [16], 2038: [5], 2137: [1], 2164: [6], 2230: [8], 2284:
base [88, {0: [8], 22: [1], 198: [3], 199: [3], 253: [7], 348: [3], 352: [2], 354: [3], 391: [9], 443: [14], 524: [2], 629: [23], 777: [6], 918: [22], 981:
search [6, {0: [9], 2573: [5], 3812: [2], 4623: [2], 4642: [3], 5507: [4]}]
commun [78, {0: [11], 7: [1], 135: [6], 254: [8], 287: [1], 305: [3], 390: [7], 413: [15], 427: [4], 506: [3], 608: [11], 648: [13], 696: [4], 740: [8], 815
platform [14, {0: [12], 1088: [0], 1568: [6], 2230: [5], 2559: [8], 3645: [4], 3877: [5], 4069: [11], 4082: [10], 4158: [9], 4186: [21], 4276: [14], 4905: [
relev [6, {0: [14], 3421: [11], 4025: [0], 4549: [9], 5528: [6]}]
multimedia [3, {0: [15], 277: [2], 4263: [1]}]
content [25, {0: [16], 390: [6], 527: [12], 1078: [13], 1606: [5], 1790: [5], 1843: [4], 1935: [2], 2146: [15], 2545: [2], 2553: [3], 2784: [9], 3260: [24],
new [274, {0: [17], 14: [6], 82: [7], 126: [2], 144: [3], 253: [0], 272: [7], 276: [2], 287: [11], 310: [2], 339: [0], 353: [0], 354: [12], 375: [6], 395: [
power [49, {0: [18], 117: [9], 606: [1], 612: [10], 715: [5], 887: [7], 1420: [10], 1486: [3], 1638: [9], 1722: [9], 1746: [7], 1795: [6], 1827: [1], 2236:
commerci [32, {0: [19], 82: [18], 142: [3], 300: [18], 327: [14], 363: [14], 1138: [2], 1431: [1], 1704: [9], 1832: [13], 1842: [9], 2015: [8], 2040: [6], 2
model [45, {0: [20], 40: [6], 167: [2], 253: [11], 297: [1], 336: [6], 396: [24], 672: [9], 786: [5], 1044: [7], 1128: [2], 1271: [9], 1291: [2], 1298: [2],
esi [1, {1: [0]}]
time: 407 ms (started: 2022-03-27 10:55:25 +00:00)
```

```
# SORTING BASED ON THE TERMS
final1
```

Performing Boolean Queries

```
[25] # 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: 5.6 ms (started: 2022-03-27 10:51:03 +00:00)
```

```
[26] 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: 20.9 ms (started: 2022-03-27 10:51:07 +00:00)
```

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

```
Enter the first input word :
new
Enter the second input word :
content
time: 5.14 s (started: 2022-03-27 10:52:15 +00:00)
```

```
[33] 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 new [0, 14, 82, 126, 144, 253, 272, 276, 287, 310, 339, 353, 354, 375, 395, 406, 412, 415, 434, 443, 464, 495, 508, 529, 582, 592, 596, 613,
posting list for content [0, 390, 527, 1078, 1606, 1790, 1843, 1935, 2146, 2545, 2553, 2784, 3260, 3393, 3736, 3961, 3986, 4173, 4412, 4534, 4670, 4989, 515(
Resultant list:  [0, 2146, 4534, 5156]
time: 5.62 ms (started: 2022-03-27 10:53:51 +00:00)
```

```
[34] 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, 14, 82, 126, 144, 253, 272, 276, 287, 310, 339, 353, 354, 375, 390, 395, 406, 412, 415, 434, 443, 464, 495, 508, 527, 529, 582, 592, 59(
Length of posting list for new 261
Length of posting list for content 24
Length of and list:  4
Resultant list : [0, 14, 82, 126, 144, 253, 272, 276, 287, 310, 339, 353, 354, 375, 390, 395, 406, 412, 415, 434, 443, 464, 495, 508, 527, 529, 582, 592, 59(
Length of the OR list: 281
time: 10.9 ms (started: 2022-03-27 10:53:57 +00:00)
```

