

▼ Preliminary Analysis of Statement Sentiment by Lexicon

Analyse statement by Loughran and McDonald word list to see if the relationship between economy and net sentiment

```
import os
import sys
IN_COLAB = 'google.colab' in sys.modules
IN_COLAB

    True

if IN_COLAB:
    from google.colab import drive
    drive.mount('/content/drive', force_remount=True)

    Mounted at /content/drive

#if IN_COLAB:
# # Uninstall existing versions:
# !pip uninstall bs4 -y
# !pip uninstall textract -y
# !pip uninstall numpy -y
# !pip uninstall pandas -y
# !pip uninstall requests -y
# !pip uninstall tqdm -y
# !pip uninstall nltk -y
# !pip uninstall quandl -y
# !pip uninstall scikit-plot -y
# !pip uninstall seaborn -y
# !pip uninstall sklearn -y
# !pip uninstall torch -y
# !pip uninstall transformers -y
# !pip uninstall wordcloud -y
# !pip uninstall xgboost -y
#
# # Install packages:
# !pip install bs4==0.0.1
# !pip install textract==1.6.3
# !pip install numpy==1.19.4
```

```

# !pip install pandas==1.1.4
# !pip install requests==2.24.0
# !pip install tqdm==4.51.0
# !pip install nltk==3.5
# !pip install quandl==3.5.3
# !pip install scikit-plot==0.3.7
# !pip install seaborn==0.11.0
# !pip install sklearn==0.0
# !pip install torch==1.7.1+cu101 torchvision==0.8.2+cu101 -f https://download.pytorch.org/whl/torch\_stable.html
# !pip install transformers==3.5.0
# !pip install wordcloud==1.8.0
# !pip install xgboost==1.2.1
# os.kill(os.getpid(), 9)

```

```

if IN_COLAB:

```

```

    employment_data_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/MarketData/Employment/'
    cpi_data_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/MarketData/CPI/'
    fed_rates_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/MarketData/FEDRates/'
    fx_rates_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/MarketData/FXRates/'
    gdp_data_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/MarketData/GDP/'
    ism_data_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/MarketData/ISM/'
    sales_data_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/MarketData/Sales/'
    treasury_data_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/MarketData/Treasury/'
    fomc_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/FOMC/'
    preprocessed_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/preprocessed/'
    train_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/train_data/'
    output_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/result/'
    keyword_lm_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/LoughranMcDonald/'
    glove_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/GloVe/'
    model_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/models/'
    graph_dir = '/content/drive/My Drive/Colab Notebooks/proj2/src/data/graphs/'

```

```

else:

```

```

    employment_data_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/MarketData/Employment/'
    cpi_data_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/MarketData/CPI/'
    fed_rates_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/MarketData/FEDRates/'
    fx_rates_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/MarketData/FXRates/'
    gdp_data_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/MarketData/GDP/'
    ism_data_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/MarketData/ISM/'
    sales_data_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/MarketData/Sales/'
    treasury_data_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/MarketData/Treasury/'
    fomc_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/FOMC/'
    preprocessed_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/preprocessed/'
    train_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/train_data/'
    output_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/result/'
    keyword_lm_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/LoughranMcDonald/'
    glove_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/GloVe/'
    model_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/models/'
    graph_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/graphs/'

```

```
tomc_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/FOMC/'
preprocessed_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/preprocessed/'
train_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/train_data/'
output_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/result/'
keyword_lm_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/LoughranMcDonald/'
glove_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/GloVe/'
model_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/models/'
graph_dir = 'C:/Users/theon/GDrive/Colab Notebooks/proj2/src/data/graphs/'
```

```
# Python libraries
```

```
import datetime as dt
import re
import pickle
from tqdm.notebook import tqdm
import time
import logging
import random
from collections import defaultdict, Counter
```

```
# Data Science modules
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
plt.style.use('ggplot')
```

```
# Import Scikit-learn modules
```

```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.metrics import accuracy_score, f1_score, plot_confusion_matrix
from sklearn.pipeline import Pipeline, FeatureUnion
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression, Perceptron, SGDClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn import model_selection
from sklearn.model_selection import GridSearchCV, cross_val_score, cross_validate, StratifiedKFold, learning_curve, RandomizedSearchCV
import scikitplot as skplt
```

```
# Import nltk modules and download dataset
```

```
import nltk
from nltk.corpus import stopwords
from nltk.util import ngrams
from nltk.tokenize import word_tokenize
```

```

nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')

stop = set(stopwords.words('english'))

# Import Pytorch modules

import torch
from torch import nn, optim
import torch.nn.functional as F
from torch.utils.data import (DataLoader, RandomSampler, SequentialSampler, TensorDataset)
from torch.autograd import Variable
from torch.optim import Adam, AdamW

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Package wordnet is already up-to-date!

## Use TPU Runtime:
#if IN_COLAB:
# assert os.environ['COLAB_TPU_ADDR'], 'Make sure to select TPU from Edit > Notebook setting > Hardware accelerator'
# VERSION = "20200220"
# !curl https://raw.githubusercontent.com/pytorch/xla/master/contrib/scripts/env-setup.py -o pytorch-xla-env-setup.py
# !python pytorch-xla-env-setup.py --version $VERSION

## Use GPU Runtime:
#if IN_COLAB:
# if torch.cuda.is_available():
#     torch.cuda.get_device_name(0)
#     gpu_info = !nvidia-smi
#     gpu_info = '\n'.join(gpu_info)
#     print(gpu_info)
# else:
#     print('Select the Runtime > "Change runtime type" menu to enable a GPU accelerator, and then re-execute this cell.')
#     os.kill(os.getpid(), 9)

# Set logger

```

```

logger = logging.getLogger('mylogger')
logger.setLevel(logging.INFO)

timestamp = time.strftime("%Y.%m.%d_%H.%M.%S", time.localtime())
formatter = logging.Formatter('[%(asctime)s][%(levelname)s] ## %(message)s')

fh = logging.FileHandler('log_model.txt')
fh.setLevel(logging.DEBUG)
fh.setFormatter(formatter)
logger.addHandler(fh)

ch = logging.StreamHandler()
ch.setLevel(logging.INFO)
ch.setFormatter(formatter)
logger.addHandler(ch)

# Set Random Seed
random.seed(42)
np.random.seed(42)
torch.manual_seed(42)
torch.cuda.manual_seed(42)
rand_seed = 42

# Set Seaborn Style
sns.set(style='white', context='notebook', palette='deep')

```

▼ Load preprocessed data

```

#file = open("/content/drive/My Drive/Colab Notebooks/proj2/src/data/FOMC/minutes.pickle", "rb")
#file = open("/content/drive/My Drive/Colab Notebooks/proj2/src/data/FOMC/statement.pickle", "rb")
file = open("/content/drive/My Drive/Colab Notebooks/proj2/src/data/FOMC/speech.pickle", "rb")
#file = open("/content/drive/My Drive/Colab Notebooks/proj2/src/data/FOMC/testimony.pickle", "rb")
Data = pickle.load(file)
file.close()

Data['text'] = Data['contents'].apply(lambda x: x.replace('\n\n[SECTION]\n\n', '').replace('\n', ' ').replace('\r', ' ').strip())
Data.drop(columns=['title'], axis=1, inplace=True)

```

```
print('Date: ', Data.iloc[-1]['date'].strftime('%Y-%m-%d'))
print('Speaker: ', Data.iloc[-1]['speaker'])
print('Text: \n\n', Data.iloc[-1]['contents'].replace('\n[SECTION]\n', '')[1192:])
```

Date: 2021-01-13
Speaker: Vice Chair Richard H. Clarida
Text:

by the Hoover Economic Policy Working Group, Hoover Institution, Stanford, California (via webcast)

On August 27, 2020, the Federal Open Market Committee (FOMC) unanimously approved a revised Statement on Longer-Run Goals and Monetary

Motivation for the Review

As my FOMC colleagues and I indicated from the outset, the fact that the Federal Reserve System chose to conduct this review does not

Perhaps the most significant change since 2012 in our understanding of the economy is our reassessment of the neutral real interest rate

The substantial decline in the neutral policy rate since 2012 has critical implications for the design, implementation, and communication

Two other, related developments that have also become more evident than they appeared in 2012 are that price inflation seems less responsive

With regard to inflation expectations, there is broad agreement among academics and policymakers that achieving price stability on a

The New Framework and Price Stability

Six features of the new framework and fall 2020 FOMC statements define how the Committee will seek to achieve its price-stability and

Second, with inflation having run persistently below 2 percent, the Committee will aim to achieve inflation moderately above 2 percent

Third, the Committee expects that appropriate monetary policy will remain accommodative for some time after the conditions to commence

Fourth, policy will aim over time to return inflation to its longer-run goal, which remains 2 percent, but not below, once the conditions

Fifth, inflation that averages 2 percent over time represents an ex ante aspiration of the FOMC, but not a time-inconsistent ex post

As I highlighted in a speech at the Brookings Institution in November, I believe that a useful way to summarize the framework defined

A policy that delays liftoff from the ELB until a threshold for average inflation has been reached is one element of a TPLT strategy.

In the TPLT studies I cited earlier, policy is assumed to revert to an inertial Taylor rule after liftoff, and therefore policy remains

Our new framework is asymmetric. That is, as in the previously cited TPLT studies, the goal of monetary policy after lifting off from

Our framework aims ex ante for inflation to average 2 percent over time, but it does not make a (time-inconsistent) commitment to achieve

The New Framework and Maximum Employment

Regarding our maximum-employment mandate—a sixth element—an important evolution in our new framework is that the Committee now de

Concluding Remarks

In closing, I think of our new flexible average inflation-targeting framework as a combination of TPLT at the ELB with flexible infla

▼ Creating a Financial Dictionary based on Loughran and McDonald

```
#Dictionary tone assessment will compare them by Index (need the numbers back)
```

```
Data['Index'] = range(0, len(Data))
```

```
# Make 'date' column as the index of Data
```

```
Data.set_index(['date'], inplace=True)
```

```
Data.head()
```

	contents	speaker	text	Index
date				
1996-06-13	Remarks by Chairman Alan Greenspan\nBank super...	Chairman Alan Greenspan	Remarks by Chairman Alan Greenspan Bank superv...	0
1996-06-18	Remarks by Governor Edward W. Kelley, Jr.\nDev...	Governor Edward W. Kelley, Jr.	Remarks by Governor Edward W. Kelley, Jr. Deve...	1
1996-09-08	Monetary Policy Objectives and Strategy\n\n[SE...	Governor Laurence H. Meyer	Monetary Policy Objectives and Strategyl want ...	2
1996-	Remarks by Chairman Alan	Chairman Alan	Remarks by Chairman Alan	3

```
import re
```

```
# Loughran and McDonald Sentiment Word Lists (https://sraf.nd.edu/textual-analysis/resources/)
```

```
lmdict = {'Negative': ['abandon', 'abandoned', 'abandoning', 'abandonment', 'abandonments', 'abandons', 'abdicated',  
    'abdicates', 'abdication', 'abdications', 'aberrant', 'aberration', 'aberrational',  
    'aberrations', 'abetting', 'abnormal', 'abnormalities', 'abnormality', 'abnormally', 'abolish',  
    'abolished', 'abolishes', 'abolishing', 'abrogate', 'abrogated', 'abrogates', 'abrogating',  
    'abrogation', 'abrogations', 'abrupt', 'abruptly', 'abruptness', 'absence', 'absences',  
    'absenteeism', 'abuse', 'abused', 'abuses', 'abusing', 'abusive', 'abusively', 'abusiveness',  
    'accident', 'accidental', 'accidentally', 'accidents', 'accusation', 'accusations', 'accuse',  
    'accused', 'accuses', 'accusing', 'acquiesce', 'acquiesced', 'acquiesces', 'acquiescing',  
    'acquit', 'acquits', 'acquittal', 'acquittals', 'acquitted', 'acquitting', 'adulterate',  
    'adulterated', 'adulterating', 'adulteration', 'adulterations', 'adversarial', 'adversaries',  
    'adversary', 'adversarial', 'adversarial', 'adversarial', 'adversarial', 'adversarial', 'adversarial', 'adversarial']
```

