UE19CS332 : Algorithms for Web and Information Retrieval
ASSIGNMENT - 1

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*

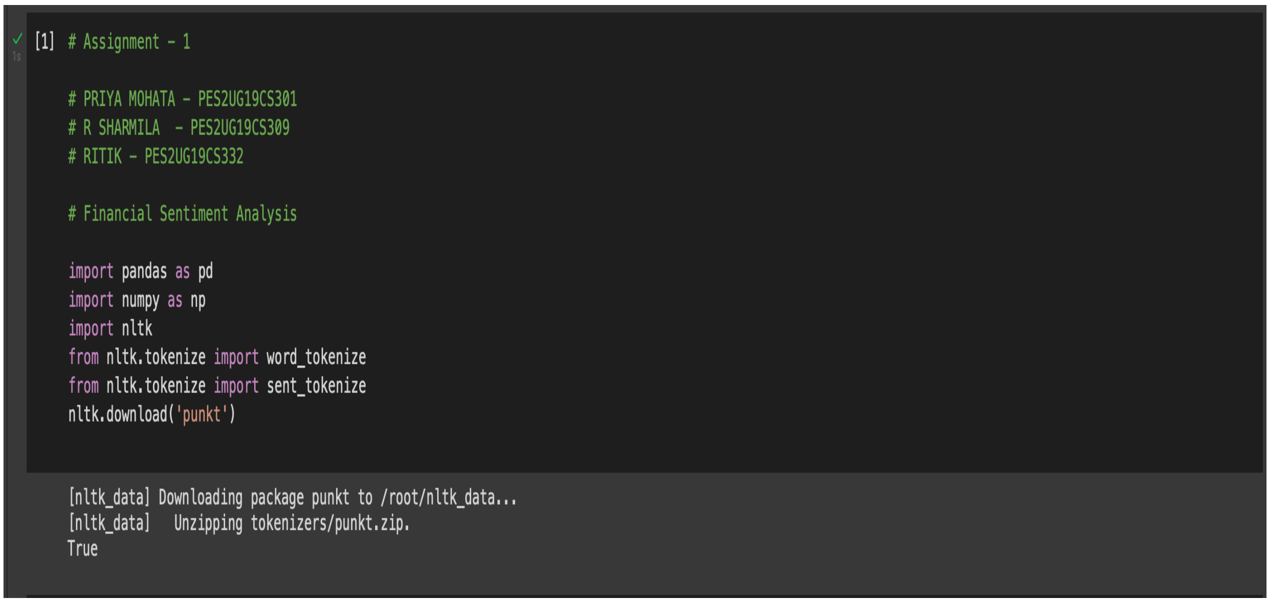
DATASET LOCATION :
https://drive.google.com/file/d/1ers7qOtpwLMUmM99YRf9uTZYDP6YdNRk/view?usp=sharing

NOTEBOOK NAME : A3_P1_TEAM-20.ipynb

Link to notebook :
https://colab.research.google.com/drive/1QRe456C7_BfwLM0_KZ9maj1bqIPb9J25?usp=sharing

STEPS :

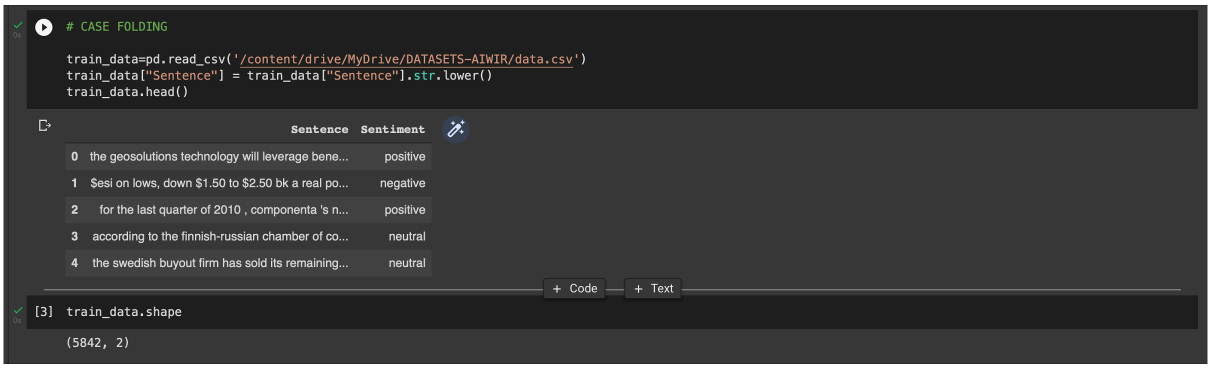
Importing all libraries



```
[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



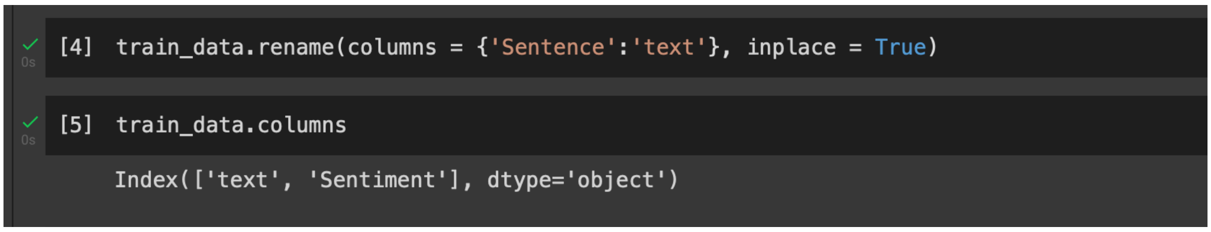
```
# CASE FOLDING