```
[35] 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 :
model
Enter the fourth input word :
multimedia
Result: [0, 40, 167, 253, 297, 336, 390, 396, 527, 672, 786, 1044, 1078, 1128, 1271, 1291, 1298, 1606, 1616, 1649, 1760, 1790, 1843, 1885, 1935, 2060, 2146,
Result length: 66
time: 51.8 s (started: 2022-03-27 10:53:59 +00:00)
```

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

```
Result: [0, 2146, 4534, 5156]
Result length: 4
time: 5.24 ms (started: 2022-03-27 10:54:54 +00:00)
```

Performing Phrase Query on Inverted Index

```
[39] # PHRASE QUERY on Inverted Index :
     def phrase_query(phr):
         query=phr.split();
         for i in range(0,len(query)-1,2):
           result=and_query(final_train[query[i]],final_train[query[i+1]])
         print(result)
     print("Enter your query")
     q=input()
     phrase_query(q)
```

```
Enter your query
new content
[0, 2146, 4534, 5156]
time: 7.26 s (started: 2022-03-27 10:56:21 +00:00)
```

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

```
Enter your query
new multimedia content
[0]
time: 9.21 s (started: 2022-03-27 10:56:29 +00:00)
```

Performing Phrase Query on Positional Index

```
[42] # 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: 6.46 ms (started: 2022-03-27 10:57:08 +00:00)
```

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

```
Enter your query
new multimedia content
[0]
time: 5.64 s (started: 2022-03-27 10:57:11 +00:00)
```

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

```
Enter your query
new content
[0, 2146, 4534, 5156]
time: 3.78 s (started: 2022-03-27 10:57:19 +00:00)
```

CORPUS – 2 : *TWITTER SENTIMENT ANALYSIS CORPUS*

DATASET LOCATION : DATASET > Twitter_sentiment.csv

NOTEBOOK NAME : A3_P2.ipynb

STEPS :

Importing all libraries

```
# Assignment - 2

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

# Twitter Sentiment 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

```
[3] # CASE FOLDING

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

	textID	text	sentiment_main	sentiment
0	p1000000000	i know i was listenin to bad habit earlier a...	empty	neutral
1	p1000000001	i should be sleep, but im not! thinking about ...	sadness	negative
2	2dfbe0b7fb	hmmm. http://www.djhero.com/ is down	worry	negative
3	6d846d7d50	i`m sorry at least it`s friday?	sadness	neutral
4	p1000000002	the storm is here and the electricity is gone	sadness	negative

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

```
(12454, 4)
```

```
train_data.columns
```

```
Index(['textID', 'text', 'sentiment_main', 'sentiment'], dtype='object')
```

Sentence Tokenization

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

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

[7] train_data.head()
```

	textID	text	sentiment_main	sentiment	sent_token
0	p1000000000	i know i was listenin to bad habit earlier a...	empty	neutral	[i know i was listenin to bad habit earlier ...
1	p1000000001	i should be sleep, but im not! thinking about ...	sadness	negative	[i should be sleep, but im not!, thinking abou...
2	2dfbe0b7fb	hmmm. http://www.djhero.com/ is down	worry	negative	[hmmm., http://www.djhero.com/ is down]
3	6d846d7d50	i`m sorry at least it`s friday?	sadness	neutral	[i`m sorry at least it`s friday?]
4	p1000000002	the storm is here and the electricity is gone	sadness	negative	[the storm is here and the electricity is gone]

Word Tokenization

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

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

[10] train_data.head()
```

	textID	text	sentiment_main	sentiment	sent_token	word_token
0	p1000000000	i know i was listenin to bad habit earlier a...	empty	neutral	[i know i was listenin to bad habit earlier ...	[i, know, i, was, listenin, to, bad, habit, ea...
1	p1000000001	i should be sleep, but im not! thinking about ...	sadness	negative	[i should be sleep, but im not!, thinking abou...	[i, should, be, sleep, ,, but, im, not, !, thi...
2	2dfbe0b7fb	hmmm. http://www.djhero.com/ is down	worry	negative	[hmmm., http://www.djhero.com/ is down]	[hmmm, ., http, :, //www.djhero.com/, is, down]
3	6d846d7d50	i`m sorry at least it`s friday?	sadness	neutral	[i`m sorry at least it`s friday?]	[i`m, sorry, at, least, it`s, friday, ?]
4	p1000000002	the storm is here and the electricity is gone	sadness	negative	[the storm is here and the electricity is gone]	[the, storm, is, here, and, the, electricity, ...

Stop Words Removal

```
[11] # 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.