adversary , adverse , adversely , adversities , adversity , attermatn , attermatns ,
'against', 'aggravate', 'aggravated', 'aggravates', 'aggravating', 'aggravation', 'aggravations',
'alerted', 'alerting', 'alienate', 'alienated', 'alienates', 'alienating', 'alienation',
'alienations', 'allegation', 'allegations', 'allege', 'alleged', 'allegedly', 'alleges',
'alleging', 'annoy', 'annoyance', 'annoyances', 'annoyed', 'annoying', 'annoys', 'annul',
'annulled', 'annulling', 'annulment', 'annulments', 'annuls', 'anomalies', 'anomalous',
'anomalously', 'anomaly', 'anticompetitive', 'antitrust', 'argue', 'argued', 'arguing',
'argument', 'argumentative', 'arguments', 'arrearage', 'arrearages', 'arrears', 'arrest',
'arrested', 'arrests', 'artificially', 'assault', 'assaulted', 'assaulting', 'assaults',
'assertions', 'attrition', 'aversely', 'backdating', 'bad', 'bail', 'bailout', 'balk', 'balked',
'bankrupt', 'bankruptcies', 'bankruptcy', 'bankrupted', 'bankrupting', 'bankrupts', 'bans',
'barred', 'barrier', 'barriers', 'bottleneck', 'bottlenecks', 'boycott', 'boycotted',
'boycotting', 'boycotts', 'breach', 'breached', 'breaches', 'breaching', 'break', 'breakage',
'breakages', 'breakdown', 'breakdowns', 'breaking', 'breaks', 'bribe', 'bribed', 'briberies',
'bribery', 'bribes', 'bribing', 'bridge', 'broken', 'burden', 'burdened', 'burdening', 'burdens',
'burdensome', 'burned', 'calamities', 'calamitous', 'calamity', 'cancel', 'canceled',
'canceling', 'cancellation', 'cancellations', 'cancelled', 'cancelling', 'cancels', 'careless',
'carelessly', 'carelessness', 'catastrophe', 'catastrophes', 'catastrophic', 'catastrophically',
'caution', 'cautionary', 'cautioned', 'cautioning', 'cautions', 'cease', 'ceased', 'ceases',
'ceasing', 'censure', 'censured', 'censures', 'censuring', 'challenge', 'challenged',
'challenges', 'challenging', 'chargeoffs', 'circumvent', 'circumvented', 'circumventing',
'circumvention', 'circumventions', 'circumvents', 'claiming', 'claims', 'clawback', 'closed',
'closeout', 'closeouts', 'closing', 'closings', 'closure', 'closures', 'coerce', 'coerced',
'coerces', 'coercing', 'coercion', 'coercive', 'collapse', 'collapsed', 'collapses',
'collapsing', 'collision', 'collisions', 'collude', 'colluded', 'colludes', 'colluding',
'collusion', 'collusions', 'collusive', 'complain', 'complained', 'complaining', 'complains',
'complaint', 'complaints', 'complicate', 'complicated', 'complicates', 'complicating',
'complication', 'complications', 'compulsion', 'concealed', 'concealing', 'concede', 'conceded',
'concedes', 'conceding', 'concern', 'concerned', 'concerns', 'conciliating', 'conciliation',
'conciliations', 'condemn', 'condemnation', 'condemnations', 'condemned', 'condemning',
'condemns', 'condone', 'condoned', 'confess', 'confessed', 'confesses', 'confessing',
'confession', 'confine', 'confined', 'confinement', 'confinements', 'confines', 'confining',
'confiscate', 'confiscated', 'confiscates', 'confiscating', 'confiscation', 'confiscations',
'conflict', 'conflicted', 'conflicting', 'conflicts', 'confront', 'confrontation',
'confrontational', 'confrontations', 'confronted', 'confronting', 'confronts', 'confuse',
'confused', 'confuses', 'confusing', 'confusingly', 'confusion', 'conspiracies', 'conspiracy',
'conspirator', 'conspiratorial', 'conspirators', 'conspire', 'conspired', 'conspires',
'conspiring', 'contempt', 'contend', 'contended', 'contending', 'contends', 'contention',
'contentions', 'contentious', 'contentiously', 'contested', 'contesting', 'contraction',
'contractions', 'contradict', 'contradicted', 'contradicting', 'contradiction', 'contradictions',
'contradictory', 'contradicts', 'contrary', 'controversial', 'controversies', 'controversy',
'convict', 'convicted', 'convicting', 'conviction', 'convictions', 'corrected', 'correcting',
'correction', 'corrections', 'corrects', 'corrupt', 'corrupted', 'corrupting', 'corruption',
'corruptions'. 'corruptlv'. 'corruptness'. 'costlv'. 'counterclaim'. 'counterclaimed'.

'counterpoise', 'counterpoised', 'counterpoises', 'counterpoising', 'counterclaim', 'counterclaimed',
'counterclaiming', 'counterclaims', 'counterfeit', 'counterfeited', 'counterfeiter',
'counterfeiters', 'counterfeiting', 'counterfeits', 'countermeasure', 'countermeasures', 'crime',
'crimes', 'criminal', 'criminally', 'criminals', 'crises', 'crisis', 'critical', 'critically',
'criticism', 'criticisms', 'criticize', 'criticized', 'criticizes', 'criticizing', 'crucial',
'crucially', 'culpability', 'culpable', 'culpably', 'cumbersome', 'curtail', 'curtailed',
'curtailing', 'curtailment', 'curtailments', 'curtails', 'cut', 'cutback', 'cutbacks',
'cyberattack', 'cyberattacks', 'cyberbullying', 'cybercrime', 'cybercrimes', 'cybercriminal',
'cybercriminals', 'damage', 'damaged', 'damages', 'damaging', 'dampen', 'dampened', 'danger',
'dangerous', 'dangerously', 'dangers', 'deadlock', 'deadlocked', 'deadlocking', 'deadlocks',
'deadweight', 'deadweights', 'debarment', 'debarments', 'debarred', 'deceased', 'deceit',
'deceitful', 'deceitfulness', 'deceive', 'deceived', 'deceives', 'deceiving', 'deception',
'deceptions', 'deceptive', 'deceptively', 'decline', 'declined', 'declines', 'declining',
'deface', 'defaced', 'defacement', 'defamation', 'defamations', 'defamatory', 'defame',
'defamed', 'defames', 'defaming', 'default', 'defaulted', 'defaulting', 'defaults', 'defeat',
'defeated', 'defeating', 'defeats', 'defect', 'defective', 'defects', 'defend', 'defendant',
'defendants', 'defended', 'defending', 'defends', 'defensive', 'defer', 'deficiencies',
'deficiency', 'deficient', 'deficit', 'deficits', 'defraud', 'defrauded', 'defrauding',
'defrauds', 'defunct', 'degradation', 'degradations', 'degrade', 'degraded', 'degrades',
'degrading', 'delay', 'delayed', 'delaying', 'delays', 'deleterious', 'deliberate',
'deliberated', 'deliberately', 'delinquencies', 'delinquency', 'delinquent', 'delinquently',
'delinquents', 'delist', 'delisted', 'delisting', 'delists', 'demise', 'demised', 'demises',
'demising', 'demolish', 'demolished', 'demolishes', 'demolishing', 'demolition', 'demolitions',
'demote', 'demoted', 'demotes', 'demoting', 'demotion', 'demotions', 'denial', 'denials',
'denied', 'denies', 'denigrate', 'denigrated', 'denigrates', 'denigrating', 'denigration',
'deny', 'denying', 'deplete', 'depleted', 'depletes', 'depleting', 'depletion', 'depletions',
'deprecation', 'depress', 'depressed', 'depresses', 'depressing', 'deprivation', 'deprive',
'deprived', 'deprives', 'depriving', 'derelict', 'dereliction', 'derogatory', 'destabilization',
'destabilize', 'destabilized', 'destabilizing', 'destroy', 'destroyed', 'destroying', 'destroys',
'destruction', 'destructive', 'detain', 'detained', 'detention', 'detentions', 'deter',
'deteriorate', 'deteriorated', 'deteriorates', 'deteriorating', 'deterioration',
'deteriorations', 'deterred', 'deterrence', 'deterrences', 'deterrent', 'deterrents',
'detering', 'deters', 'detract', 'detracted', 'detracting', 'detriment', 'detrimental',
'detrimentally', 'detriments', 'devalue', 'devalued', 'devalues', 'devaluing', 'devastate',
'devastated', 'devastating', 'devastation', 'deviate', 'deviated', 'deviates', 'deviating',
'deviation', 'deviations', 'devolve', 'devolved', 'devolves', 'devolving', 'difficult',
'difficulties', 'difficultly', 'difficulty', 'diminish', 'diminished', 'diminishes',
'diminishing', 'diminution', 'disadvantage', 'disadvantaged', 'disadvantageous', 'disadvantages',
'disaffiliation', 'disagree', 'disagreeable', 'disagreed', 'disagreeing', 'disagreement',
'disagreements', 'disagrees', 'disallow', 'disallowance', 'disallowances', 'disallowed',
'disallowing', 'disallows', 'disappear', 'disappearance', 'disappearances', 'disappeared',
'disappearing', 'disappears', 'disappoint', 'disappointed', 'disappointing', 'disappointingly',
'disappointment', 'disappointments', 'disappoints', 'disapproval', 'disapprovals', 'disapprove',
'disapproved', 'disapproves', 'disapproving', 'disassociates', 'disassociating',

'disassociation', 'disassociations', 'disaster', 'disasters', 'disastrous', 'disastrously',
'disavow', 'disavowal', 'disavowed', 'disavowing', 'disavows', 'disciplinary', 'disclaim',
'disclaimed', 'disclaimer', 'disclaimers', 'disclaiming', 'disclaims', 'disclose', 'disclosed',
'discloses', 'disclosing', 'discontinuance', 'discontinuances', 'discontinuation',
'discontinuations', 'discontinue', 'discontinued', 'discontinues', 'discontinuing', 'discourage',
'discouraged', 'discourages', 'discouraging', 'discredit', 'discredited', 'discrediting',
'discredits', 'discrepancies', 'discrepancy', 'disfavor', 'disfavored', 'disfavoring',
'disfavors', 'disgorge', 'disgorged', 'disgorgement', 'disgorgements', 'disgorges', 'disgorging',
'disgrace', 'disgraceful', 'disgracefully', 'dishonest', 'dishonestly', 'dishonesty', 'dishonor',
'dishonorable', 'dishonorably', 'dishonored', 'dishonoring', 'dishonors', 'disincentives',
'disinterested', 'disinterestedly', 'disinterestedness', 'disloyal', 'disloyally', 'disloyalty',
'dismal', 'dismally', 'dismiss', 'dismissal', 'dismissals', 'dismissed', 'dismisses',
'dismissing', 'disorderly', 'disparage', 'disparaged', 'disparagement', 'disparagements',
'disparages', 'disparaging', 'disparagingly', 'disparities', 'disparity', 'displace',
'displaced', 'displacement', 'displacements', 'displaces', 'displacing', 'dispose', 'dispossess',
'dispossessed', 'dispossesses', 'dispossessing', 'disproportion', 'disproportional',
'disproportionate', 'disproportionately', 'dispute', 'disputed', 'disputes', 'disputing',
'disqualification', 'disqualifications', 'disqualified', 'disqualifies', 'disqualify',
'disqualifying', 'disregard', 'disregarded', 'disregarding', 'disregards', 'disreputable',
'disrepute', 'disrupt', 'disrupted', 'disrupting', 'disruption', 'disruptions', 'disruptive',
'disrupts', 'dissatisfaction', 'dissatisfied', 'dissent', 'dissented', 'dissenter', 'dissenters',
'dissenting', 'dissents', 'dissident', 'dissidents', 'dissolution', 'dissolutions', 'distort',
'distorted', 'distorting', 'distortion', 'distortions', 'distorts', 'distract', 'distracted',
'distracting', 'distraction', 'distractions', 'distracts', 'distress', 'distressed', 'disturb',
'disturbance', 'disturbances', 'disturbed', 'disturbing', 'disturbs', 'diversion', 'divert',
'diverted', 'diverting', 'diverts', 'divest', 'divested', 'divesting', 'divestiture',
'divestitures', 'divestment', 'divestments', 'divests', 'divorce', 'divorced', 'divulge',
'divulged', 'divulges', 'divulging', 'doubt', 'doubted', 'doubtful', 'doubts', 'downgrade',
'downgraded', 'downgrades', 'downgrading', 'downsize', 'downsized', 'downsizes', 'downsizing',
'downsizings', 'downtime', 'downtimes', 'downturn', 'downturns', 'downward', 'downwards', 'drag',
'drastic', 'drastically', 'drawback', 'drawbacks', 'drop', 'dropped', 'drought', 'droughts', 'duress',
'dysfunction', 'dysfunctional', 'dysfunctions', 'easing', 'egregious', 'egregiously', 'embargo',
'embargoed', 'embargoes', 'embargoing', 'embarrass', 'embarrassed', 'embarrasses',
'embarrassing', 'embarrassment', 'embarrassments', 'embezzle', 'embezzled', 'embezzlement',
'embezzlements', 'embezzler', 'embezzles', 'embezzling', 'encroach', 'encroached', 'encroaches',
'encroaching', 'encroachment', 'encroachments', 'encumber', 'encumbered', 'encumbering',
'encumbers', 'encumbrance', 'encumbrances', 'endanger', 'endangered', 'endangering',
'endangerment', 'endangers', 'enjoin', 'enjoined', 'enjoining', 'enjoins', 'erode', 'eroded',
'erodes', 'eroding', 'erosion', 'erratic', 'erratically', 'erred', 'erring', 'erroneous',
'erroneously', 'error', 'errors', 'errs', 'escalate', 'escalated', 'escalates', 'escalating',
'evade', 'evaded', 'evades', 'evading', 'evasion', 'evasions', 'evasive', 'evict', 'evicted',
'evicting', 'eviction', 'evictions', 'evicts', 'exacerbate', 'exacerbated', 'exacerbates',
'exacerbating', 'exacerbation', 'exacerbations', 'exaggerate', 'exaggerated', 'exaggerates',

