```
In [1]:  import eikon as ek
         from collections import Counter
         import pandas as pd
         from pandas import DataFrame
         import numpy as np
         from bs4 import BeautifulSoup
         from textblob import TextBlob
         import datetime
         from datetime import time
         import warnings
         warnings.filterwarnings("ignore")
         ek.set_app_key('....')
         import nltk
         from nltk.tokenize import RegexpTokenizer
         from nltk.stem import WordNetLemmatizer,PorterStemmer
         import re
```

```
In [2]:  import csv


         labels = {}


         with open(r'C:\Users\Smita Roy Trivedi\Desktop\forebroding_dict.csv', mode='r') as inp:
             reader = csv.reader(inp)
             labels = {rows[0]:rows[1] for rows in reader}


         print(labels)
```

{'achieved': 'negative', 'achievement': 'negative', 'acknowledged': 'negative', 'acknowledgemen
t': 'negative', 'adaptive': 'negative', 'advancement': 'negative', 'affirm': 'negative', 'affir
mation': 'negative', 'assurance': 'negative', 'assured': 'negative', 'blissful': 'negative', 'b
lithe': 'negative', 'boosted': 'negative', 'bounce': 'negative', 'bounced': 'negative', 'calm':
'negative', 'carefree': 'negative', 'certain': 'negative', 'certainty': 'negative', 'certitud
e': 'negative', 'cheer': 'negative', 'cheerful': 'negative', 'cheerfulness': 'negative', 'comfo
rt': 'negative', 'comfortable': 'negative', 'comforting': 'negative', 'confidence': 'negative',
'confident': 'negative', 'congratulatory': 'negative', 'constructive': 'negative', 'contended':
'negative', 'content': 'negative', 'contentment': 'negative', 'control': 'negative', 'controlla
bility': 'negative', 'controllable': 'negative', 'controlled': 'negative', 'credence': 'negativ
e', 'delight': 'negative', 'delighted': 'negative', 'delightful': 'negative', 'determination':
'negative', 'ecstastic ': 'negative', 'ecstasy': 'positive', 'effectively': 'negative', 'encour
age': 'negative', 'encouraged': 'negative', 'endurance': 'negative', 'exultant': 'negative', 'f
orbearance': 'negative', 'fortunate': 'negative', 'fortunately': 'negative', 'fruitful': 'negat
ive', 'fullfillment': 'negative', 'glad': 'negative', 'glee': 'negative', 'gleeful': 'negativ
e', 'happiness': 'negative', 'happy': 'negative', 'hope': 'negative', 'hopeful ': 'negative',
'hopefulness': 'negative', 'improvement': 'negative', 'improvement ': 'negative', 'improvement
s': 'negative', 'jovial': 'negative', 'joyful': 'negative', 'joyous': 'negative', 'lightearte
d': 'negative', 'lucky': 'negative', 'nonchalant': 'negative', 'optimism': 'negative', 'optimis
tic': 'negative', 'pleasing': 'negative', 'positivity': 'negative', 'recovery ': 'negative', 'r
ehabilitation ': 'negative', 'relaxed': 'negative', 'resilience': 'negative', 'resilient': 'neg
ative', 'restructuring': 'negative', 'resumption ': 'negative', 'revitalization ': 'negative',
'robustness': 'negative', 'self-assurance': 'negative', 'self-confidence': 'negative', 'stron
g': 'negative', 'stronger': 'negative', 'supporting': 'negative', 'supportive': 'negative', 'to
ughness': 'negative', 'tranquil': 'negative', 'trust': 'negative', 'trusting': 'negative', 'unc
oncerned': 'negative', 'undisturbed': 'negative', 'unperturbed': 'negative', 'unruffled': 'nega
tive', 'untroubled': 'negative', 'unworried': 'negative', 'unafraid': 'negative', 'afraid': 'po
sitive', 'alarmed': 'positive', 'alarming': 'positive', 'ambiguity': 'positive', 'ambiguous':
'positive', 'anxiety': 'positive', 'anxious': 'positive', 'anxiousness': 'positive', 'calamit
y': 'positive', 'cataclysms': 'positive', 'catastrophe': 'positive', 'devastated': 'positive',
'devastating': 'positive', 'disappointed': 'positive', 'disappointing': 'positive', 'disappoint
ment': 'positive', 'distress': 'positive', 'distressful': 'positive', 'distressing': 'positiv
e', 'disturbed': 'positive', 'disturbing': 'positive', 'doubt': 'positive', 'doubtful': 'positi
ve', 'defeated': 'positive', 'exacerbate': 'positive', 'exacerbating': 'positive', 'exacerbatio
n': 'positive', 'fatigue': 'positive', 'fear': 'positive', 'fearful': 'positive', 'forebrode':
'positive', 'forebroding': 'positive', 'forewarning': 'positive', 'indecision': 'positive', 'in
decisive': 'positive', 'misfortunes': 'positive', 'negativity': 'positive', 'nightmare': 'posit
ive', 'ominous': 'positive', 'overburdened': 'positive', 'panic': 'positive', 'panicked': 'posi
tive', 'premonition': 'positive', 'selfdefeating': 'positive', 'stressful': 'positive', 'suffe
r': 'positive', 'suffered': 'positive', 'suffering': 'positive', 'tentative': 'positive', 'thre
at': 'positive', 'threatened': 'positive', 'threatening': 'positive', 'trauma': 'positive', 'tr
aumas': 'positive', 'traumatic': 'positive', 'traumatizing': 'positive', 'trouble': 'positive',

```
'troublesome': 'positive', 'troubling': 'positive', 'turbulent': 'positive', 'uncertain': 'posi
tive', 'uncertainties': 'positive', 'uncertainty': 'positive', 'unclear': 'positive', 'unknow
n': 'positive', 'uncontrollable': 'positive', 'undecided': 'positive', 'undefined': 'positive',
'undermine': 'positive', 'undermined': 'positive', 'undermines': 'positive', 'unease': 'positiv
e', 'uneasily': 'positive', 'unfortunate': 'positive', 'unfortunately': 'positive', 'unpleasan
t': 'positive', 'unpredictable': 'positive', 'unravelling': 'positive', 'unreliable': 'positiv
e', 'unresolved': 'positive', 'unrest': 'positive', 'unsettled': 'positive', 'unsure': 'positiv
e', 'vague': 'positive', 'vulnerability': 'positive', 'vulnerable': 'positive', 'warn ': 'posit
ive', 'warning': 'positive', 'weak': 'positive', 'weakness': 'positive', 'worried': 'positive',
'worriers': 'positive', 'worries': 'positive', 'worrisome': 'positive', 'worry': 'positive', 'w
orrying': 'positive', 'apprehensive': 'positive', 'apprehension': 'positive', 'worsen': 'positi
ve', 'worst': 'positive', 'worse': 'positive'}
```