[12] 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

[13] train_data.head()
```

	textID	text	sentiment_main	sentiment	sent_token	word_token	stop_words_removed
0	p1000000000	i know i was listenin to bad habit earlier a...	empty	neutral	[i know i was listenin to bad habit earlier ...	[i, know, i, was, listenin, to, bad, habit, ea...	[know, listenin, bad, habit, earlier, started,...
1	p1000000001	i should be sleep, but im not! thinking about ...	sadness	negative	[i should be sleep, but im not!, thinking abou...	[i, should, be, sleep, ,, but, im, not, !, thi...	[sleep, ,, im, !, thinking, old, friend, want,...
2	2dfbe0b7fb	hmmm. http://www.djhero.com/ is down	worry	negative	[hmmm., http://www.djhero.com/ is down]	[hmmm, ., http, :, //www.djhero.com/, is, down]	[hmmm, ., http, :, //www.djhero.com/]
3	6d846d7d50	i`m sorry at least it`s friday?	sadness	neutral	[i`m sorry at least it`s friday?]	[i`m, sorry, at, least, it`s, friday, ?]	[i`m, sorry, least, it`s, friday, ?]
4	p1000000002	the storm is here and the electricity is gone	sadness	negative	[the storm is here and the electricity is gone]	[the, storm, is, here, and, the, electricity, ...	[storm, electricity, gone]

Stemming

```
[18] # 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])

     [['know', 'listenin', 'bad', 'habit', 'earlier', 'start', 'freakin', 'part'], ['sleep', 'im', 'think', 'old', 'friend', 'want', 'marri', 'want', '2', 'scand

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

	textID	text	sentiment_main	sentiment	sent_token	word_token	stop_words_removed	stemmed_words
0	p1000000000	i know i was listenin to bad habit earlier a...	empty	neutral	[i know i was listenin to bad habit earlier ...	[i, know, i, was, listenin, to, bad, habit, ea...	[know, listenin, bad, habit, earlier, started,...	[know, listenin, bad, habit, earlier, start, f...
1	p1000000001	i should be sleep, but im not! thinking about ...	sadness	negative	[i should be sleep, but im not!, thinking abou...	[i, should, be, sleep, ,, but, im, not, !, thi...	[sleep, ,, im, !, thinking, old, friend, want,...	[sleep, im, think, old, friend, want, marri, w...
2	2dfbe0b7fb	hmmm. http://www.djhero.com/ is down	worry	negative	[hmmm., http://www.djhero.com/ is down]	[hmmm, ., http, :, //www.djhero.com/, is, down]	[hmmm, ., http, :, //www.djhero.com/]	[hmmm, http]
3	6d846d7d50	i`m sorry at least it`s friday?	sadness	neutral	[i`m sorry at least it`s friday?]	[i`m, sorry, at, least, it`s, friday, ?]	[i`m, sorry, least, it`s, friday, ?]	[sorri, least, friday]
4	p1000000002	the storm is here and the electricity is gone	sadness	negative	[the storm is here and the electricity is gone]	[the, storm, is, here, and, the, electricity, ...	[storm, electricity, gone]	[storm, electr, gone]