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
```

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

```
(5842, 2)
```

Renaming Columns

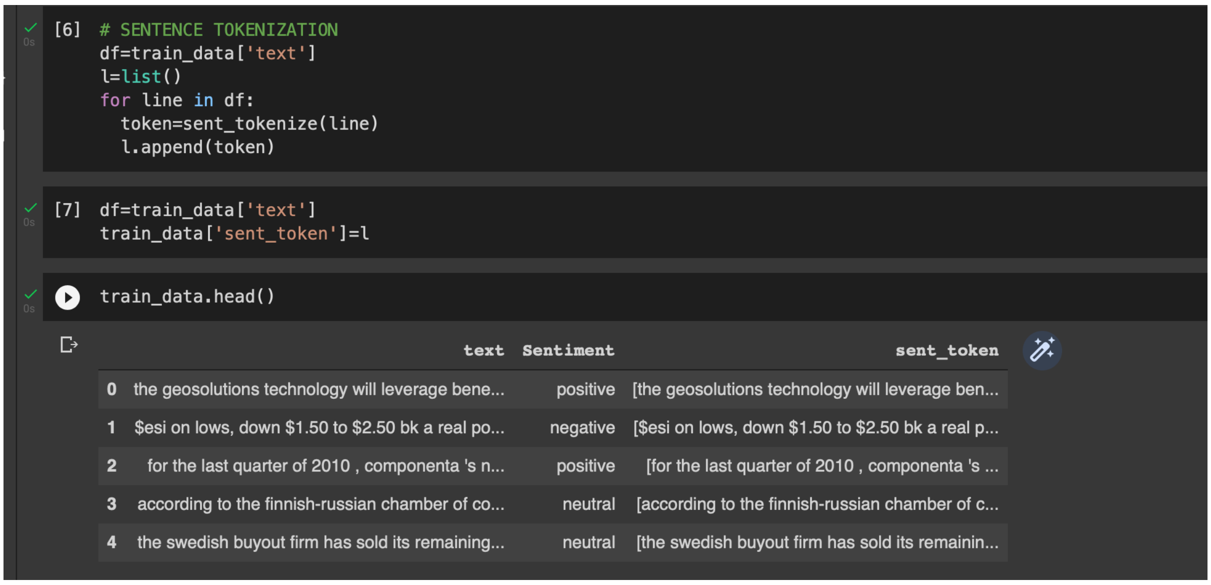


```
[4] train_data.rename(columns = {'Sentence':'text'}, inplace = True)
```

```
[5] train_data.columns
```

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

Sentence Tokenization



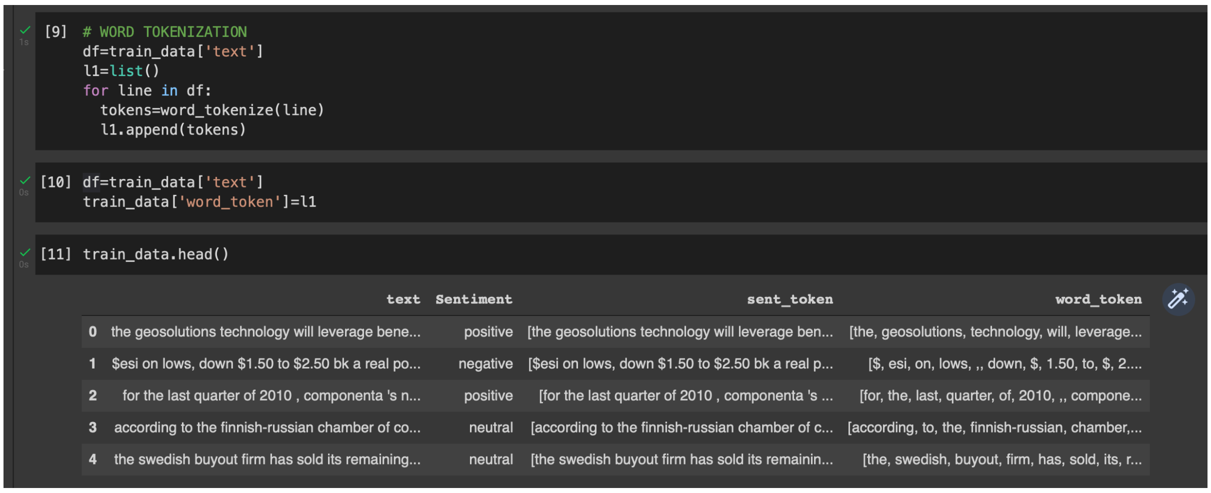
```
[6] # SENTENCE TOKENIZATION
    df=train_data['text']
    l=list()
    for line in df:
      token=sent_tokenize(line)
      l.append(token)
```