'exaggerating', 'exaggeration', 'excessive', 'excessively', 'exculpate', 'exculpated',
'exculpates', 'exculpating', 'exculpation', 'exculpations', 'exculpatory', 'exonerate',
'exonerated', 'exonerates', 'exonerating', 'exoneration', 'exonerations', 'exploit',
'exploitation', 'exploitations', 'exploitative', 'exploited', 'exploiting', 'exploits', 'expose',
'exposed', 'exposes', 'exposing', 'expropriate', 'expropriated', 'expropriates', 'expropriating',
'expropriation', 'expropriations', 'expulsion', 'expulsions', 'extenuating', 'fail', 'failed',
'failing', 'failings', 'fails', 'failure', 'failures', 'fallout', 'false', 'falsely',
'falsification', 'falsifications', 'falsified', 'falsifies', 'falsify', 'falsifying', 'falsity',
'fatalities', 'fatality', 'fatally', 'fault', 'faulted', 'faults', 'faulty', 'fear', 'fears',
'felonies', 'felonious', 'felony', 'fictitious', 'fined', 'fines', 'fired', 'firing', 'flaw',
'flawed', 'flaws', 'forbid', 'forbidden', 'forbidding', 'forbids', 'force', 'forced', 'forcing',
'foreclose', 'foreclosed', 'forecloses', 'foreclosing', 'foreclosure', 'foreclosures', 'forego',
'foregoes', 'foregone', 'forestall', 'forestalled', 'forestalling', 'forestalls', 'forfeit',
'forfeited', 'forfeiting', 'forfeits', 'forfeiture', 'forfeitures', 'forgers', 'forgery',
'fraud', 'frauds', 'fraudulence', 'fraudulent', 'fraudulently', 'frivolous', 'frivolously',
'frustrate', 'frustrated', 'frustrates', 'frustrating', 'frustratingly', 'frustration',
'frustrations', 'fugitive', 'fugitives', 'gratuitous', 'gratuitously', 'grievance', 'grievances',
'grossly', 'groundless', 'guilty', 'halt', 'halted', 'hamper', 'hampered', 'hampering',
'hampers', 'harass', 'harassed', 'harassing', 'harassment', 'hardship', 'hardships', 'harm',
'harmed', 'harmful', 'harmfully', 'harming', 'harms', 'harsh', 'harsher', 'harshest', 'harshly',
'harshness', 'hazard', 'hazardous', 'hazards', 'hinder', 'hindered', 'hindering', 'hinders',
'hindrance', 'hindrances', 'hostile', 'hostility', 'hurt', 'hurting', 'idle', 'idled', 'idling',
'ignore', 'ignored', 'ignores', 'ignoring', 'ill', 'illegal', 'illegalities', 'illegality',
'illegally', 'illegible', 'illicit', 'illicitly', 'illiquid', 'illiquidity', 'imbalance',
'imbalances', 'immature', 'immoral', 'impair', 'impaired', 'impairing', 'impairment',
'impairments', 'impairs', 'impasse', 'impasses', 'impede', 'impeded', 'impedes', 'impediment',
'impediments', 'impeding', 'impending', 'imperative', 'imperfection', 'imperfections', 'imperial',
'impermissible', 'implicate', 'implicated', 'implicates', 'implicating', 'impossibility',
'impossible', 'impound', 'impounded', 'impounding', 'impounds', 'impracticable', 'impractical',
'impracticalities', 'impracticality', 'imprisonment', 'improper', 'improperly', 'improprieties',
'impropriety', 'imprudent', 'imprudently', 'inability', 'inaccessible', 'inaccuracies',
'inaccuracy', 'inaccurate', 'inaccurately', 'inaction', 'inactions', 'inactivate', 'inactivated',
'inactivates', 'inactivating', 'inactivation', 'inactivations', 'inactivity', 'inadequacies',
'inadequacy', 'inadequate', 'inadequately', 'inadvertent', 'inadvertently', 'inadvisability',
'inadvisable', 'inappropriate', 'inappropriately', 'inattention', 'incapable', 'incapacitated',
'incapacity', 'incarcerate', 'incarcerated', 'incarcerates', 'incarcerating', 'incarceration',
'incarcerations', 'incidence', 'incidences', 'incident', 'incidents', 'incompatibilities',
'incompatibility', 'incompatible', 'incompetence', 'incompetency', 'incompetent',
'incompetently', 'incompetents', 'incomplete', 'incompletely', 'incompleteness', 'inconclusive',
'inconsistencies', 'inconsistency', 'inconsistent', 'inconsistently', 'inconvenience',
'inconveniences', 'inconvenient', 'incorrect', 'incorrectly', 'incorrectness', 'indecent',
'indecent', 'indefeasible', 'indefeasibly', 'indict', 'indictable', 'indicted', 'indicting',
'indictment', 'indictments', 'ineffective', 'ineffectively', 'ineffectiveness', 'inefficiencies',
'inefficiency', 'inefficient', 'inefficiently', 'ineligibility', 'ineligible', 'inequitable'

inefficiency, inefficient, inefficiently, ineligibility, ineligible, inequitable, 'inequitably', 'inequities', 'inequity', 'inevitable', 'inexperience', 'inexperienced', 'inferior', 'inflicted', 'infraction', 'infractions', 'infringe', 'infringed', 'infringement', 'infringements', 'infringes', 'infringing', 'inhibited', 'inimical', 'injunction', 'injunctions', 'injure', 'injured', 'injures', 'injuries', 'injuring', 'injurious', 'injury', 'inordinate', 'inordinately', 'inquiry', 'insecure', 'insensitive', 'insolvencies', 'insolvency', 'insolvent', 'instability', 'insubordination', 'insufficiency', 'insufficient', 'insufficiently', 'insurrection', 'insurrections', 'intentional', 'interfere', 'interfered', 'interference', 'interferences', 'interferes', 'interfering', 'intermittent', 'intermittently', 'interrupt', 'interrupted', 'interrupting', 'interruption', 'interruptions', 'interrupts', 'intimidation', 'intrusion', 'invalid', 'invalidate', 'invalidated', 'invalidates', 'invalidating', 'invalidation', 'invalidity', 'investigate', 'investigated', 'investigates', 'investigating', 'investigation', 'investigations', 'involuntarily', 'involuntary', 'irreconcilable', 'irreconcilably', 'irrecoverable', 'irrecoverably', 'irregular', 'irregularities', 'irregularity', 'irregularly', 'irreparable', 'irreparably', 'irreversible', 'jeopardize', 'jeopardized', 'justifiable', 'kickback', 'kickbacks', 'knowingly', 'lack', 'lacked', 'lacking', 'lackluster', 'lacks', 'lag', 'lagged', 'lagging', 'lags', 'lapse', 'lapsed', 'lapses', 'lapsing', 'late', 'laundering', 'layoff', 'layoffs', 'lie', 'limitation', 'limitations', 'lingering', 'liquidate', 'liquidated', 'liquidates', 'liquidating', 'liquidation', 'liquidations', 'liquidator', 'liquidators', 'litigant', 'litigants', 'litigate', 'litigated', 'litigates', 'litigating', 'litigation', 'litigations', 'lockout', 'lockouts', 'lose', 'loses', 'losing', 'loss', 'losses', 'lost', 'lying', 'malfeasance', 'malfunction', 'malfunctioned', 'malfunctioning', 'malfunctions', 'malice', 'malicious', 'maliciously', 'malpractice', 'manipulate', 'manipulated', 'manipulates', 'manipulating', 'manipulation', 'manipulations', 'manipulative', 'markdown', 'markdowns', 'misapplication', 'misapplications', 'misapplied', 'misapplies', 'misapply', 'misapplying', 'misappropriate', 'misappropriated', 'misappropriates', 'misappropriating', 'misappropriation', 'misappropriations', 'misbranded', 'miscalculate', 'miscalculated', 'miscalculates', 'miscalculating', 'miscalculation', 'miscalculations', 'mischaracterization', 'mischief', 'misclassification', 'misclassifications', 'misclassified', 'misclassify', 'miscommunication', 'misconduct', 'misdated', 'misdemeanor', 'misdemeanors', 'misdirected', 'mishandle', 'mishandled', 'mishandles', 'mishandling', 'misinform', 'misinformation', 'misinformed', 'misinforming', 'misinforms', 'misinterpret', 'misinterpretation', 'misinterpretations', 'misinterpreted', 'misinterpreting', 'misinterprets', 'misjudge', 'misjudged', 'misjudges', 'misjudging', 'misjudgment', 'misjudgments', 'mislabel', 'mislabeled', 'mislabeling', 'mislabelled', 'mislabels', 'mislead', 'misleading', 'misleadingly', 'misleads', 'misled', 'mismanage', 'mismanaged', 'mismanagement', 'mismanages', 'mismanaging', 'mismatch', 'mismatched', 'mismatches', 'mismatching', 'misplaced', 'misprice', 'mispricing', 'mispricings', 'misrepresent', 'misrepresentation', 'misrepresentations', 'misrepresented', 'misrepresenting', 'misrepresents', 'miss', 'missed', 'misses', 'misstate', 'misstated', 'misstatement', 'misstatements', 'misstates', 'misstating', 'misstep', 'missteps', 'mistake', 'mistaken', 'mistakenly', 'mistakes', 'mistaking', 'mistrial', 'mistrials', 'misunderstand', 'misunderstanding', 'misunderstandings', 'misunderstood', 'misuse', 'misused', 'misuses', 'misusing', 'monopolistic', 'monopolists', 'monopolization', 'monopolize', 'monopolized', 'monopolizes', 'monopolizing', 'monopoly', 'moratoria', 'moratorium', 'moratoriums'.

'mothballed', 'mothballing', 'negative', 'negatively', 'negatives', 'neglect', 'neglected',
'neglectful', 'neglecting', 'neglects', 'negligence', 'negligences', 'negligent', 'negligently',
'nonattainment', 'noncompetitive', 'noncompliance', 'noncompliances', 'noncompliant',
'noncomplying', 'nonconforming', 'nonconformities', 'nonconformity', 'nondisclosure',
'nonfunctional', 'nonpayment', 'nonpayments', 'nonperformance', 'nonperformances',
'nonperforming', 'nonproducing', 'nonproductive', 'nonrecoverable', 'nonrenewal', 'nuisance',
'nuisances', 'nullification', 'nullifications', 'nullified', 'nullifies', 'nullify',
'nullifying', 'objected', 'objecting', 'objection', 'objectionable', 'objectionably',
'objections', 'obscene', 'obscenity', 'obsolescence', 'obsolete', 'obstacle', 'obstacles',
'obstruct', 'obstructed', 'obstructing', 'obstruction', 'obstructions', 'offence', 'offences',
'offend', 'offended', 'offender', 'offenders', 'offending', 'offends', 'omission', 'omissions',
'omit', 'omits', 'omitted', 'omitting', 'onerous', 'opportunistic', 'opportunistically',
'oppose', 'opposed', 'opposes', 'opposing', 'opposition', 'oppositions', 'outage', 'outages',
'outdated', 'outmoded', 'overage', 'overages', 'overbuild', 'overbuilding', 'overbuilds',
'overbuilt', 'overburden', 'overburdened', 'overburdening', 'overcapacities', 'overcapacity',
'overcharge', 'overcharged', 'overcharges', 'overcharging', 'overcome', 'overcomes',
'overcoming', 'overdue', 'overestimate', 'overestimated', 'overestimates', 'overestimating',
'overestimation', 'overestimations', 'overload', 'overloaded', 'overloading', 'overloads',
'overlook', 'overlooked', 'overlooking', 'overlooks', 'overpaid', 'overpayment', 'overpayments',
'overproduced', 'overproduces', 'overproducing', 'overproduction', 'overrun', 'overrunning',
'overruns', 'overshadow', 'overshadowed', 'overshadowing', 'overshadows', 'overstate',
'overstated', 'overstatement', 'overstatements', 'overstates', 'overstating', 'oversupplied',
'oversupplies', 'oversupply', 'oversupplying', 'overtly', 'overturn', 'overturned',
'overturning', 'overturns', 'overvalue', 'overvalued', 'overvaluing', 'panic', 'panics',
'penalize', 'penalized', 'penalizes', 'penalizing', 'penalties', 'penalty', 'peril', 'perils',
'perjury', 'perpetrate', 'perpetrated', 'perpetrates', 'perpetrating', 'perpetration', 'persist',
'persisted', 'persistence', 'persistent', 'persistently', 'persisting', 'persists', 'pervasive',
'pervasively', 'pervasiveness', 'petty', 'picket', 'picketed', 'picketing', 'plaintiff',
'plaintiffs', 'plea', 'plead', 'pleaded', 'pleading', 'pleadings', 'pleads', 'pleas', 'pled',
'poor', 'poorly', 'poses', 'posing', 'postpone', 'postponed', 'postponement', 'postponements',
'postpones', 'postponing', 'precipitated', 'precipitous', 'precipitously', 'preclude',
'precluded', 'precludes', 'precluding', 'predatory', 'prejudice', 'prejudiced', 'prejudices',
'prejudicial', 'prejudicing', 'premature', 'prematurely', 'pressing', 'pretrial', 'preventing',
'prevention', 'prevents', 'problem', 'problematic', 'problematical', 'problems', 'prolong',
'prolongation', 'prolongations', 'prolonged', 'prolonging', 'prolongs', 'prone', 'prosecute',
'prosecuted', 'prosecutes', 'prosecuting', 'prosecution', 'prosecutions', 'protest', 'protested',
'protester', 'protesters', 'protesting', 'protestor', 'protestors', 'protests', 'protracted',
'protraction', 'provoke', 'provoked', 'provokes', 'provoking', 'punished', 'punishes',
'punishing', 'punishment', 'punishments', 'punitive', 'purport', 'purported', 'purportedly',
'purporting', 'purports', 'question', 'questionable', 'questionably', 'questioned',
'questioning', 'questions', 'quit', 'quitting', 'racketeer', 'racketeering', 'rationalization',
'rationalizations', 'rationalize', 'rationalized', 'rationalizes', 'rationalizing',
'reassessment', 'reassessments', 'reassign', 'reassigned', 'reassigning', 'reassignment',