Lemmatization

```
[16] # 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.
     [['know', 'listenin', 'bad', 'habit', 'earlier', 'started', 'freakin', 'part'], ['sleep', 'im', 'thinking', 'old', 'friend', 'want', 'married', 'want', '2',

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

[17]

	textID	text	sentiment_main	sentiment	sent_token	word_token	stop_words_removed	stemmed_words	lemmatized_words
0	p1000000000	i know i was listenin to bad habit earlier a...	empty	neutral	[i know i was listenin to bad habit earlier ...	[i, know, i, was, listenin, to, bad, habit, ea...	[know, listenin, bad, habit, earlier, started,...	[know, listenin, bad, habit, earlier, start, f...	[know, listenin, bad, habit, earlier, started,...
1	p1000000001	i should be sleep, but im not! thinking about ...	sadness	negative	[i should be sleep, but im not!, thinking abou...	[i, should, be, sleep, ,, but, im, not, !, thi...	[sleep, ,, im, !, thinking, old, friend, want,...	[sleep, im, think, old, friend, want, marri, w...	[sleep, im, thinking, old, friend, want, marri...
2	2dfbe0b7fb	hmmm. http://www.djhero.com/ is down	worry	negative	[hmmm., http://www.djhero.com/ is down]	[hmmm, ., http, :, //www.djhero.com/, is, down]	[hmmm, ., http, :, //www.djhero.com/]	[hmmm, http]	[hmmm, http]
3	6d846d7d50	i`m sorry at least it`s friday?	sadness	neutral	[i`m sorry at least it`s friday?]	[i`m, sorry, at, least, it`s, friday, ?]	[i`m, sorry, least, it`s, friday, ?]	[sorri, least, friday]	[sorry, least, friday]
4	p1000000002	the storm is here and the electricity is gone	sadness	negative	[the storm is here and the electricity is gone]	[the, storm, is, here, and, the, electricity, ...	[storm, electricity, gone]	[storm, electr, gone]	[storm, electricity, gone]

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
```

```
know : [0, 6, 64, 104, 114, 134, 153, 166, 171, 210, 217, 243, 278, 292, 295, 325, 344, 449, 465, 467, 487, 492, 511, 550, 566, 593, 595, 667, 680, 726, 754,
listenin : [0, 5864, 9881, 11009]
bad : [0, 16, 52, 121, 130, 158, 232, 257, 278, 324, 415, 424, 479, 480, 625, 696, 722, 732, 733, 752, 822, 829, 855, 903, 936, 943, 971, 1041, 1097, 1112,
habit : [0, 6345, 8659, 10521]
earlier : [0, 633, 1372, 1804, 2053, 2892, 3031, 3765, 3943, 4064, 4456, 6123, 7910, 8572, 8700, 11104, 12030]
start : [0, 61, 122, 144, 163, 263, 326, 343, 387, 582, 711, 861, 885, 1007, 1040, 1081, 1145, 1297, 1313, 1415, 1531, 1581, 1586, 1590, 1647, 1732, 1958, 19
freakin : [0, 794, 1325, 2103, 2222, 2374, 3293, 3714, 3824, 6355, 6823, 9057, 11436]
part : [0, 24, 436, 2544, 3304, 3410, 3961, 4068, 4187, 6209, 6277, 6626, 6924, 7195, 7908, 8768, 8872, 9347, 9897, 9900, 10060, 10304, 11015, 11259, 11275,
sleep : [1, 24, 116, 192, 196, 214, 250, 293, 301, 305, 312, 340, 352, 373, 392, 412, 481, 491, 507, 519, 525, 565, 637, 638, 711, 720, 723, 729, 733, 737,
im : [1, 35, 39, 54, 76, 77, 144, 217, 265, 269, 290, 339, 385, 408, 441, 478, 481, 491, 493, 497, 523, 551, 564, 584, 591, 597, 601, 608, 636, 637, 736, 74
think : [1, 7, 11, 21, 34, 38, 44, 48, 70, 74, 109, 115, 171, 172, 187, 189, 197, 217, 267, 317, 341, 343, 363, 370, 382, 404, 457, 462, 464, 474, 483, 495,
old : [1, 40, 155, 366, 499, 575, 677, 681, 755, 913, 1549, 1772, 1804, 1817, 1953, 2390, 2818, 2900, 2971, 3067, 3079, 3354, 3375, 3669, 4187, 4254, 4511,
friend : [1, 52, 72, 134, 193, 217, 266, 357, 403, 405, 445, 512, 516, 576, 620, 732, 849, 922, 1066, 1099, 1152, 1162, 1199, 1456, 1465, 1486, 1563, 1581,
want : [1, 29, 48, 61, 64, 91, 124, 150, 153, 226, 281, 292, 293, 298, 306, 312, 326, 335, 345, 346, 355, 410, 425, 440, 444, 450, 454, 471, 474, 479, 493,
marri : [1, 2061, 3125, 3447, 4551, 4935, 7970, 9224, 11085, 11132]
2 : [1, 69, 77, 144, 313, 341, 343, 349, 352, 389, 478, 503, 534, 622, 632, 658, 712, 719, 840, 879, 899, 943, 944, 961, 987, 1022, 1035, 1137, 1202, 1298,
scandal : [1, 10223]
```

[19]
```
hmmm : [2, 1921, 2267, 4416, 5510, 6122, 6436, 6679, 7131, 7203, 8351, 8583, 8731, 8891, 10870, 10950, 11560, 11574, 11713, 12083, 12426]
http : [2, 5, 33, 85, 191, 204, 207, 212, 230, 242, 272, 329, 330, 368, 376, 380, 389, 416, 437, 458, 520, 541, 556, 580, 591, 593, 600, 622, 663, 668, 677,
sorri : [3, 103, 121, 128, 175, 181, 218, 243, 267, 302, 324, 351, 428, 441, 489, 630, 736, 815, 845, 979, 990, 1033, 1158, 1174, 1207, 1327, 1374, 1421, 153
least : [3, 312, 571, 870, 961, 1023, 1318, 1345, 1506, 1705, 1714, 1818, 1859, 1861, 2064, 2213, 2384, 2406, 2707, 2793, 2877, 3156, 3321, 3601, 3737, 3744,
```

```
# Sorting index based on terms
final_train1=sorted(final_train.items())
final_train1
```

```
   11487]),
 ('awwwww', [4286, 4905, 5200, 7554, 9687, 9945]),
 ('awwwwww', [1547, 4118]),
 ('awwwwwww', [1343, 12353]),
 ('axayi', [6303]),
 ('axe', [11112]),
 ('aye', [4005, 6745, 9323, 12310]),
 ('ayi', [3241]),
 ('ayshea', [6949]),
 ('az', [1035, 2777]),
 ('azeroth', [3786]),
 ('aztec', [9218]),
 ('aztex', [3617]),
 ('azz', [6181, 10098]),
 ('azza', [10967]),
 ('b',
  [217,
   432,
   485,
   588,
   1071,
   1698,
   1766,
   2166,
   2716,
   2924,
   3692,
   3755,
   4584,
   4774,
```