```
[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



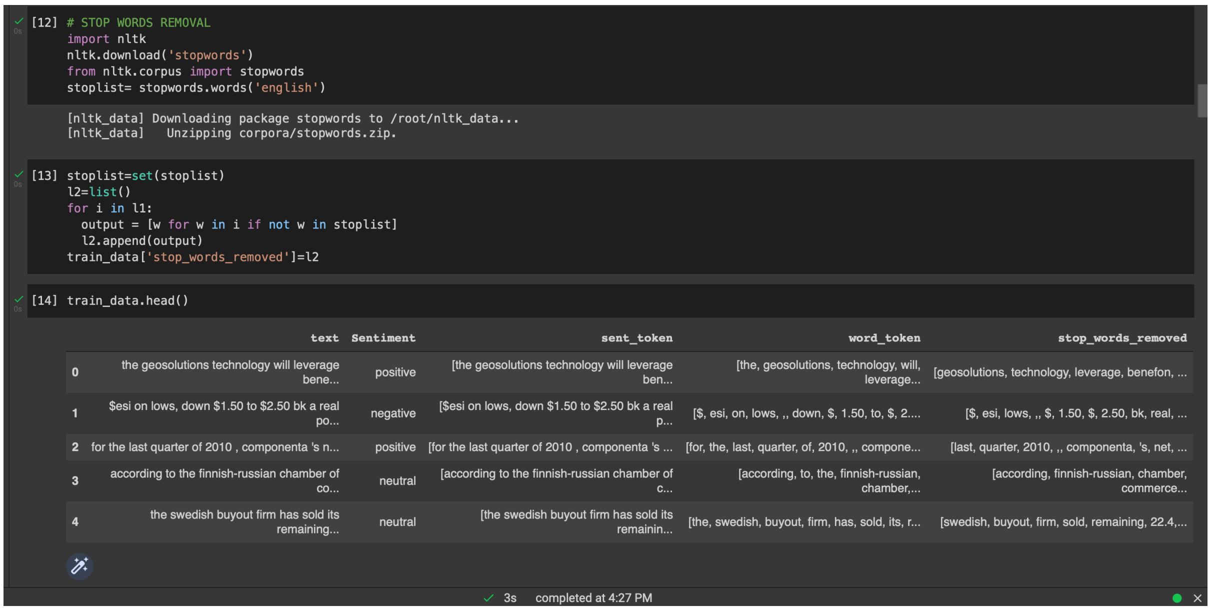
```
[9]  # WORD TOKENIZATION
     df=train_data['text']
     l1=list()
     for line in df:
       tokens=word_tokenize(line)
       l1.append(tokens)
```

```
[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.
```

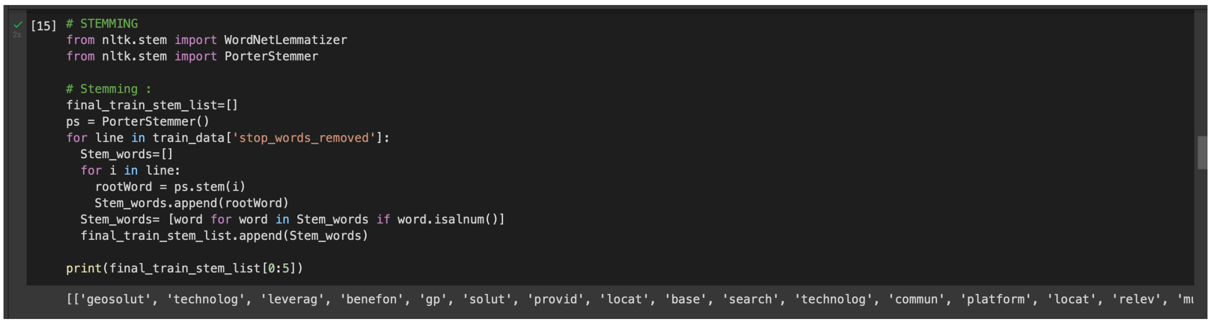
```
[13] stoplist=set(stoplist)
     l2=list()
     for i in l1:
       output = [w for w in i if not w in stoplist]
       l2.append(output)
     train_data['stop_words_removed']=l2
```

```
[14] train_data.head()
```

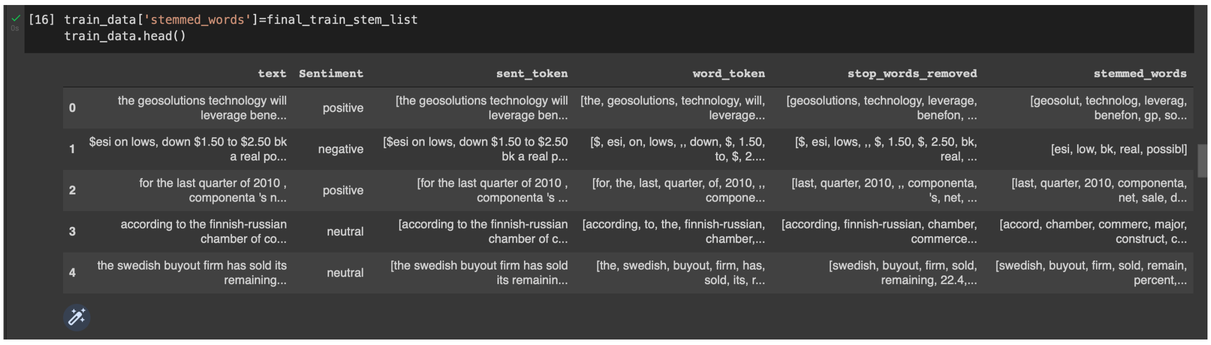
	text	Sentiment	sent_token	word_token	stop_words_removed
0	the geosolutions technology will leverage bene...	positive	[the geosolutions technology will leverage ben...	[the, geosolutions, technology, will, leverage...	[geosolutions, technology, leverage, benefon, ...
1	$esi on lows, down $1.50 to $2.50 bk a real po...	negative	[$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



