In [ ]:

```
# -*- coding: utf-8 -*-
"""
Created on Fri May 22 16:13:14 2020

@author: victo
"""

###necessary libraries###
import re
import nltk
nltk.downloader.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import pandas as pd
import glob
import os
from datetime import datetime

# file where csv files lies
path = r'C:\Users\victo\Master_Thesis\scraperproject\volvo\volvo_scraper\spiders\news'
all_files = glob.glob(os.path.join(path, "*.csv"))

# read files to pandas frame
list_of_files = []

for filename in all_files:
    list_of_files.append(pd.read_csv(filename,
                                     sep=',',
                                     encoding='cp1252',
                                     header=None,
                                     names=["url", "header", "release time", "article content"]
                                     )
                         )

# Concatenate all content of files into one DataFrame
concatenate_list_of_files = pd.concat(list_of_files,
                                      ignore_index=True,
                                      axis=0,
                                      )

# removing duplicates
cleaned_dataframe = concatenate_list_of_files.sort_values(by='url', ascending=False)
cleaned_dataframe = cleaned_dataframe.drop_duplicates(subset=["url"], keep='first', ignore_index=True)

print(cleaned_dataframe)

##formatting date column
dates = []
times = []
regex = r'(.*)(((1[0-2]|0?[1-9])\/(3[01]|[12][0-9]|0?[1-9])\/(?:[0-9]{2})?[0-9]{2})|((Jan(uary)?|Feb(ruary)?|Mar(ch)?|Apr(il)?|May|Jun(e)?|Jul(y)?|Aug(ust)?|Sep(tember)?|Oct(ober)?|Nov(ember)?|Dec(ember)?)\s+\d{1,2},\s+\d{4}))'
regex2 = r'((1[0-2]|0?[1-9]):([0-5][0-9]) ?([AaPp][Mm]))'

for date in cleaned_dataframe['release time']:
    matches = re.finditer(regex, date)
    for m in matches:
        date = m.group()
        date_formatted = date.replace(date[:2], '')
        convert_date = datetime.strptime(date_formatted, '%B %d, %Y')
        final_date = datetime.strftime(convert_date, "%Y-%m-%d")
        print(final_date)
        dates.append(final_date)

for time in cleaned_dataframe['release time']:
    matches = re.finditer(regex2, time)
    for t in matches:
        time = t.group()
        convert_time = datetime.strptime(time, '%I:%M %p')
        time_formatted = datetime.strftime(convert_time, '%H:%M:%S')
        print(time_formatted)
        times.append(time_formatted)

## adding modified date to data frame
cleaned_dataframe['date'] = dates
cleaned_dataframe['time'] = times
cleaned_dataframe['formatted date'] = cleaned_dataframe['date'] + str(' ') + cleaned_dataframe['time']

## dropping unnecessary columns
del cleaned_dataframe['date']
del cleaned_dataframe['time']

# New words and values
new_words = {'crushes': 10,
             'beats': 5,
             'misses': -5,
             'trouble': -10,
             'falls': -100,
             }

print('Start!')
# Instantiate the sentiment intensity analyzer with the existing lexicon
vader = SentimentIntensityAnalyzer()
# Update the lexicon
vader.lexicon.update(new_words)

print('ok!')

## analysis of article header
score_header = []

for header in cleaned_dataframe['header']:
    polarity_score = vader.polarity_scores(header)
    score_header.append(polarity_score)

# Join the DataFrames
cleaned_dataframe[['neg_vader_header',
                   'neu_vader_header',
                   'pos_vader_header',
                   'compound_vader_header'
                   ]] = pd.DataFrame(score_header)[['neg',
                                                    'neu',
                                                    'pos',
                                                    'compound'
                                                    ]]

## analysis of article content
score_content = []

for articlecontent in cleaned_dataframe['article content']:
    polarity_score = vader.polarity_scores(articlecontent)
    score_content.append(polarity_score)
```

```
# Join the DataFrames
cleaned_dataframe[['neg_vader_articel_content',
                   'neu_vader_articel_content',
                   'pos_vader_articel_content',
                   'compound_vader_articel_content'
                   ]] = pd.DataFrame(score_content)[['neg',
                                                     'neu',
                                                     'pos',
                                                     'compound'
                                                     ]]

print(cleaned_dataframe)

## saving outcome of vader to csv
current_date = datetime.today().strftime('%Y-%m-%d')
cleaned_dataframe.to_csv(r'C:\Users\victo\Master_Thesis\semanticanalysis\analysis_w
ith_vader\volvo\outcome_using_vader\outcome_of_vader_on_volvo_news_' + str(current_
date) + '.csv', index=False)
```