Adding the module for timing the query response

```
[21] !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: 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: pexpect in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (4.8.0)
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: pygments in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (2.6.1)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (5.1.1)
Requirement already satisfied: simplegeneric>0.8 in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (0.8.1)
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: 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: 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: 136 µs (started: 2022-03-27 11:24:18 +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;
```

```
[30] know [430, {0: [0], 6: [2], 64: [8], 104: [0], 114: [1], 134: [8], 153: [8], 166: [0], 171: [4], 210: [3], 217: [9], 243: [5], 278: [13], 292: [1], 295: [1],
listenin [4, {0: [1], 5864: [3], 9881: [0], 11009: [2]}]
bad [189, {0: [2], 16: [10], 52: [0], 121: [7], 130: [0], 158: [7], 232: [1], 257: [6], 278: [12], 324: [5], 415: [0], 424: [5], 479: [5], 480: [3], 625: [6
habit [4, {0: [3], 6345: [2], 8659: [17], 10521: [8]}]
earlier [17, {0: [4], 633: [6], 1372: [4], 1804: [6], 2053: [3], 2892: [1], 3031: [9], 3765: [3], 3943: [8], 4064: [4], 4456: [3], 6123: [4], 7910: [1], 857
start [139, {0: [5], 61: [9], 122: [2], 144: [3], 163: [12], 263: [5], 326: [1], 343: [7], 387: [2], 582: [3], 711: [8], 861: [0], 885: [3], 1007: [4], 1040
freakin [14, {0: [6], 794: [0], 1325: [4], 2103: [5], 2222: [7], 2374: [0], 3293: [1], 3714: [1], 3824: [1], 6355: [7], 6823: [5], 9057: [4], 11436: [2]}]
part [30, {0: [7], 24: [6], 436: [8], 2544: [2], 3304: [6], 3410: [1], 3961: [10], 4068: [2], 4187: [15], 6209: [8], 6277: [2], 6626: [10], 6924: [2], 7195:
sleep [191, {1: [0], 24: [1], 116: [3], 192: [1], 196: [1], 214: [0], 250: [4], 293: [7], 301: [0], 305: [2], 312: [4], 340: [7], 352: [6], 373: [4], 392: [2
im [409, {1: [1], 35: [3], 39: [6], 54: [2], 76: [4], 77: [18], 144: [5], 217: [1], 265: [3], 269: [1], 290: [3], 339: [2], 385: [0], 408: [3], 441: [10], 4
think [407, {1: [2], 7: [1], 11: [7], 21: [0], 34: [5], 38: [2], 44: [3], 48: [0], 70: [4], 74: [8], 109: [3], 115: [3], 171: [9], 172: [0], 187: [0], 189:
old [90, {1: [3], 40: [17], 155: [5], 366: [10], 499: [5], 575: [3], 677: [5], 681: [1], 755: [1], 913: [1], 1549: [5], 1772: [6], 1804: [8], 1817: [4], 195
friend [203, {1: [4], 52: [1], 72: [0], 134: [3], 193: [1], 217: [4], 266: [3], 357: [9], 403: [3], 405: [4], 445: [3], 512: [4], 516: [3], 576: [2], 620: [4
want [396, {1: [7], 29: [0], 48: [1], 61: [8], 64: [0], 91: [5], 124: [0], 150: [0], 153: [9], 226: [0], 281: [5], 292: [5], 293: [0], 298: [3], 306: [3], 3
marri [11, {1: [6], 2061: [0], 3125: [2], 3447: [3], 4551: [2], 4935: [11], 7970: [7], 9224: [3], 11085: [7], 11132: [1]}]
2 [251, {1: [8], 69: [2], 77: [4], 144: [7], 313: [10], 341: [7], 343: [1], 349: [11], 352: [10], 389: [5], 478: [14], 503: [1], 534: [2], 622: [5], 632: [1
scandal [2, {1: [9], 10223: [1]}]
hmmm [23, {2: [0], 1921: [0], 2267: [0], 4416: [10], 5510: [0], 6122: [2], 6436: [0], 6679: [0], 7131: [0], 7203: [9], 8351: [0], 8583: [0], 8731: [2], 8891
http [591, {2: [1], 5: [9], 33: [4], 85: [7], 191: [0], 204: [0], 207: [4], 212: [10], 230: [0], 242: [5], 272: [4], 329: [0], 330: [6], 368: [6], 376: [6],
sorri [201, {3: [0], 103: [0], 121: [2], 128: [0], 175: [0], 181: [2], 218: [7], 243: [0], 267: [0], 302: [0], 324: [1], 351: [12], 428: [0], 441: [11], 489
time: 297 ms (started: 2022-03-27 11:28:51 +00:00)
```

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

```
 [6, {4286: [0], 4905: [1], 5200: [0], 7554: [0], 9687: [0], 9945: [2]}]),
('awwwwww', [2, {1547: [0], 4118: [0]}]),
('awwwwwww', [2, {1343: [0], 12353: [9]}]),
('axayi', [1, {6303: [5]}]),
('axe', [1, {11112: [4]}]),
('aye', [4, {4005: [5], 6745: [3], 9323: [0], 12310: [0]}]),
('ayi', [1, {3241: [0]}]),
('ayshea', [1, {6949: [2]}]),
('az', [2, {1035: [2], 2777: [1]}]),
('azeroth', [1, {3786: [6]}]),
('aztec', [1, {9218: [0]}]),
('aztex', [1, {3617: [2]}]),
('azz', [2, {6181: [1], 10098: [4]}]),
('azza', [1, {10967: [1]}]),
('b',
 [45,
  {217: [8],
   432: [9],
   485: [1],
   588: [15],
   1071: [0],
   1698: [4],
   1766: [5],
   2166: [9],
```

Performing Boolean Queries

```
[22] # 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: 6.52 ms (started: 2022-03-27 11:25:03 +00:00)
```

```
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: 15.3 ms (started: 2022-03-27 11:25:08 +00:00)
```

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

```
Enter the first input word :
bad
Enter the second input word :
start
time: 7.11 s (started: 2022-03-27 11:26:03 +00:00)
```

```
[26] 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 bad [0, 16, 52, 121, 130, 158, 232, 257, 278, 324, 415, 424, 479, 480, 625, 696, 722, 732, 733, 752, 822, 829, 855, 903, 936, 943, 971, 104
posting list for start [0, 61, 122, 144, 163, 263, 326, 343, 387, 582, 711, 861, 885, 1007, 1040, 1081, 1145, 1297, 1313, 1415, 1531, 1581, 1586, 1590, 1647,
Resultant list:  [0, 1531, 1581, 7320]
time: 7.71 ms (started: 2022-03-27 11:26:37 +00:00)
```

```
[27] 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)))
```

```
[27] Resultant list:  [0, 16, 52, 61, 121, 122, 130, 144, 158, 163, 232, 257, 263, 278, 324, 326, 343, 387, 415, 424, 479, 480, 582, 625, 696, 711, 722, 732, 733,
     Length of posting list for bad 186
     Length of posting list for start 139
     Length of and list:  4
     Resultant list : [0, 16, 52, 61, 121, 122, 130, 144, 158, 163, 232, 257, 263, 278, 324, 326, 343, 387, 415, 424, 479, 480, 582, 625, 696, 711, 722, 732, 733,
     Length of the OR list: 321
     time: 16.3 ms (started: 2022-03-27 11:26:43 +00:00)

[28] 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 :
     want
     Enter the fourth input word :
     sleep
     Result: [0, 1, 29, 48, 61, 64, 91, 122, 124, 144, 150, 153, 163, 226, 263, 281, 292, 293, 298, 306, 312, 326, 335, 343, 345, 346, 355, 387, 410, 425, 440, 44
     Result length: 523
     time: 6.91 s (started: 2022-03-27 11:27:13 +00:00)

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

     Result: [0, 1, 61, 293, 312, 326, 519, 733, 1046, 1112, 1455, 1531, 1581, 2528, 3366, 4674, 4812, 4986, 7000, 7320, 10314, 10902]
     Result length: 22
     time: 7.39 ms (started: 2022-03-27 11:27:38 +00:00)
```

Performing Phrase Query on Inverted Index

```
[32] # PHRASE QUERY on Inverted Index :
     def phrase_query(phr):
         query=phr.split();
         for i in range(0,len(query)-1,2):
           result=and_query(final_train[query[i]],final_train[query[i+1]])
         print(result)
     print("Enter your query")
     q=input()
     phrase_query(q)

     Enter your query
     want sleep
     [1, 293, 312, 519, 1046, 2528, 3366, 4812, 4986, 10314]
     time: 17.1 s (started: 2022-03-27 11:31:56 +00:00)

     print("Enter your query")
     q=input()
     phrase_query(q)

     Enter your query
     old friend want
     [1, 2971, 3067, 3079, 11077]
     time: 4.38 s (started: 2022-03-27 11:32:17 +00:00)
```

Performing Phrase Query on Positional Index

```
[34] # 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: 7.98 ms (started: 2022-03-27 11:33:24 +00:00)
```

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

```
Enter your query
want sleep friend
[1, 293, 312, 519, 1046, 2528, 3366, 4812, 4986, 10314]
time: 5.12 s (started: 2022-03-27 11:33:32 +00:00)
```

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

```
Enter your query
think sleep old friend
[1, 2971, 3067, 3079, 11077]
time: 5.56 s (started: 2022-03-27 11:33:41 +00:00)
```

CORPUS – 3 : *NEWS HEADLINES ANALYSIS CORPUS*

DATASET LOCATION : DATASET > Headlines_5000.csv

NOTEBOOK NAME : A3_P3.ipynb

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)
```