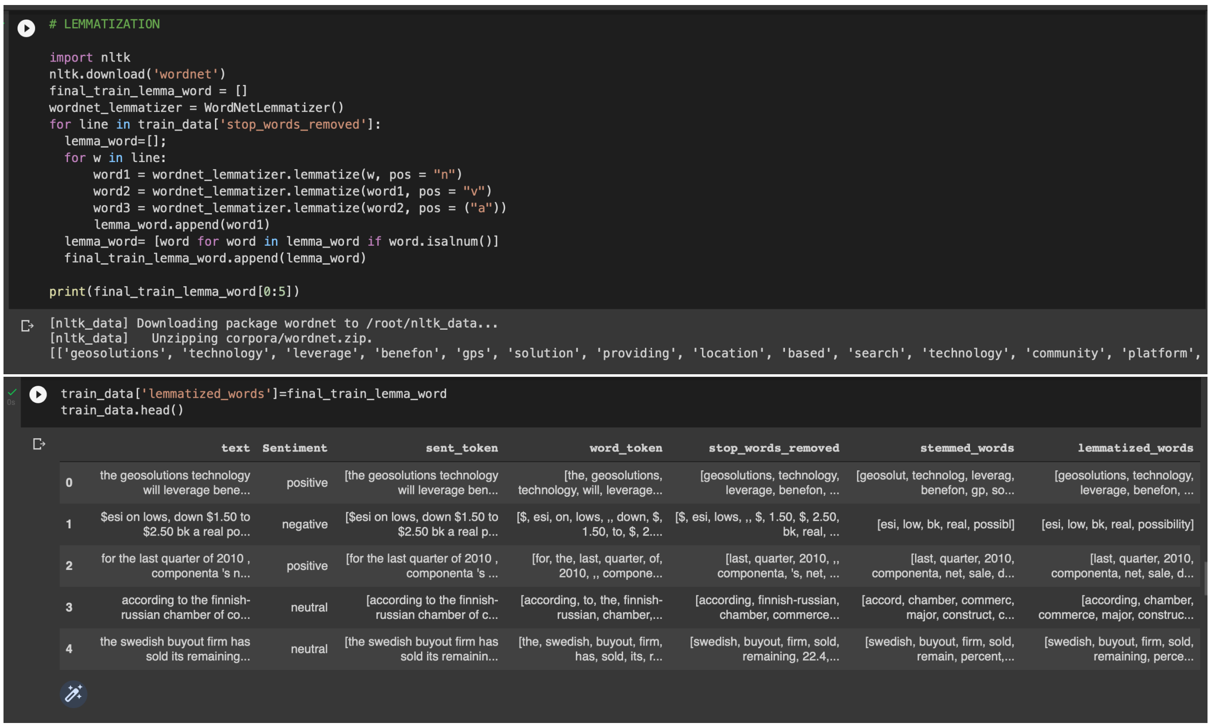
```
[15] # STEMMING
     from nltk.stem import WordNetLemmatizer
     from nltk.stem import PorterStemmer

     # 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



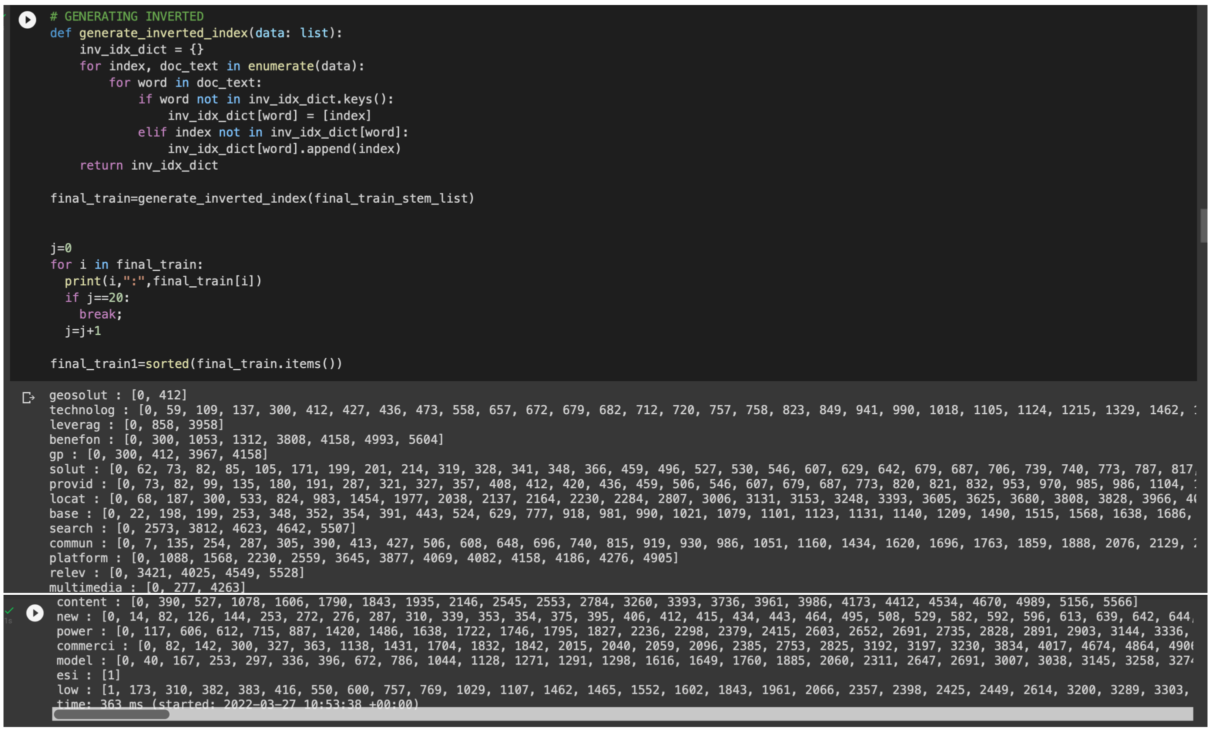
```
# LEMMATIZATION

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...

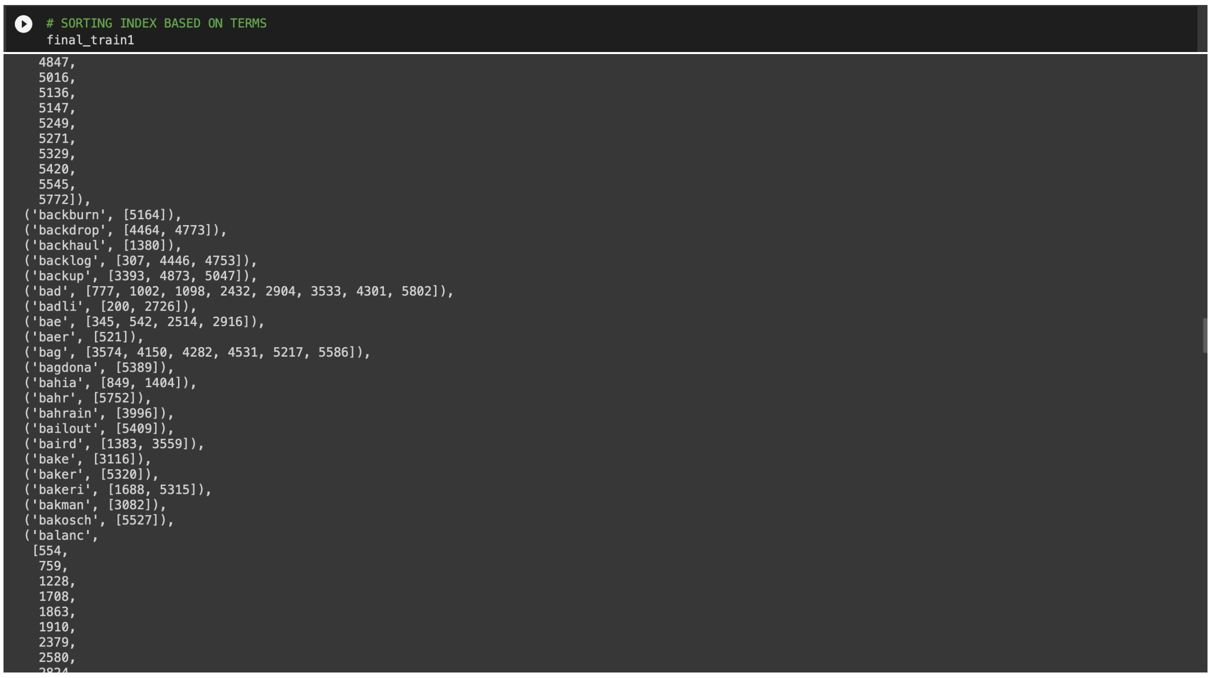
Building Inverted Index



```
# 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,
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,
 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,
 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)
```

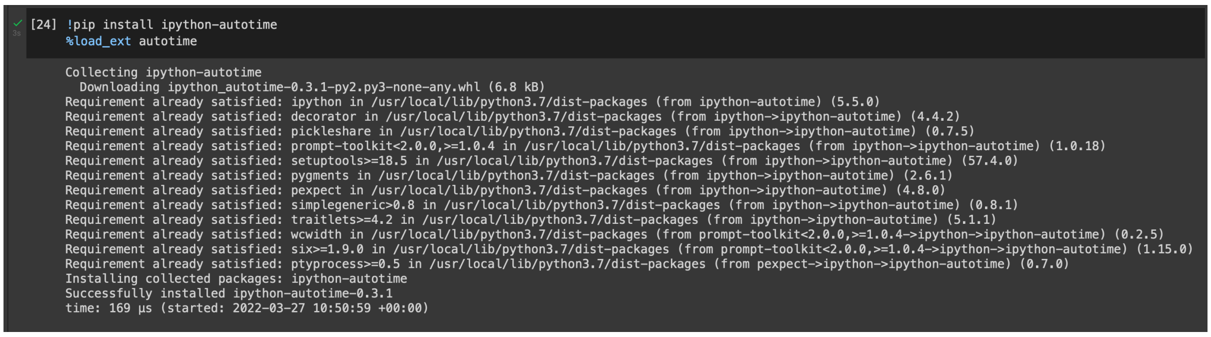
Sorting the index based on terms



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

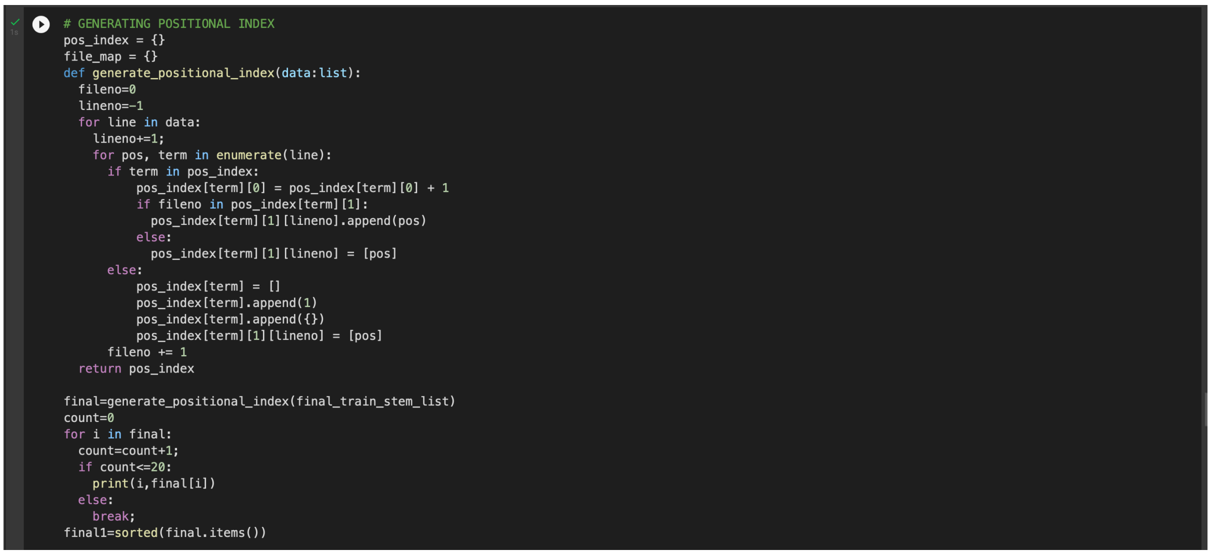
```
   4847,
   5016,
   5136,
   5147,
   5249,
   5271,
   5329,
   5420,
   5545,
   5772]),
 ('backburn', [5164]),
 ('backdrop', [4464, 4773]),
 ('backhaul', [1380]),
 ('backlog', [307, 4446, 4753]),
 ('backup', [3393, 4873, 5047]),
 ('bad', [777, 1002, 1098, 2432, 2904, 3533, 4301, 5802]),
 ('badli', [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

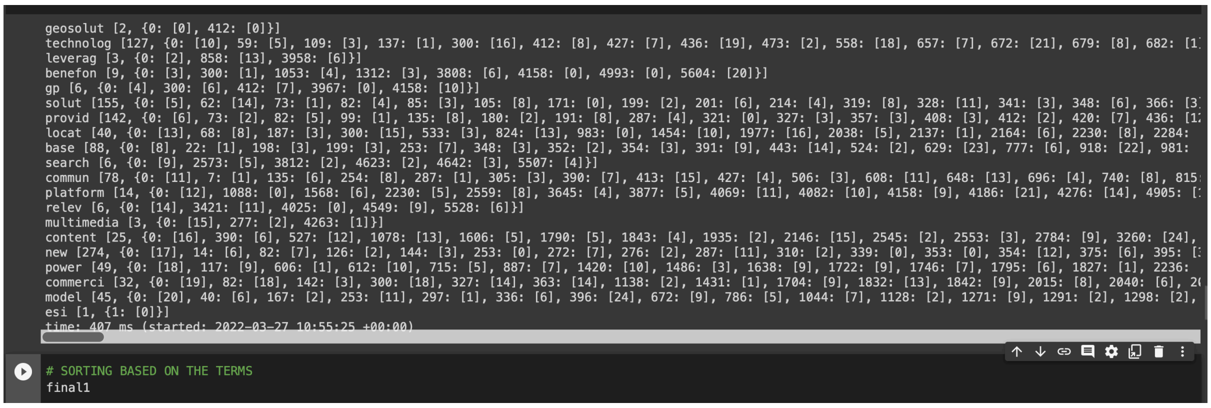


```
[24] !pip install ipython-autotime
     %load_ext autotime

     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



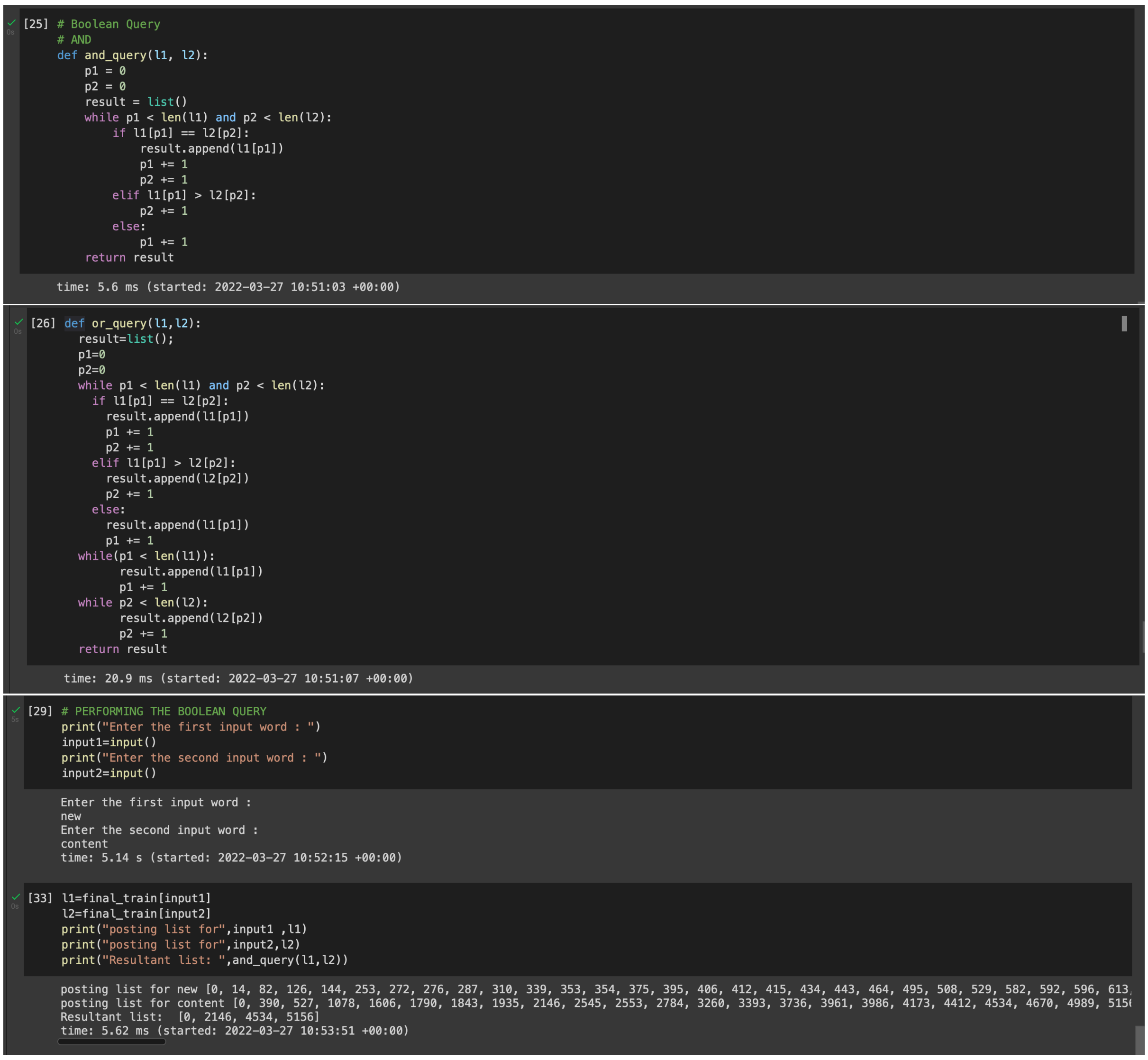
```python
# GENERATING POSITIONAL INDEX
pos_index = {}
file_map = {}
def generate_positional_index(data:list):
  fileno=0
  lineno=-1
  for line in data:
    lineno+=1;
    for pos, term in enumerate(line):
      if term in pos_index:
          pos_index[term][0] = pos_index[term][0] + 1
          if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

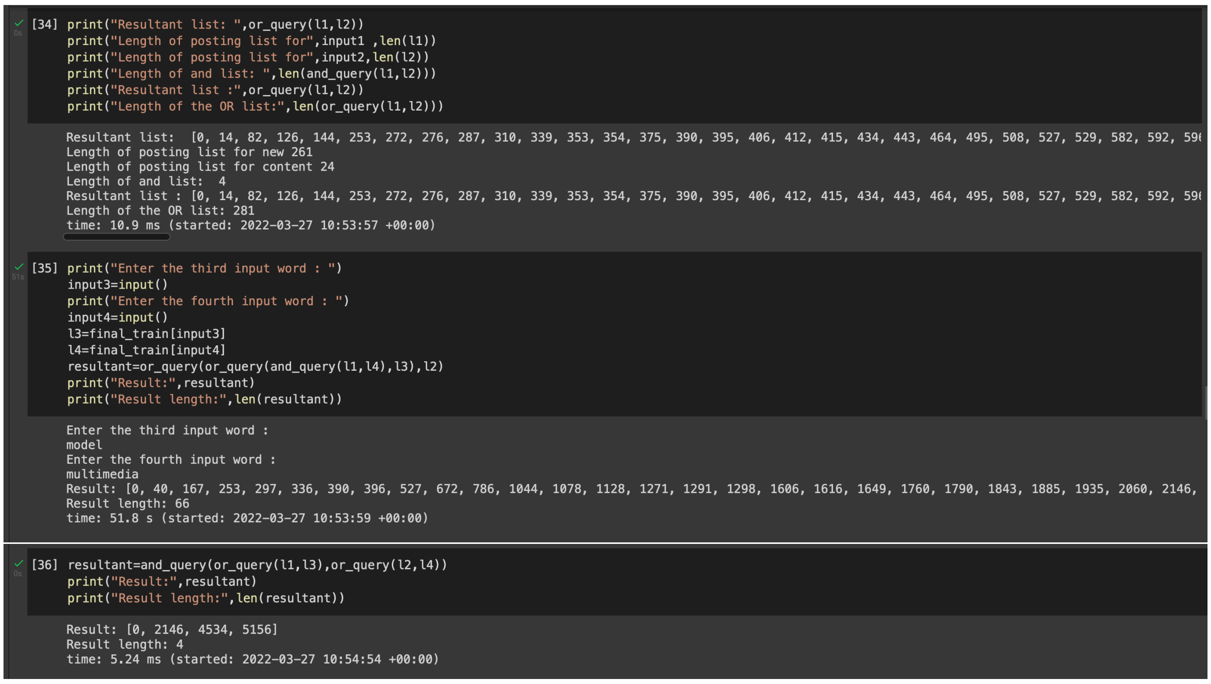


```
[25] # Boolean Query
     # AND
     def and_query(l1, l2):
         p1 = 0
         p2 = 0
         result = list()
         while p1 < len(l1) and p2 < len(l2):
             if l1[p1] == l2[p2]:
                 result.append(l1[p1])
                 p1 += 1
                 p2 += 1
          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

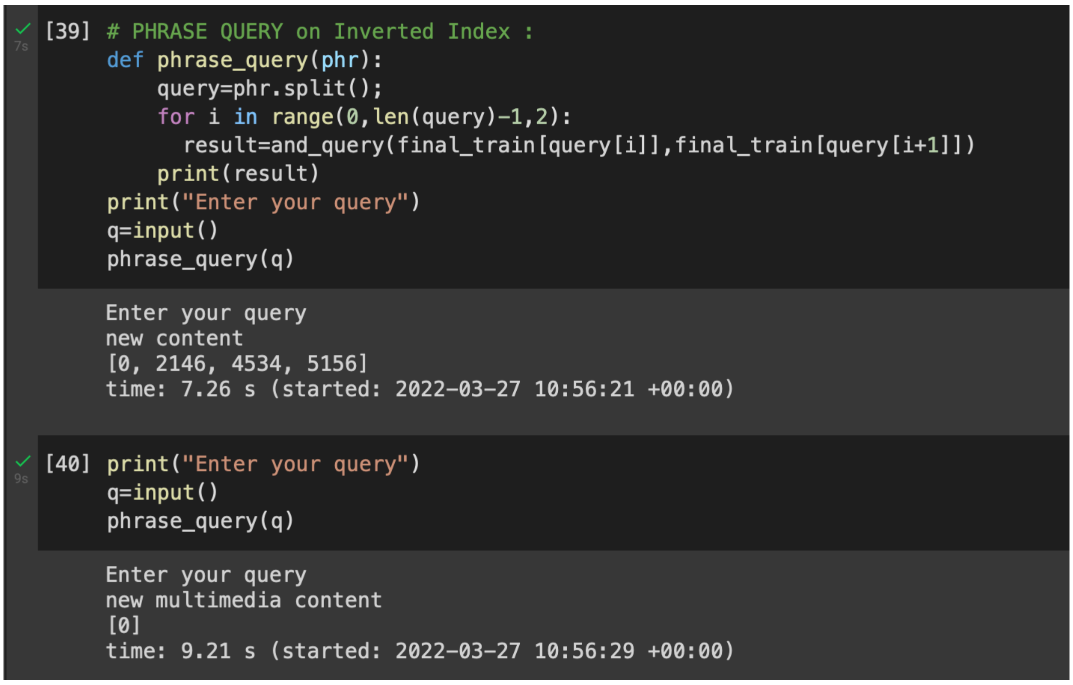
```
[29] # PERFORMING THE BOOLEAN QUERY
     print("Enter the first input word : ")
     input1=input()
     print("Enter the second input word : ")
     input2=input()

     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)
```



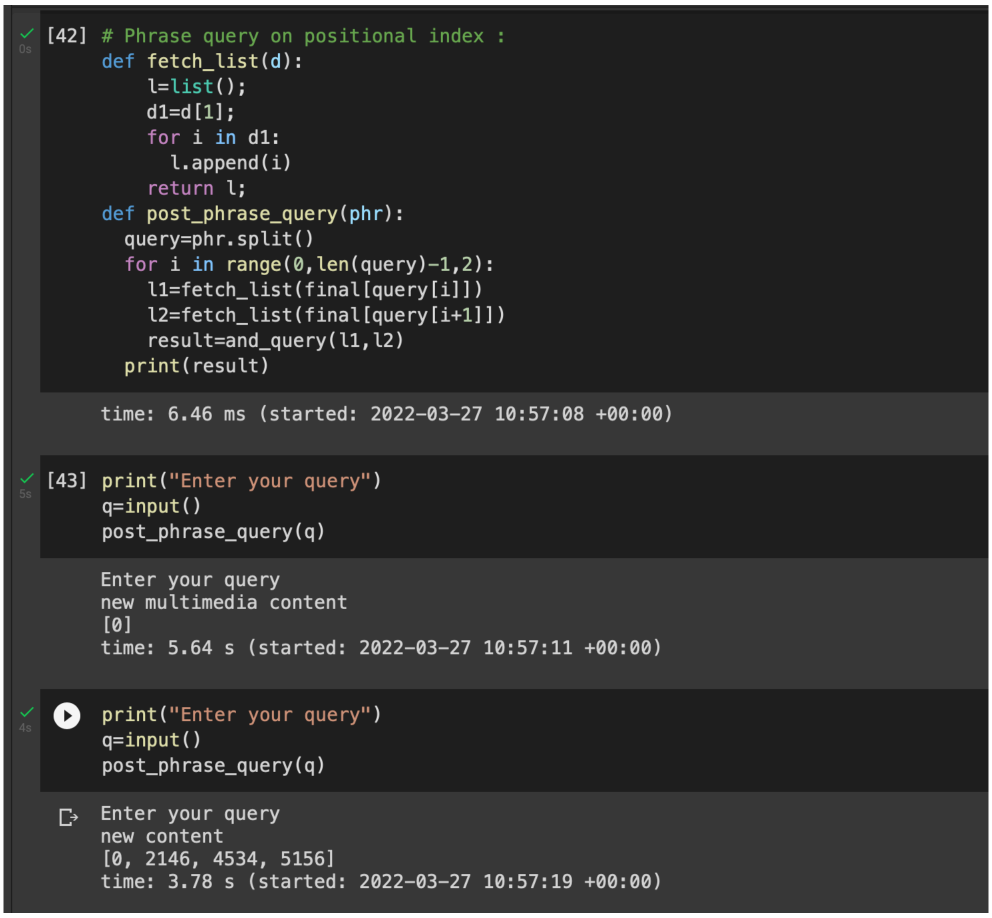
```
[34] print("Resultant list: ",or_query(l1,l2))
     print("Length of posting list for",input1 ,len(l1))
     print("Length of posting list for",input2,len(l2))
     print("Length of and list: ",len(and_query(l1,l2)))
     print("Resultant list :",or_query(l1,l2))
     print("Length of the OR list:",len(or_query(l1,l2)))

     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)
```