In [3]:
```python
def matcher(k):
    x = (i for i in labels if i in k)
    return ' | '.join(map(labels.get, x))
```

In [4]:
```python
df = ek.get_news_headlines('Topic:COVID AND Language:LEN AND Source:RTRS', date_from='2021-04-0
df.head()
```

Out[4]:

	versionCreated	text	storyId	sourceCode
2021-04-02 09:00:00	2021-04-02 09:00:00+00:00	Food bank, charities busy in Algarve as pandem...	urn:newsml:reuters.com:20210402:nL8N2LT6LR:1	NS:RTRS
2021-04-02 08:40:55	2021-04-02 08:40:55+00:00	Japanese copper smelter output seen falling de...	urn:newsml:reuters.com:20210402:nL1N2LV099:1	NS:RTRS
2021-04-02 08:15:01	2021-04-02 08:15:01+00:00	Russia reports 8,792 new COVID-19 cases, 400 d...	urn:newsml:reuters.com:20210402:nR4N2LL01B:1	NS:RTRS
2021-04-02 08:15:00	2021-04-02 08:15:00+00:00	RUSSIA REPORTS 8,792 NEW CORONAVIRUS CASES AND...	urn:newsml:reuters.com:20210402:nR4N2LL01B:1	NS:RTRS
2021-04-02 08:12:23	2021-04-02 08:12:23+00:00	BRIEF-Lenzing transferring stake in face-mask ...	urn:newsml:reuters.com:20210402:nFWN2LU11T:1	NS:RTRS