'reassignments', 'reassigns', 'recall', 'recalled', 'recalling', 'recalls', 'recession',
'recessionary', 'recessions', 'reckless', 'recklessly', 'recklessness', 'redact', 'redacted',
'redacting', 'redaction', 'redactions', 'redefault', 'redefaulted', 'redefaults', 'redress',
'redressed', 'redresses', 'redressing', 'refusal', 'refusals', 'refuse', 'refused', 'refuses',
'refusing', 'reject', 'rejected', 'rejecting', 'rejection', 'rejections', 'rejects',
'relinquish', 'relinquished', 'relinquishes', 'relinquishing', 'relinquishment',
'relinquishments', 'reluctance', 'reluctant', 'renegotiate', 'renegotiated', 'renegotiates',
'renegotiating', 'renegotiation', 'renegotiations', 'renounce', 'renounced', 'renouncement',
'renouncements', 'renounces', 'renouncing', 'reparation', 'reparations', 'repossessed',
'repossesses', 'repossessing', 'repossession', 'repossessions', 'repudiate', 'repudiated',
'repudiates', 'repudiating', 'repudiation', 'repudiations', 'resign', 'resignation',
'resignations', 'resigned', 'resigning', 'resigns', 'restate', 'restated', 'restatement',
'restatements', 'restates', 'restating', 'restructure', 'restructured', 'restructures',
'restructuring', 'restructurings', 'retaliate', 'retaliated', 'retaliates', 'retaliating',
'retaliation', 'retaliations', 'retaliatory', 'retribution', 'retributions', 'revocation',
'revocations', 'revoke', 'revoked', 'revokes', 'revoking', 'ridicule', 'ridiculed', 'ridicules',
'ridiculing', 'riskier', 'riskiest', 'risky', 'sabotage', 'sacrifice', 'sacrificed',
'sacrifices', 'sacrificial', 'sacrificing', 'scandalous', 'scandals', 'scrutinize',
'scrutinized', 'scrutinizes', 'scrutinizing', 'scrutiny', 'secrecy', 'seize', 'seized', 'seizes',
'seizing', 'sentenced', 'sentencing', 'serious', 'seriously', 'seriousness', 'setback',
'setbacks', 'sever', 'severe', 'severed', 'severely', 'severities', 'severity', 'sharply',
'shocked', 'shortage', 'shortages', 'shortfall', 'shortfalls', 'shrinkage', 'shrinkages', 'shut',
'shutdown', 'shutdowns', 'shuts', 'shutting', 'slander', 'slandered', 'slanderos', 'slanders',
'slippage', 'slippages', 'slow', 'slowdown', 'slowdowns', 'slowed', 'slower', 'slowest',
'slowing', 'slowly', 'slowness', 'sluggish', 'sluggishly', 'sluggishness', 'solvencies',
'solvency', 'spam', 'spammers', 'spamming', 'staggering', 'stagnant', 'stagnate', 'stagnated',
'stagnates', 'stagnating', 'stagnation', 'standstill', 'standstills', 'stolen', 'stoppage',
'stoppages', 'stopped', 'stopping', 'stops', 'strain', 'strained', 'straining', 'strains',
'stress', 'stressed', 'stresses', 'stressful', 'stressing', 'stringent', 'strong', 'subjected',
'subjecting', 'subjection', 'subpoena', 'subpoenaed', 'subpoenas', 'substandard', 'sue', 'sued',
'sues', 'suffer', 'suffered', 'suffering', 'suffers', 'suing', 'summoned', 'summoning',
'summons', 'summonses', 'susceptibility', 'susceptible', 'suspect', 'suspected', 'suspects',
'suspend', 'suspended', 'suspending', 'suspends', 'suspension', 'suspensions', 'suspicion',
'suspicious', 'suspiciously', 'taint', 'tainted', 'tainting', 'taints', 'tampered',
'tense', 'terminate', 'terminated', 'terminates', 'terminating', 'termination', 'terminations',
'testify', 'testifying', 'threat', 'threaten', 'threatened', 'threatening', 'threatens',
'threats', 'tightening', 'tolerate', 'tolerated', 'tolerates', 'tolerating', 'toleration',
'tortuous', 'tortuously', 'tragedies', 'tragedy', 'tragic', 'tragically', 'traumatic', 'trouble',
'troubled', 'troubles', 'turbulence', 'turmoil', 'unable', 'unacceptable', 'unacceptably',
'unaccounted', 'unannounced', 'unanticipated', 'unapproved', 'unattractive', 'unauthorized',
'unavailability', 'unavailable', 'unavoidable', 'unavoidably', 'unaware', 'uncollectable',
'uncollected', 'uncollectibility', 'uncollectible', 'uncollectibles', 'uncompetitive',
'uncompleted', 'unconscionable', 'unconscionably', 'uncontrollable', 'uncontrollably',

'uncontrolled', 'uncorrected', 'uncover', 'uncovered', 'uncovering', 'uncovers', 'undeliverable',
'undelivered', 'undercapitalized', 'undercut', 'undercuts', 'undercutting', 'underestimate',
'underestimated', 'underestimates', 'underestimating', 'underestimation', 'underfunded',
'underinsured', 'undermine', 'undermined', 'undermines', 'undermining', 'underpaid',
'underpayment', 'underpayments', 'underpays', 'underperform', 'underperformance',
'underperformed', 'underperforming', 'underperforms', 'underproduced', 'underproduction',
'underreporting', 'understate', 'understated', 'understatement', 'understatements',
'understates', 'understating', 'underutilization', 'underutilized', 'undesirable', 'undesired',
'undetected', 'undetermined', 'undisclosed', 'undocumented', 'undue', 'unduly', 'uneconomic',
'uneconomical', 'uneconomically', 'unemployed', 'unemployment', 'unethical', 'unethically',
'unexcused', 'unexpected', 'unexpectedly', 'unfair', 'unfairly', 'unfavorability', 'unfavorable',
'unfavorably', 'unfavourable', 'unfeasible', 'unfit', 'unfitness', 'unforeseeable', 'unforeseen',
'unforseen', 'unfortunate', 'unfortunately', 'unfounded', 'unfriendly', 'unfulfilled',
'unfunded', 'uninsured', 'unintended', 'unintentional', 'unintentionally', 'unjust',
'unjustifiable', 'unjustifiably', 'unjustified', 'unjustly', 'unknowing', 'unknowingly',
'unlawful', 'unlawfully', 'unlicensed', 'unliquidated', 'unmarketable', 'unmerchantable',
'unmeritorious', 'unnecessarily', 'unnecessary', 'unneeded', 'unobtainable', 'unoccupied',
'unpaid', 'unperformed', 'unplanned', 'unpopular', 'unpredictability', 'unpredictable',
'unpredictably', 'unpredicted', 'unproductive', 'unprofitability', 'unprofitable', 'unqualified',
'unrealistic', 'unreasonable', 'unreasonableness', 'unreasonably', 'unreceptive',
'unrecoverable', 'unrecovered', 'unreimbursed', 'unreliable', 'unremedied', 'unreported',
'unresolved', 'unrest', 'unsafe', 'unsalable', 'unsaleable', 'unsatisfactory', 'unsatisfied',
'unsavory', 'unscheduled', 'unsellable', 'unsold', 'unsound', 'unstabilized', 'unstable',
'unsubstantiated', 'unsuccessful', 'unsuccessfully', 'unsuitability', 'unsuitable', 'unsuitably',
'unsuited', 'unsure', 'unsuspected', 'unsuspecting', 'unsustainable', 'untenable', 'untimely',
'untrusted', 'untruth', 'untruthful', 'untruthfully', 'untruthfulness', 'untruths', 'unusable',
'unwanted', 'unwarranted', 'unwelcome', 'unwilling', 'unwillingness', 'upset', 'urgency',
'urgent', 'usurious', 'usurp', 'usurped', 'usurping', 'usurps', 'usury', 'vandalism', 'verdict',
'verdicts', 'vetoed', 'victims', 'violate', 'violated', 'violates', 'violating', 'violation',
'violations', 'violative', 'violation', 'violators', 'violence', 'violent', 'violently',
'vitiate', 'vitiates', 'vitiating', 'vitiating', 'vitiating', 'vitiating', 'voided', 'voiding', 'volatile',
'volatility', 'vulnerabilities', 'vulnerability', 'vulnerable', 'vulnerably', 'warn', 'warned',
'warning', 'warnings', 'warns', 'wasted', 'wasteful', 'wasting', 'weak', 'weaken', 'weakened',
'weakening', 'weakens', 'weaker', 'weakest', 'weakly', 'weakness', 'weaknesses', 'willfully',
'worries', 'worry', 'worrying', 'worse', 'worsen', 'worsened', 'worsening', 'worsens', 'worst',
'worthless', 'writedown', 'writedowns', 'writeoff', 'writeoffs', 'wrong', 'wrongdoing',
'wrongdoings', 'wrongful', 'wrongfully', 'wrongly',
'negative', 'negatives', 'fail', 'fails', 'failing', 'failure', 'weak', 'weakness', 'weaknesses',
'difficult', 'difficulty', 'hurdle', 'hurdles', 'obstacle', 'obstacles', 'slump', 'slumps',
'slumping', 'slumped', 'uncertain', 'uncertainty', 'unsettled', 'unfavorable', 'downturn',
'depressed', 'disappoint', 'disappoints', 'disappointing', 'disappointed', 'disappointment',
'risk', 'risks', 'risky', 'threat', 'threats', 'penalty', 'penalties', 'down', 'decrease',
'decreases', 'decreasing', 'decreased', 'decline', 'declines', 'declining', 'declined', 'fall',
'falls', 'falling', 'fall', 'fallen', 'drop', 'drops', 'dropping', 'dropped', 'deteriorate'

falls', 'falling', 'fell', 'fallen', 'drop', 'drops', 'dropping', 'dropped', 'deteriorate',
'deteriorates', 'deteriorating', 'deteriorated', 'worsen', 'worsens', 'worsening', 'weaken',
'weakens', 'weakening', 'weakened', 'worse', 'worst', 'low', 'lower', 'lowest', 'less', 'least',
'smaller', 'smallest', 'shrink', 'shrinks', 'shrinking', 'shrunk', 'below', 'under', 'challenge',
'challenges', 'challenging', 'challenged'

],

'Positive': ['able', 'abundance', 'abundant', 'acclaimed', 'accomplish', 'accomplished', 'accomplishes',
'accomplishing', 'accomplishment', 'accomplishments', 'achieve', 'achieved', 'achievement',
'achievements', 'achieves', 'achieving', 'adequately', 'advancement', 'advancements', 'advances',
'advancing', 'advantage', 'advantaged', 'advantageous', 'advantageously', 'advantages',
'alliance', 'alliances', 'assure', 'assured', 'assures', 'assuring', 'attain', 'attained',
'attaining', 'attainment', 'attainments', 'attains', 'attractive', 'attractiveness', 'beautiful',
'beautifully', 'beneficial', 'beneficially', 'benefit', 'benefited', 'benefiting', 'benefitted',
'benefitting', 'best', 'better', 'bolstered', 'bolstering', 'bolsters', 'boom', 'booming',
'boost', 'boosted', 'breakthrough', 'breakthroughs', 'brilliant', 'charitable', 'collaborate',
'collaborated', 'collaborates', 'collaborating', 'collaboration', 'collaborations',
'collaborative', 'collaborator', 'collaborators', 'compliment', 'complimentary', 'complimented',
'complimenting', 'compliments', 'conclusive', 'conclusively', 'conducive', 'confident',
'constructive', 'constructively', 'courteous', 'creative', 'creatively', 'creativeness',
'creativity', 'delight', 'delighted', 'delightful', 'delightfully', 'delighting', 'delights',
'dependability', 'dependable', 'desirable', 'desired', 'despite', 'destined', 'diligent',
'diligently', 'distinction', 'distinctions', 'distinctive', 'distinctively', 'distinctiveness',
'dream', 'easier', 'easily', 'easy', 'effective', 'efficiencies', 'efficiency', 'efficient',
'efficiently', 'empower', 'empowered', 'empowering', 'empowers', 'enable', 'enabled', 'enables',
'enabling', 'encouraged', 'encouragement', 'encourages', 'encouraging', 'enhance', 'enhanced',
'enhancement', 'enhancements', 'enhances', 'enhancing', 'enjoy', 'enjoyable', 'enjoyably',
'enjoyed', 'enjoying', 'enjoyment', 'enjoys', 'enthusiasm', 'enthusiastic', 'enthusiastically',
'excellence', 'excellent', 'excelling', 'excels', 'exceptional', 'exceptionally', 'excited',
'excitement', 'exciting', 'exclusive', 'exclusively', 'exclusiveness', 'exclusives',
'exclusivity', 'exemplary', 'fantastic', 'favorable', 'favorably', 'favored', 'favoring',
'favorite', 'favorites', 'friendly', 'gain', 'gained', 'gaining', 'gains', 'good', 'great',
'greater', 'greatest', 'greatly', 'greatness', 'happiest', 'happily', 'happiness', 'happy',
'highest', 'honor', 'honorable', 'honored', 'honoring', 'honors', 'ideal', 'impress',
'impressed', 'impresses', 'impressing', 'impressive', 'impressively', 'improve', 'improved',
'improvement', 'improvements', 'improves', 'improving', 'incredible', 'incredibly',
'influential', 'informative', 'ingenuity', 'innovate', 'innovated', 'innovates', 'innovating',
'innovation', 'innovations', 'innovative', 'innovativeness', 'innovator', 'innovators',
'insightful', 'inspiration', 'inspirational', 'integrity', 'invent', 'invented', 'inventing',
'invention', 'inventions', 'inventive', 'inventiveness', 'inventor', 'inventors', 'leadership',
'leading', 'loyal', 'lucrative', 'meritorious', 'opportunities', 'opportunity', 'optimistic',
'outperform', 'outperformed', 'outperforming', 'outperforms', 'perfect', 'perfected',
'perfectly', 'perfects', 'pleasant', 'pleasantly', 'pleased', 'pleasure', 'plentiful', 'popular',
'popularity', 'positive', 'positively', 'preeminence', 'preeminent', 'premier', 'premiere',
'prestige', 'prestigious', 'proactive', 'proactively', 'proficiency', 'proficient',

'proficiently', 'profitability', 'profitable', 'profitably', 'progress', 'progressed',
'progresses', 'progressing', 'prospered', 'prospering', 'prosperity', 'prosperous', 'prosper',
'rebound', 'rebounded', 'rebounding', 'receptive', 'regain', 'regained', 'regaining', 'resolve',
'revolutionize', 'revolutionized', 'revolutionizes', 'revolutionizing', 'reward', 'rewarded',
'rewarding', 'rewards', 'satisfaction', 'satisfactorily', 'satisfactory', 'satisfied',
'satisfies', 'satisfy', 'satisfying', 'smooth', 'smoothing', 'smoothly', 'smooths', 'solves',
'solving', 'spectacular', 'spectacularly', 'stability', 'stabilization', 'stabilizations',
'stabilize', 'stabilized', 'stabilizes', 'stabilizing', 'stable', 'strength', 'strengthen',
'strengthened', 'strengthening', 'strengthens', 'strengths', 'strong', 'stronger', 'strongest',
'succeed', 'succeeded', 'succeeding', 'succeeds', 'success', 'successes', 'successful',
'successfully', 'superior', 'surpass', 'surpassed', 'surpasses', 'surpassing', "sustainable", 'transparency',
'tremendous', 'tremendously', 'unmatched', 'unparalleled', 'unsurpassed', 'upturn', 'upturns',
'valuable', 'versatile', 'versatility', 'vibrancy', 'vibrant', 'win', 'winner', 'winners', 'winning', 'worthy',
'positive', 'positives', 'success', 'successes', 'successful', 'succeed', 'succeeds',
'succeeding', 'succeeded', 'accomplish', 'accomplishes', 'accomplishing', 'accomplished',
'accomplishment', 'accomplishments', 'strong', 'strength', 'strengths', 'certain', 'certainty',
'definite', 'solid', 'excellent', 'good', 'leading', 'achieve', 'achieves', 'achieved',
'achieving', 'achievement', 'achievements', 'progress', 'progressing', 'deliver', 'delivers',
'delivered', 'delivering', 'leader', 'leading', 'pleased', 'reward', 'rewards', 'rewarding',
'rewarded', 'opportunity', 'opportunities', 'enjoy', 'enjoys', 'enjoying', 'enjoyed',
'encouraged', 'encouraging', 'up', 'increase', 'increases', 'increasing', 'increased', 'rise',
'rises', 'rising', 'rose', 'risen', 'improve', 'improves', 'improving', 'improved', 'improvement',
'improvements', 'strengthen', 'strengthens', 'strengthening', 'strengthened', 'stronger',
'strongest', 'better', 'best', 'more', 'most', 'above', 'record', 'high', 'higher', 'highest',
'greater', 'greatest', 'larger', 'largest', 'grow', 'grows', 'growing', 'grew', 'grown', 'growth',
'expand', 'expands', 'expanding', 'expanded', 'expansion', 'exceed', 'exceeds', 'exceeded',
'exceeding', 'beat', 'beats', 'beating']

}

negate = ["aint", "arent", "cannot", "cant", "couldnt", "darent", "didnt", "doesnt", "ain't", "aren't", "can't",
"couldn't", "daren't", "didn't", "doesn't", "dont", "hadnt", "hasnt", "havent", "isnt", "mightnt", "mustnt",
"neither", "don't", "hadn't", "hasn't", "haven't", "isn't", "mightn't", "mustn't", "neednt", "needn't",
"never", "none", "nope", "nor", "not", "nothing", "nowhere", "oughtnt", "shant", "shouldnt", "wasnt",
"werent", "oughtn't", "shan't", "shouldn't", "wasn't", "weren't", "without", "wont", "wouldnt", "won't",
"wouldn't", "rarely", "seldom", "despite", "no", "nobody"]

```
def negated(word):
    """
    Determine if preceding word is a negation word
    """
    if word.lower() in negate:
        return True
    ,
```

```
else:
    return False
```

```
def tone_count_with_negation_check(dict, article):
    """
    Count positive and negative words with negation check. Account for simple negation only for positive words.
    Simple negation is taken to be observations of one of negate words occurring within three words
    preceding a positive words.
    """
    pos_count = 0
    neg_count = 0

    pos_words = []
    neg_words = []

    input_words = re.findall(r'\b([a-zA-Z]+n?t|[a-zA-Z]+\s|[a-zA-Z]+)\b', article.lower())

    word_count = len(input_words)

    for i in range(0, word_count):
        if input_words[i] in dict['Negative']:
            neg_count += 1
            neg_words.append(input_words[i])
        if input_words[i] in dict['Positive']:
            if i >= 3:
                if negated(input_words[i - 1]) or negated(input_words[i - 2]) or negated(input_words[i - 3]):
                    neg_count += 1
                    neg_words.append(input_words[i] + ' (with negation)')
                else:
                    pos_count += 1
                    pos_words.append(input_words[i])
            elif i == 2:
                if negated(input_words[i - 1]) or negated(input_words[i - 2]):
                    neg_count += 1
                    neg_words.append(input_words[i] + ' (with negation)')
                else:
                    pos_count += 1
                    pos_words.append(input_words[i])
            elif i == 1:
                if negated(input_words[i - 1]):
                    neg_count += 1
                    neg_words.append(input_words[i] + ' (with negation)')
                else:
```

```

        pos_count += 1
        pos_words.append(input_words[i])
    elif i == 0:
        pos_count += 1
        pos_words.append(input_words[i])

results = [word_count, pos_count, neg_count, pos_words, neg_words]

return results

print(len(Data))
temp = [tone_count_with_negation_check(lmdict,x) for x in Data.text]
temp = pd.DataFrame(temp)

Data['wordcount'] = temp.iloc[:,0].values
Data['NPositiveWords'] = temp.iloc[:,1].values
Data['NNegativeWords'] = temp.iloc[:,2].values

# Sentiment Score normalized by the number of words
Data['sentiment'] = (Data['NPositiveWords'] - Data['NNegativeWords']) / Data['wordcount'] * 100
Data['Poswords'] = temp.iloc[:,3].values
Data['Negwords'] = temp.iloc[:,4].values

```

1500

temp.head()

	0	1	2	3	4
0	3209	98	149	[honored, delighted, opportunity, more, great,...	[recall, difficult, adverse, negative, risk, c...
1	2692	52	54	[pleasure, progress, more, most, innovation, g...	[slow, disappear, slowly, succeed (with negati...
2	2900	102	111	[achieving, stability, delivering, good, good,...	[challenges, defend, stops, deficit, overstate...
3	1986	46	56	[versatile, high, popular, more, innovation, m...	[challenge, low, premature, crisis, challenge,...
4	2177	70	41	[pleasure, more, better, more, larger, profita...	[question, concern, less, ill, force, force, b...

Data.head()

	contents	speaker	text	Index	wordcount	NPositiveWords	NNegativeWords	sentiment	Poswords	Negword:
date										
1996-06-13	Remarks by Chairman Alan Greenspan\nBank super...	Chairman Alan Greenspan	Remarks by Chairman Alan Greenspan Bank superv...	0	3209	98	149	-1.589280	[honored, delighted, opportunity, more, great,...	[recall difficult adverse negative risk, c...
1996-06-18	Remarks by Governor Edward W. Kelley, Jr.\nDev...	Governor Edward W. Kelley, Jr.	Remarks by Governor Edward W. Kelley, Jr. Deve...	1	2692	52	54	-0.074294	[pleasure, progress, more, most, innovation, g...	[slow disappear slowly succeed (with negative...
1996-09-08	Monetary Policy Objectives and Strategy\n\n[SE...	Governor Laurence H. Meyer	Monetary Policy Objectives and StrategyI want ...	2	2900	102	111	-0.310345	[achieving, stability, delivering, good, good,...	[challenges defend stops deficit overstate...
1996-09-19	Remarks by Chairman Alan Greenspan\nRegulation...	Chairman Alan Greenspan	Remarks by Chairman Alan Greenspan Regulation ...	3	1986	46	56	-0.503525	[versatile, high, popular, more, innovation, m...	[challenge low premature crisis challenge,...
1996-	Remarks by Governor	Governor	Remarks by Governor						[pleasure, more, better	[question concern

▼ Plots of the sentiment analysis

Plot positive and negative word counts

```
NetSentiment = Data['NPositiveWords'] - Data['NNegativeWords']
```

```
fig, ax = plt.subplots(figsize=(15,7))
```

```
ax.plot(Data.index, Data['NPositiveWords'], c='green', linewidth= 1.0)
plt.plot(Data.index, Data['NNegativeWords']*-1, c='red', linewidth=1.0)
plt.plot(Data.index, NetSentiment, c='grey', linewidth=1.0)

plt.title('The number of positive/negative words in statement', fontsize=16)
plt.legend(['Positive Words', 'Negative Words', 'Net Sentiment'], prop={'size': 14}, loc=1)

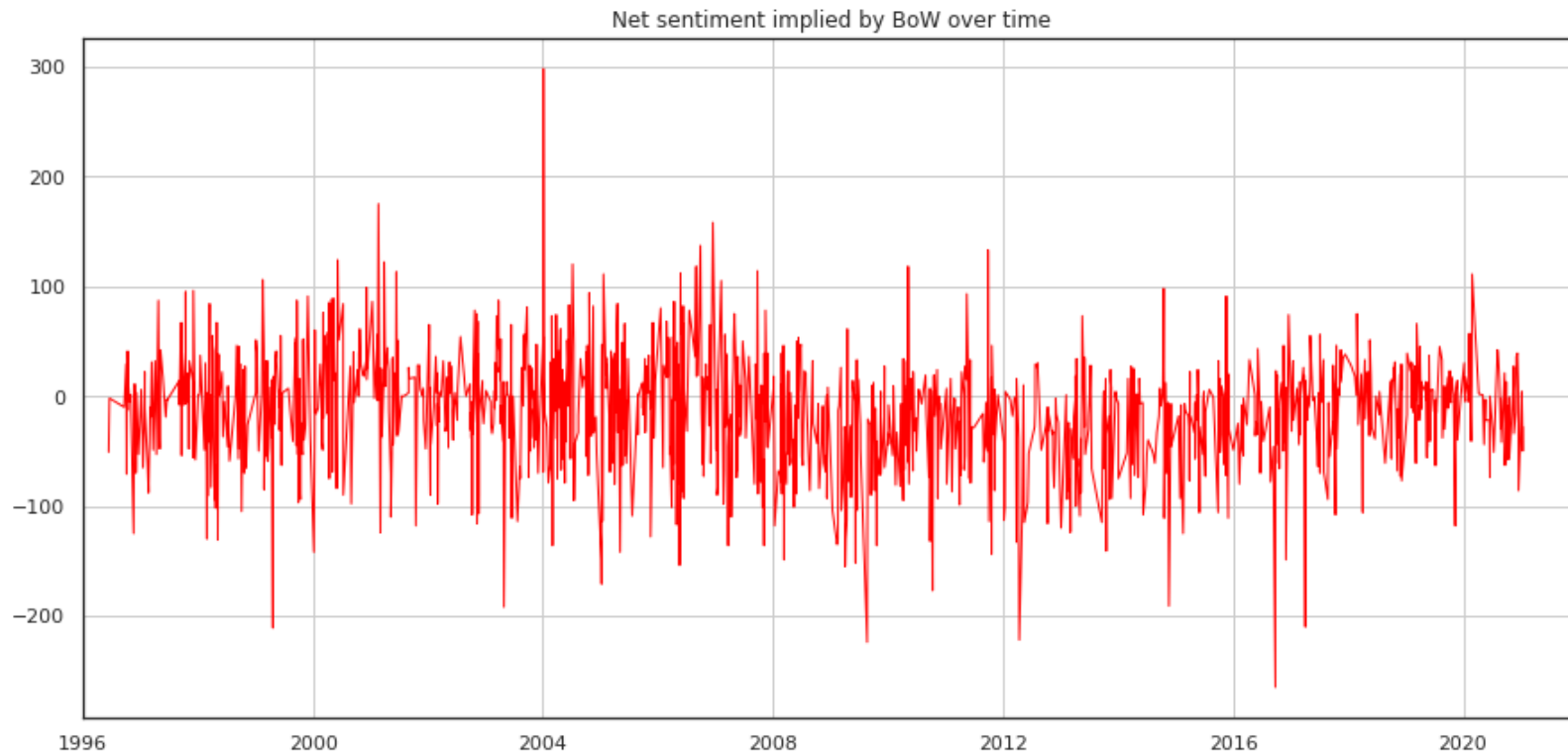
ax.fill_between(Data.index, NetSentiment, where=(NetSentiment > 0), color='green', alpha=0.3,
               interpolate=True)
ax.fill_between(Data.index, NetSentiment, where=(NetSentiment <= 0), color='red', alpha=0.3,
               interpolate=True)

## format the ticks
# round to nearest years.
import matplotlib.dates as mdates
years = mdates.YearLocator()    # every year
months = mdates.MonthLocator()  # every month
years_fmt = mdates.DateFormatter('%Y')

datemin = np.datetime64(Data.index[0], 'Y')
datemax = np.datetime64(Data.index[-1], 'Y') + np.timedelta64(1, 'Y')
ax.set_xlim(datemin, datemax)

# format the coords message box
ax.format_xdata = mdates.DateFormatter('%Y-%m-%d')
ax.grid(True)

plt.show()
plt.savefig(graph_dir + 'sentiment_pos_neg_word_count_per_type' + '.png')#bbox_inches='tight')
```

Plot derivative to see the changes in net sentiment

```
firstderivative = (NetSentiment.shift(1) - NetSentiment) / NetSentiment

fig, ax = plt.subplots(figsize=(15,7))

ax.plot(Data.index, firstderivative,
        c = 'red')

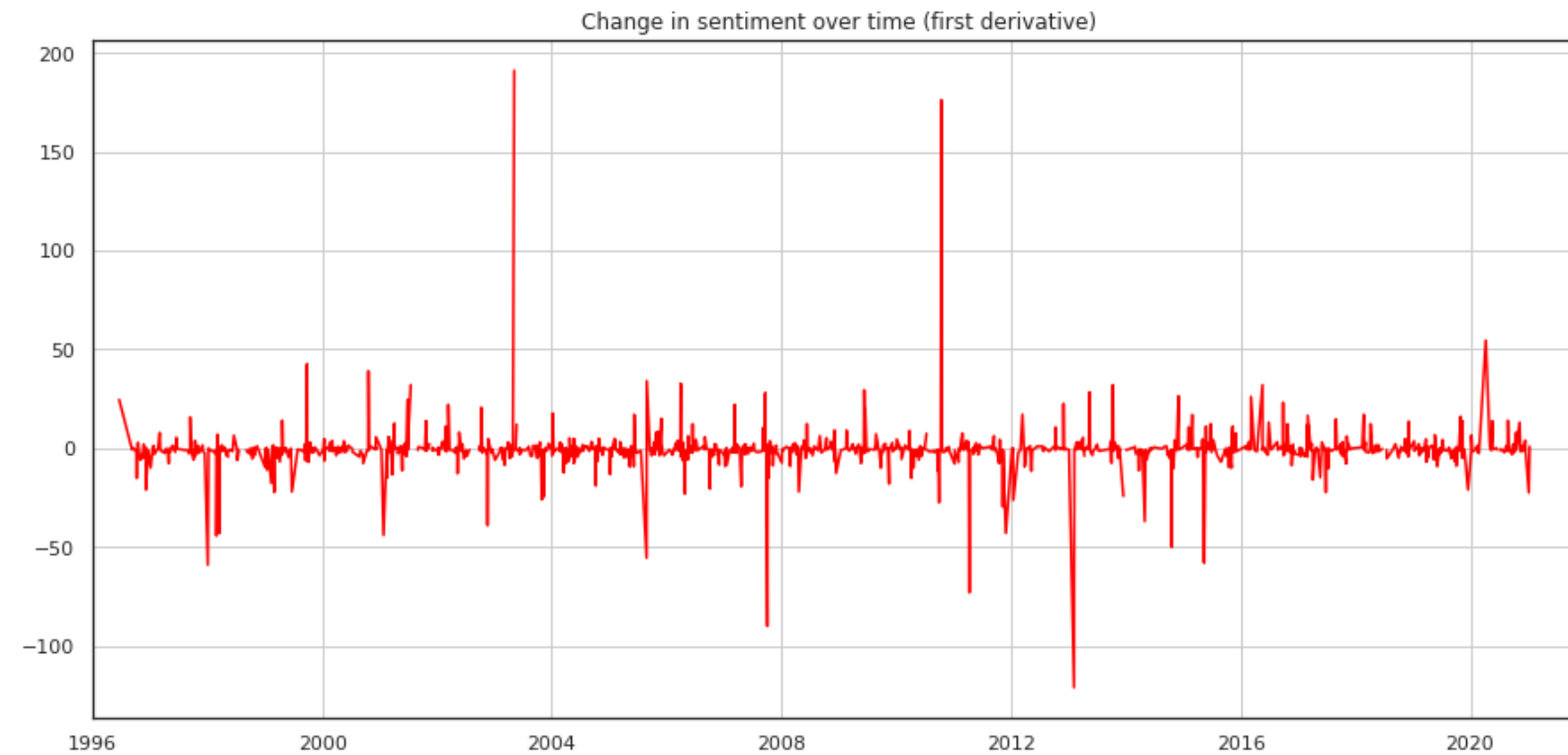
plt.title('Change in sentiment over time (first derivative)')

# format the ticks
# round to nearest years.
datemin = np.datetime64(Data.index[0], 'Y')
datemax = np.datetime64(Data.index[-1], 'Y') + np.timedelta64(1, 'Y')
ax.set_xlim(datemin, datemax)

# format the coords message box
ax.format_xdata = mdates.DateFormatter('%Y-%m-%d')
```

```
ax.grid(True)

plt.show()
plt.savefig(graph_dir + 'sentiment_pct_change_vs_time' + '.png')#bbox_inches='tight')
```



<Figure size 432x288 with 0 Axes>

```
# Normalize data
NPositiveWordsNorm = Data['NPositiveWords'] / Data['wordcount'] * np.mean(Data['wordcount'])
NNegativeWordsNorm = Data['NNegativeWords'] / Data['wordcount'] * np.mean(Data['wordcount'])
NetSentimentNorm = (NPositiveWordsNorm - NNegativeWordsNorm)

fig, ax = plt.subplots(figsize=(15,7))
ax.plot(Data.index, NPositiveWordsNorm, c='green', linewidth= 1.0)
plt.plot(Data.index, NNegativeWordsNorm, c='red', linewidth=1.0)

plt.title('Counts normalized by the number of words', fontsize=16)
plt.legend(['Count of Positive Words', 'Count of Negative Words'],
           prop={'size': 14},
```



```

        loc = 1
    )

# format the ticks
# round to nearest years.
import matplotlib.dates as mdates
years = mdates.YearLocator()    # every year
months = mdates.MonthLocator() # every month
years_fmt = mdates.DateFormatter('%Y')

datemin = np.datetime64(Data.index[0], 'Y')
datemax = np.datetime64(Data.index[-1], 'Y') + np.timedelta64(1, 'Y')
ax.set_xlim(datemin, datemax)

# format the coords message box
ax.format_xdata = mdates.DateFormatter('%Y-%m-%d')
ax.grid(True)

plt.show()
plt.savefig(graph_dir + 'counts_normby_num_words' + '.png')#bbox_inches='tight')

fig, ax = plt.subplots(figsize=(15,7))
ax.plot(Data.index, NetSentimentNorm, c='red', linewidth=1.0)

plt.title('Net sentiment implied by BoW over time',size = 'medium')

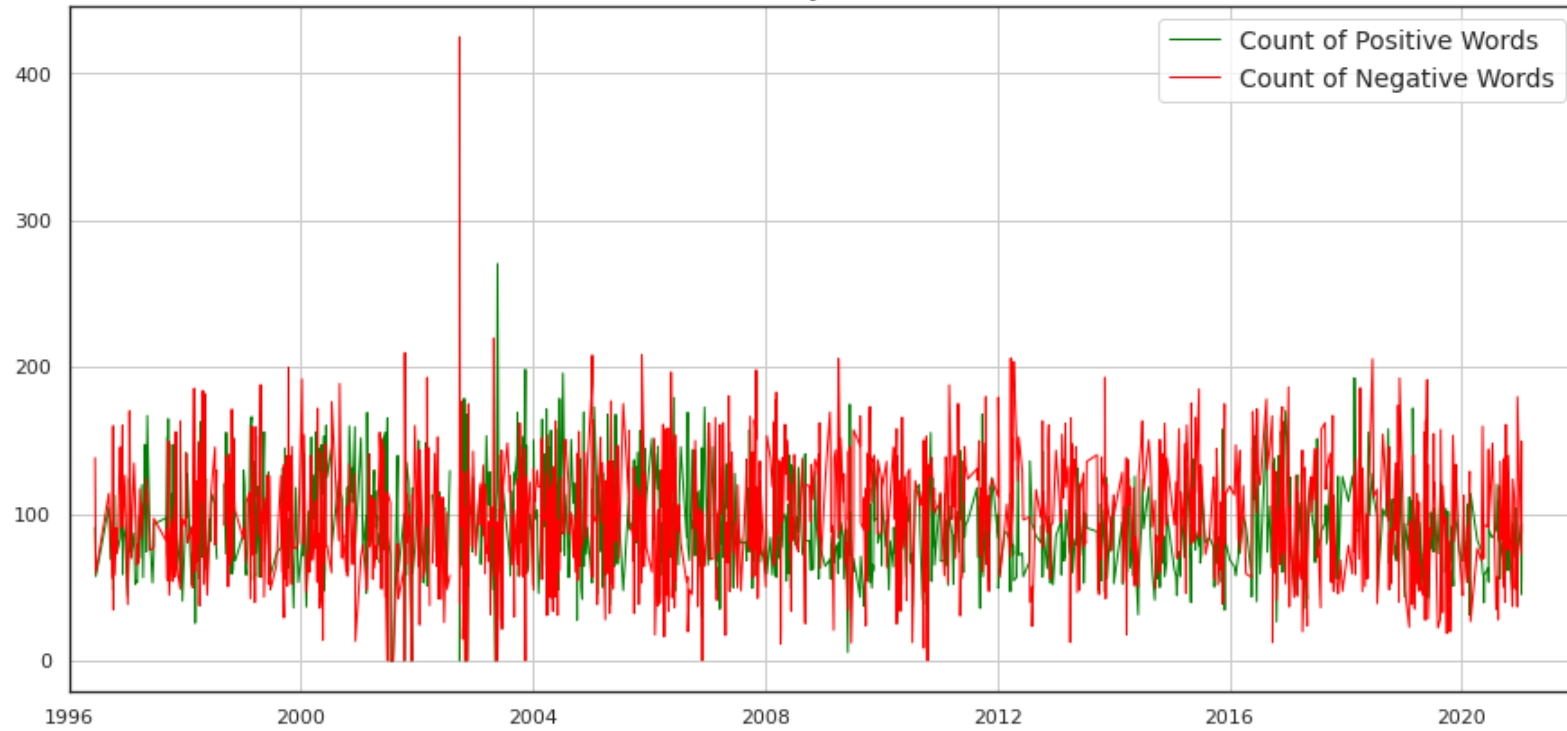
# format the ticks
# round to nearest years.
datemin = np.datetime64(Data.index[0], 'Y')
datemax = np.datetime64(Data.index[-1], 'Y') + np.timedelta64(1, 'Y')
ax.set_xlim(datemin, datemax)

# format the coords message box
ax.format_xdata = mdates.DateFormatter('%Y-%m-%d')
ax.grid(True)

plt.show()
plt.savefig(graph_dir + 'net_sentiment_bow_vs_time' + '.png')#bbox_inches='tight')

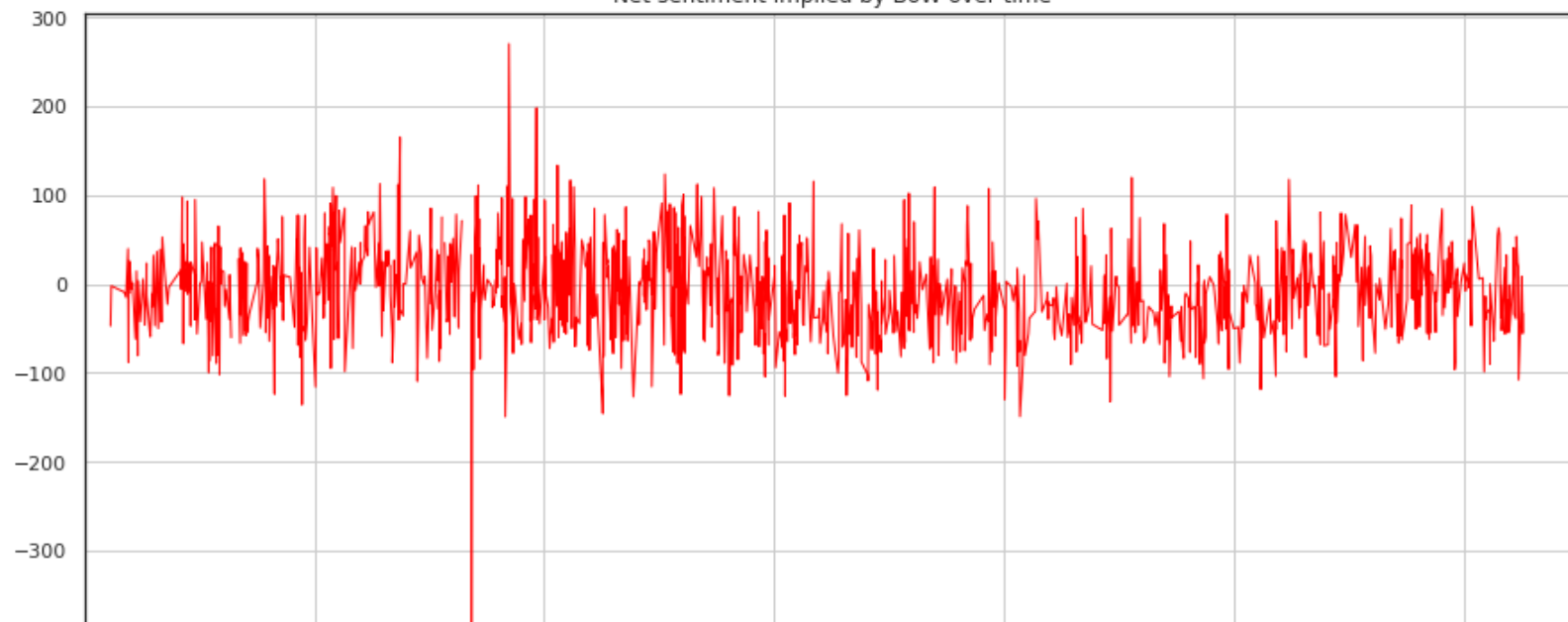
```

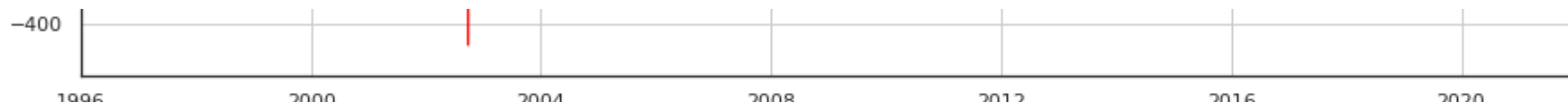
Counts normalized by the number of words



<Figure size 432x288 with 0 Axes>

Net sentiment implied by BoW over time





▼ Function for extracting the direction of the rate change (hike, keep, lower)

Gets Basis Points Move and End Rate Value for Fed Rate and Discount Rate as well as Preferred Fed Rate Move from Statement Text

```
import nltk.data
tokenizer = nltk.data.load('tokenizers/punkt/english.pickle')

def ExtractKeyValues(text):
    '''
    First, the text is tokenized in to sentences and the fed funds target rate sentence is extracted
    Secondly, the decision whether the fed target rate was hiked, kept or lowered is retrieved to RateHike variable
    Third, the target rate range is extracted and saved in TargetRange (The max is extracted as it conveys most info)
    '''

    Potentials = [] #For storing the potential decision sentences

    #Remove the \n and replace them with dots
    text = text.replace('\n', '. ')

    #Split the text in to a list of sentences
    paragraph_sentence_list = tokenizer.tokenize(text)
    #print('paragraph_sentence_list: ', paragraph_sentence_list)

    #Finds the sentences with the decisions
    for Sentence in paragraph_sentence_list:

        #print('Sentence: ', Sentence)
        #Make the text lower case
        Sentence = Sentence.lower()

        if 'federal funds ' in Sentence and 'target' in Sentence:

            #Check if it contains a number
            if bool(re.search(r'\d', Sentence)):

                Potentials.append(Sentence)
```

```

Potentials = ''.join(Potentials)
#print('Potentials: ', Potentials)

#Look for decision words related to raising rates
r = Potentials.find('rais')
i = Potentials.find('increas')
upscore = max([r,i])

#Look for decision words related to keeping rates
k = Potentials.find('keep')
m = Potentials.find('maintain')
r = Potentials.find('remain')
c = Potentials.find('current')
keepscore = max([r,k,m,c])

#Look for decision words related to decreasing rates
l = Potentials.find('lower')
d = Potentials.find('decreas')
lowerscore = max([l,d])

Summary = {1:upscore,
           0: keepscore,
          -1:lowerscore
           }

#take the key that was mentioned first in the text
RateHike = max(Summary.keys(), key=(lambda x: Summary[x]))

#For extracting the target range
def RangeReader(Nums):

    def FractionReader(Fraction):
        '''
        Converts the fraction to a float
        '''
        i = 0
        if '/' in Fraction:
            i, Fraction = Fraction.split(' ')
        if '-' in Fraction:
            i, Fraction = Fraction.split('-')
        if '-' in Fraction:

```

```

        i, Fraction = Fraction.split('-')
        N, D = Fraction.split('/')
        return float(i) + float(N) / float(D)

    else:
        return float(i)+float(Fraction)

#Splits the range based on to or -
Splitters = [' to ', '-']

for Splitter in Splitters:
    if Splitter in Nums:
        TargetRange = Nums.split(Splitter)
        return FractionReader(TargetRange[0]), FractionReader(TargetRange[1])

#If there was no splitter in the range (no range)
return FractionReader(Nums)

#Find the percentage change and take the closest digits
Rate = Potentials[ : Potentials.find('percent') -1 ]
Rate = Rate[-16:]
# print(Rate)
if re.search("\d", Rate):
    Rate = Rate[ re.search("\d", Rate).start() : ]
    TargetRange = np.max(RangeReader(Rate))
else:
    Rate = None
    TargetRange = None

return RateHike, TargetRange

```

Appends decision, dummy and Moves and End Rate Values for Fed and Disc Rate also appends preferred fed rate Move while measuring the speed of the extraction

```

Data.tail()
#Data.drop(index=[pd.to_datetime("2020-03-03")], inplace=True)

```

	contents	speaker	text	Index	wordcount	NPositiveWords	NNegativeWords	sentiment	Poswords	Negwords
date										
2020-12-18	The Federal Reserve, the central bank of the U...	Governor Lael Brainard	The Federal Reserve, the central bank of the U...	1495	2368	57	143	-3.631757	[stable, progress, progress, sustainable, stre...	[imperative, risks, challenge, challenge, chal...
2021-01-08	The Federal Reserve, the central bank of the U...	Vice Chair Richard H. Clarida	The Federal Reserve, the central bank of the U...	1496	1382	43	39	0.289436	[stable, pleasure, delivered, most, great, reb...	[questions, severe, depressed, strong, unemplo...
2021-01-12	The Federal Reserve, the central bank of the U...	Governor Lael Brainard	The Federal Reserve, the central bank of the U...	1497	2892	44	71	-0.933610	[stable, improving, enable, more, growing, inc...	[risks, challenges, risk, risk, fraud, losses,...
2021-01-13	The Federal Reserve, the central bank of the U...	Governor Lael Brainard	The Federal Reserve, the central bank of the U...	1498	2669	84	134	-1.873361	[stable, pleasure, honored, deliver, best, mos...	[critical, lower, unemployment, critically, da...
	The Federal	Vice Chair	The Federal						[stable, achieve	[poorly, argue,

Tried to take rate and decision from the statement text but it sometimes fails. Debugging takes time, so switch to use data instead of reading from text.

```
# import timeit

# start = timeit.timeit()

# #Append fed funds rates and decisions to dataframe
# Hikes = []
# TargetRate = []

# for i in range(len(Data)):
#     if Data.iloc[i,1] == True:
#         Hikes.append(np.nan)
#         TargetRate.append(0.025)
```

```

#         TargetRate.append(np.nan)
#     else:
#         Hikes.append( ExtractKeyValues( Data.iloc[i,0] )[0] )
#         TargetRate.append( ExtractKeyValues( Data.iloc[i,0] )[1] )

# Data['RateDecision'] = Hikes
# Data['FedFundRateEndValue'] = TargetRate

# end = timeit.timeit()

# print (str ( end - start ) + ' seconds elapsed for ' + str(i) + ' statements')

# Data.tail()

fedtar = pd.read_csv(fed_rates_dir + 'DFEDTAR.csv', names=['Date', 'Rate'], header=0)
fedtaru = pd.read_csv(fed_rates_dir + 'DFEDTARU.csv', names=['Date', 'Rate'], header=0)
fedtarl = pd.read_csv(fed_rates_dir + 'DFEDTARL.csv', names=['Date', 'Rate'], header=0)

fedrate_df = pd.concat([fedtar, fedtaru], axis=0)
fedrate_df.index = pd.to_datetime(fedrate_df.Date, format="%Y-%m-%d")
fedrate_df.drop(columns=['Date'], inplace=True)
fedrate_df['Rate'] = fedrate_df['Rate'].map(lambda x: np.float(x))
fedrate_df

```

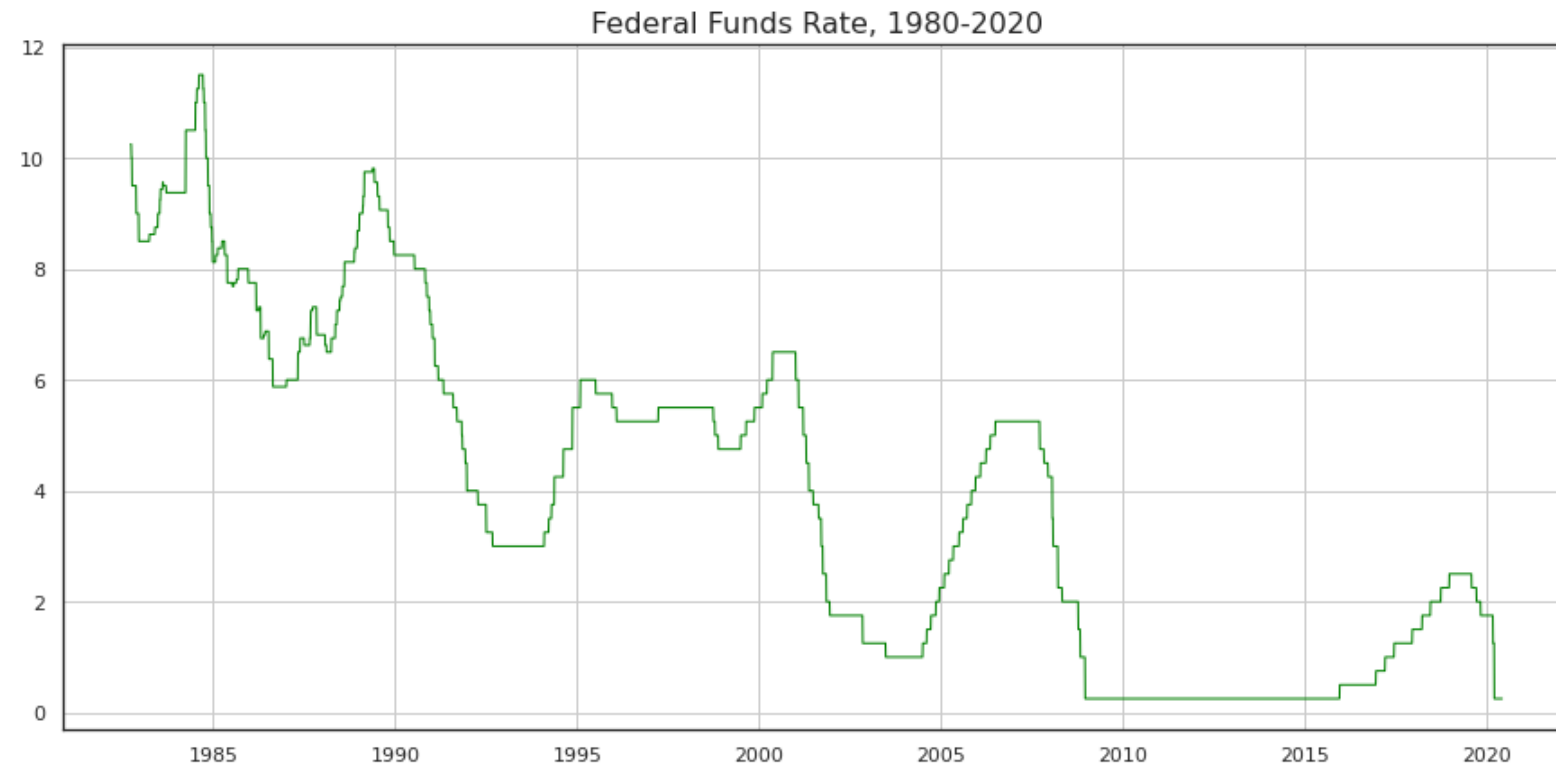
Rate

Date

1982-09-27 10.25

```
fig, ax = plt.subplots(figsize=(15,7))
ax.plot(fedrate_df.index, fedrate_df['Rate'].values,
        c = 'green',
        linewidth= 1.0)
ax.grid(True)
plt.title('Federal Funds Rate, 1980-2020', fontsize=16)

plt.show()
plt.savefig(graph_dir + 'fed_rate_vs_time' + '.png')#bbox_inches='tight')
```



<Figure size 432x288 with 0 Axes>

```
Data['RateDecision'] = None
Data['Rate'] = None
```



```
Data['Rate'] = None
```

```
for i in range(len(Data)):
    for j in range(len(fedrate_df)):
        if Data.index[i] == fedrate_df.index[j]:
            Data['Rate'][i] = float(fedrate_df['Rate'][j+1])
            if fedrate_df['Rate'][j-1] == fedrate_df['Rate'][j+1]:
                Data['RateDecision'][i] = 0
            elif fedrate_df['Rate'][j-1] < fedrate_df['Rate'][j+1]:
                Data['RateDecision'][i] = 1
            elif fedrate_df['Rate'][j-1] > fedrate_df['Rate'][j+1]:
                Data['RateDecision'][i] = -1
```

```
Data.head(10)
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
import sys
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
if __name__ == '__main__':
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:13: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
del sys.path[0]
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

	contents	speaker	text	Index	wordcount	NPositiveWords	NNegativeWords	sentiment	Poswords	Negw
date										
1996-06-13	Remarks by Chairman Alan Greenspan\nBank super...	Chairman Alan Greenspan	Remarks by Chairman Alan Greenspan Bank superv...	0	3209	98	149	-1.589280	[honored, delighted, opportunity, more, great,...	[r dif adv negative
1996-06-18	Remarks by Governor Edward W. Kelley, Jr.\nDev...	Governor Edward W. Kelley, Jr.	Remarks by Governor Edward W. Kelley, Jr. Deve...	1	2692	52	54	-0.074294	[pleasure, progress, more, most, innovation, g...	[disap si succeed ne
1996-09-08	Monetary Policy Objectives and Strategy\n\n[SE...	Governor Laurence H. Meyer	Monetary Policy Objectives and StrategyI want ...	2	2900	102	111	-0.310345	[achieving, stability, delivering, good, good,...	[challe de stops, d overst
1996-09-19	Remarks by Chairman Alan Greenspan\nRegulation...	Chairman Alan Greenspan	Remarks by Chairman Alan Greenspan	3	1986	46	56	-0.503525	[versatile, high, popular, more,	[chall prem

			Regulation ...						innovation, m...	challen		
			Remarks by Governor Lawrence B. Lindsey\r\nAt ...	Governor Lawrence B. Lindsey	Remarks by Governor Lawrence B. Lindsey At th...	4	2177	70	41	1.332108	[pleasure, more, better, more, larger, profita...	[que concern, ill, t forc
1996-10-02											Istrong	
			Remarks by									
			Remarks by Chairman	Chairman	Remarks by Chairman						[bu	

```
def pickle_dump(df, filename='no-filename.pickle'):
    if filename:
        with open(filename, "wb") as output_file:
            pickle.dump(df, output_file)
```

▼ Plot the results and compare to the economical uncertainty / systemic risk periods

```
#Dot-com bubble
#https://en.wikipedia.org/wiki/Dot-com_bubble
DotCom = np.logical_and(Data.index > '2000-03',
                        Data.index < '2002-10'
                        )

#Financial crisis of 2007-2008
#https://en.wikipedia.org/wiki/Financial_crisis_of_2007%E2%80%932008
OhEight = np.logical_and(Data.index > '2007-04',
                        Data.index < '2009-03'
                        )

#European debt crisis
#https://en.wikipedia.org/wiki/European_debt_crisis
EuroDebt = np.logical_and(Data.index > '2010-09',
                        Data.index < '2012-09'
                        )

#2015-16 Chinese stock market turbulence
#https://en.wikipedia.org/wiki/2015%E2%80%9316_Chinese_stock_market_turbulence
Asian = np.logical_and(Data.index > '2015-01',
                        Data.index < '2016-06'
                        )
```

```
Recessions = np.logical_or.reduce( ( DotCom,
                                     OhEight,
                                     EuroDebt,
                                     Asian
                                   )
                                )
```

```
Window = 16
CompToMA = NetSentimentNorm.rolling(Window).mean()
```

```
fig, ax = plt.subplots(figsize=(30,14))
```

```
ax.plot(Data.index,
        CompToMA,
        c = 'r',
        linewidth= 2)
```

```
ax.plot(Data.index, NetSentimentNorm,
        c = 'green',
        linewidth= 1,
        alpha = 0.5)
```

```
# format the ticks
ax.xaxis.set_major_locator(years)
ax.xaxis.set_major_formatter(years_fmt)
ax.xaxis.set_minor_locator(months)
```

```
# round to nearest years.
datemin = np.datetime64(Data.index[0], 'Y')
datemax = np.datetime64(Data.index[-1], 'Y') + np.timedelta64(1, 'Y')
ax.set_xlim(datemin, datemax)
```

```
# format the coords message box
ax.format_xdata = mdates.DateFormatter('%Y-%m-%d')
ax.grid(True)
```

```
plt.title( str('Moving average of last ' + str(Window) + ' statements (~2 Year Window) seems to match with periods of economic uncertainty
```

```
ax.legend([str(str(Window) + ' statement MA'), 'Net sentiment of individual statements'], prop={'size': 16}, loc = 2)
```



```

    )

Bernanke = np.logical_and(Data.index > '2006-02-01',
                           Data.index < '2014-01-31'
                           )

Yellen = np.logical_and(Data.index > '2014-02-03',
                         Data.index < '2018-02-03'
                         )

Powell = np.logical_and(Data.index > '2018-02-05',
                         Data.index < '2022-02-05'
                         )

Speaker = np.logical_or.reduce((Greenspan, Yellen))


Window = 8
CompToMA = NetSentimentNorm.rolling(Window).mean()

fig, ax = plt.subplots(figsize=(15,7))

ax.scatter(Data.index,
           Data['Rate']*3,
           c = 'g')

ax.plot(Data.index,
        CompToMA,
        c = 'r',
        linewidth= 1.0)

ax.plot(Data.index, NetSentimentNorm,
        c = 'green',
        linewidth= 1,
        alpha = 0.5)

ax.legend([str(str(Window) + ' statements moving average') , 'Net sentiment of individual statements', 'Fed Funds Rate'],
          prop={'size': 14},
          loc = 1
          )

import matplotlib.dates as mdates
years = mdates.YearLocator()    # every year

```

```

months = mdates.MonthLocator() # every month
years_fmt = mdates.DateFormatter('%Y')

# format the ticks
ax.xaxis.set_major_locator(years)
ax.xaxis.set_major_formatter(years_fmt)
ax.xaxis.set_minor_locator(months)

# round to nearest years.
datemin = np.datetime64(Data.index[18], 'Y')
datemax = np.datetime64(Data.index[-1], 'Y') + np.timedelta64(1, 'Y')
ax.set_xlim(datemin, datemax)
ax.set_ylim(-10,30)

# format the coords message box
ax.format_xdata = mdates.DateFormatter('%Y-%m-%d')
ax.grid(True)
ax.tick_params(axis='both', which='major', labelsize=12)

import matplotlib.transforms as mtransforms
trans = mtransforms.blended_transform_factory(ax.transData, ax.transAxes)
theta = 0.9
ax.fill_between(Data.index, 0, 10, where = Speaker,
                facecolor='lightblue', alpha=0.5, transform=trans)

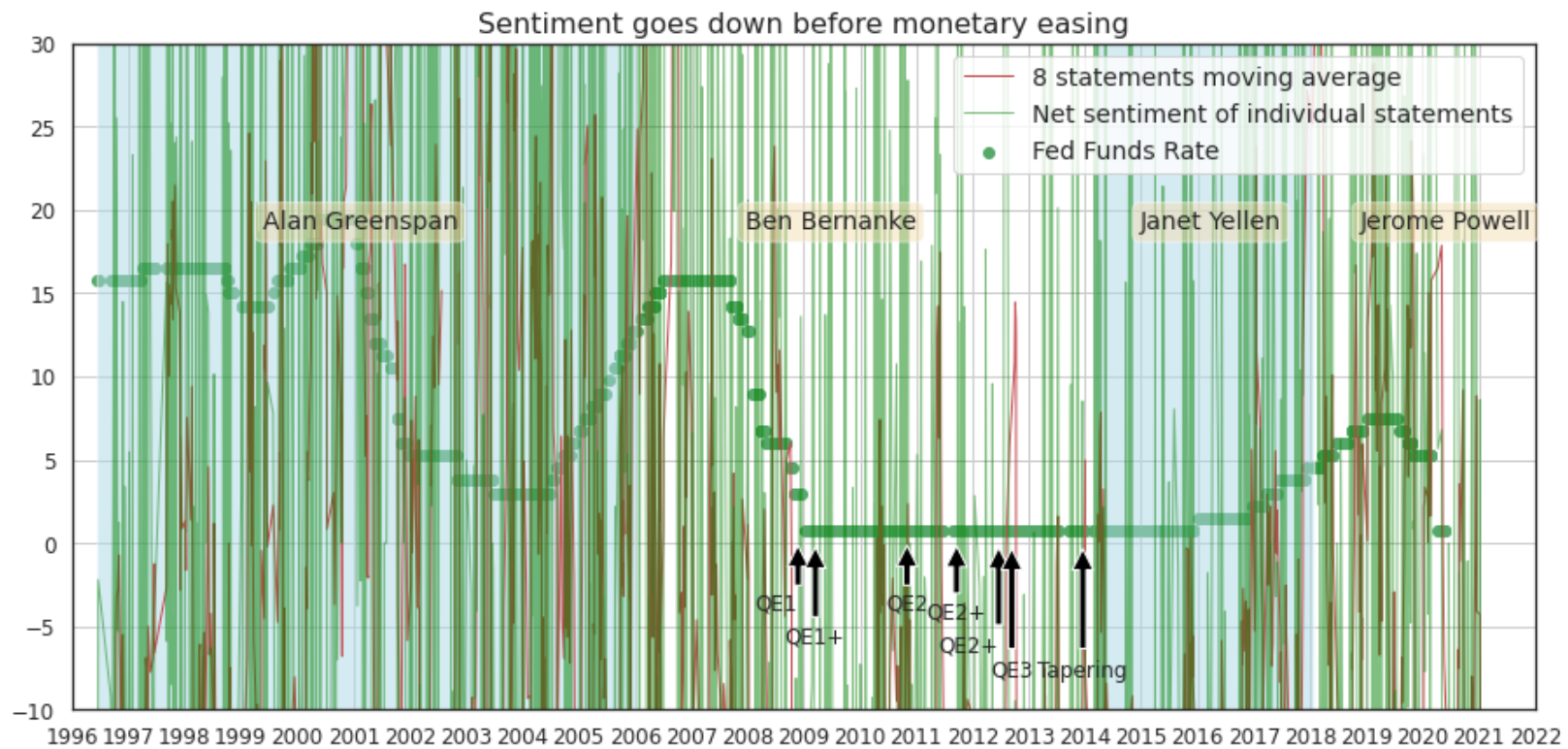
props = dict(boxstyle='round', facecolor='wheat', alpha=0.5)
ax.text(0.13, 0.75, "Alan Greenspan", transform=ax.transAxes, fontsize=14,
        verticalalignment='top', bbox=props)
ax.text(0.46, 0.75, "Ben Bernanke", transform=ax.transAxes, fontsize=14,
        verticalalignment='top', bbox=props)
ax.text(0.73, 0.75, "Janet Yellen", transform=ax.transAxes, fontsize=14,
        verticalalignment='top', bbox=props)
ax.text(0.88, 0.75, "Jerome Powell", transform=ax.transAxes, fontsize=14,
        verticalalignment='top', bbox=props)

arrow_style = dict(facecolor='black', edgecolor='white', shrink=0.05)
ax.annotate('QE1', xy=(pd.to_datetime('2008-11-25'), 0), xytext=(pd.to_datetime('2008-11-25'), -4), size=12, ha='right', arrowprops=arrow_
ax.annotate('QE1+', xy=(pd.to_datetime('2009-03-18'), 0), xytext=(pd.to_datetime('2009-03-18'), -6), size=12, ha='center', arrowprops=arrow
ax.annotate('QE2', xy=(pd.to_datetime('2010-11-03'), 0), xytext=(pd.to_datetime('2010-11-03'), -4), size=12, ha='center', arrowprops=arrow
ax.annotate('QE2+', xy=(pd.to_datetime('2011-09-21'), 0), xytext=(pd.to_datetime('2011-09-21'), -4.5), size=12, ha='center', arrowprops=ari
ax.annotate('QE2+', xy=(pd.to_datetime('2012-06-20'), 0), xytext=(pd.to_datetime('2012-06-20'), -6.5), size=12, ha='right', arrowprops=arrow
ax.annotate('QE3', xy=(pd.to_datetime('2012-09-13'), 0), xytext=(pd.to_datetime('2012-09-13'), -8), size=12, ha='center', arrowprops=arrow
ax.annotate('Tapering', xy=(pd.to_datetime('2013-12-18'), 0), xytext=(pd.to_datetime('2013-12-18'), -8), size=12, ha='center', arrowprops=

```

```
plt.title('Sentiment goes down before monetary easing', fontsize=16)
```

```
plt.show()
```



```
fig, ax = plt.subplots(figsize=(15,7))
```

```
Count = Data['wordcount']
```

```
ax.plot(Data.index, Count,
        c = 'red',
        linewidth= 1.5)
```

```
plt.title('Count of words per statement over time')
```

```
import matplotlib.dates as mdates
```



```

years = mdates.YearLocator() # every year
months = mdates.MonthLocator() # every month
years_fmt = mdates.DateFormatter('%Y')

# format the ticks
# round to nearest years.
datemin = np.datetime64(Data.index[0], 'Y')
datemax = np.datetime64(Data.index[-1], 'Y') + np.timedelta64(1, 'Y')
ax.set_xlim(datemin, datemax)

# format the coords message box
ax.format_xdata = mdates.DateFormatter('%Y-%m-%d')
ax.grid(True)

import matplotlib.transforms as mtransforms
trans = mtransforms.blended_transform_factory(ax.transData, ax.transAxes)
theta = 0.9
ax.fill_between(Data.index, 0, 10, where = Speaker,
                facecolor='lightblue', alpha=0.5, transform=trans)

props = dict(boxstyle='round', facecolor='wheat', alpha=0.5)
ax.text(0.05, 0.95, "Alan Greenspan", transform=ax.transAxes, fontsize=14,
        verticalalignment='top', bbox=props)
ax.text(0.50, 0.95, "Ben Bernanke", transform=ax.transAxes, fontsize=14,
        verticalalignment='top', bbox=props)
ax.text(0.77, 0.95, "Janet Yellen", transform=ax.transAxes, fontsize=14,
        verticalalignment='top', bbox=props)
ax.text(0.90, 0.95, "Jerome Powell", transform=ax.transAxes, fontsize=14,
        verticalalignment='top', bbox=props)

plt.show()

```



Count of words per statement over time