In [6]:
```python
df2 = ek.get_news_headlines('Topic:COVID AND Language:LEN AND Source:RTRS', date_from='2021-04-
df2['filtered_words']=np.nan
df2['flag_list']=np.nan
df2['counter']=np.nan
for idx, storyId in enumerate(df2['storyId'].values):  #for each row in our df2 dataframe
    newsText = ek.get_news_story(storyId) #get the news story
    if newsText:
        soup = BeautifulSoup(newsText,"lxml") #create a BeautifulSoup object from our HTML news
        sentence = newsText.lower()
        sentence=sentence.replace('{html}',"")
        cleanr = re.compile('<.*?>')
        cleantext = re.sub(cleanr, '', sentence)
        rem_url=re.sub(r'http\S+', '',cleantext)
        rem_num = re.sub('[0-9]+', '', rem_url)
        tokenizer = RegexpTokenizer(r'\w+')
        tokens = tokenizer.tokenize(rem_num)
        filtered_words = [w for w in tokens if len(w) > 2 ]
        df2['filtered_words'].iloc[idx] =  filtered_words
        df2['filtered_words'] = df2['filtered_words'].astype(str,'ignore')
        df2['flag_list']=df2['filtered_words'].apply(matcher)
        df2['flag_list'] = df2['flag_list'].astype(str,'ignore')
        df2['number']=df2['filtered_words'].str.count(r'[a-z]')
        df2['number2']=df2['flag_list'].str.count(r'[a-z]')
        df2['fore']=df2['filtered_words'].str.findall(r'\'\b(forebroding|uncertianty|fear|worry
        df2['fore'] = df2['fore'].astype(str,'ignore')
        df2['count_flag']=df2['fore'].str.count(r'[a-z]')
        df2['term_frequency']=df2['count_flag']/df2['number']
```

```python
df2['flag_list'] = df2['flag_list'].astype(str,'ignore')
df2['count_flag_pos']=df2['flag_list'].str.findall(r'(pos)')
df2['count_flag_pos'] = df2['count_flag_pos'].astype(str,'ignore')
df2['count_flag_pos1']=df2['count_flag_pos'].str.count(r'[a-z]')
df2['count_flag_neg']=df2['flag_list'].str.findall(r'neg')
df2['count_flag_neg'] = df2['count_flag_neg'].astype(str,'ignore')
df2['count_flag_neg1']=df2['count_flag_neg'].str.count(r'[a-z]')
df2['sentiment']=(df2['count_flag_pos1']+1)/(df2['count_flag_neg1']+1)
df2['sentiment2']=(df2['count_flag_pos1'])/(df2['count_flag_neg1']+df2['count_flag_pos1
df2.head()
```
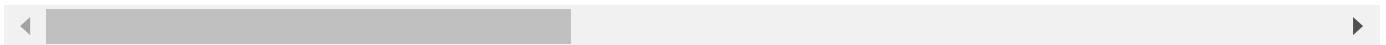
Out[6]:

	versionCreated	text	storyId	sourceCode	filtered_words
2021-04-03 08:11:05	2021-04-03 08:24:12+00:00	Philippines reports 12,576 new coronavirus cas…	urn:newsml:reuters.com:20210403:nL4N2LW027:1	NS:RTRS	['storycontent' 'border' 'color' 'inherit',…
2021-04-03 08:11:05	2021-04-03 08:11:05+00:00	PHILIPPINES HEALTH MINISTRY REPORTS 12,576 NEW…	urn:newsml:reuters.com:20210403:nL4N2LW027:1	NS:RTRS	['storycontent' 'border' 'color' 'inherit',…
2021-04-03 08:00:01	2021-04-03 08:00:01+00:00	Russia reports 9,021 new COVID-19 cases, 384 d…	urn:newsml:reuters.com:20210403:nR4N2JV02K:1	NS:RTRS	['storycontent' 'border' 'color' 'inherit',…
2021-04-03 08:00:00	2021-04-03 08:00:00+00:00	RUSSIA REPORTS 9,021 NEW CORONAVIRUS CASES AND…	urn:newsml:reuters.com:20210403:nR4N2JV02K:1	NS:RTRS	['storycontent' 'border' 'color' 'inherit',…
2021-04-03 07:32:27	2021-04-03 07:32:27+00:00	UPDATE 1-India braces for COVID-19 curbs as in…	urn:newsml:reuters.com:20210403:nL1N2LW02A:3	NS:RTRS	['storycontent' 'border' 'color' 'inherit',…

In [ ]:

In [ ]: