In [56]:
```
from textstat.textstat import textstat
%matplotlib inline
import numpy as np
import pandas as pd
import scipy
import sklearn
#import spacy
import matplotlib.pyplot as plt
import seaborn as sns
import re
from collections import Counter
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn import linear_model
from sklearn.metrics import accuracy_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import GaussianNB
from gensim.models import doc2vec
from collections import namedtuple
#nlp = spacy.load('en')
import en_core_web_sm
nlp = en_core_web_sm.load()
from nltk.corpus import stopwords
from sklearn.model_selection import train_test_split
import pyphen
```

For this challenge, you will need to choose a corpus of data from nltk or another source that includes categories you can predict and create an analysis pipeline that includes the following steps:

1. Data cleaning / processing / language parsing
2. Create features using two different NLP methods: For example, BoW vs tf-idf.
3. Use the features to fit supervised learning models for each feature set to predict the category outcomes.
4. Assess your models using cross-validation and determine whether one model performed better.
5. Pick one of the models and try to increase accuracy by at least 5 percentage points.

In [57]:
```
import nltk
# Launch the installer to download "gutenberg" and "stop words" corpora.
nltk.download()
```

```
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
```

Out[57]: True

In [58]:
```
# Import the data we just downloaded and installed.
from nltk.corpus import inaugural

# Grab and process the raw data.
inaugural.fileids()
```

```
Out[58]: ['1789-Washington.txt',
          '1793-Washington.txt',
          '1797-Adams.txt',
          '1801-Jefferson.txt',
          '1805-Jefferson.txt',
          '1809-Madison.txt',
          '1813-Madison.txt',
          '1817-Monroe.txt',
          '1821-Monroe.txt',
          '1825-Adams.txt',
          '1829-Jackson.txt',
          '1833-Jackson.txt',
          '1837-VanBuren.txt',
          '1841-Harrison.txt',
          '1845-Polk.txt',
          '1849-Taylor.txt',
          '1853-Pierce.txt',
          '1857-Buchanan.txt',
          '1861-Lincoln.txt',
          '1865-Lincoln.txt',
          '1869-Grant.txt',
          '1873-Grant.txt',
          '1877-Hayes.txt',
          '1881-Garfield.txt',
          '1885-Cleveland.txt',
          '1889-Harrison.txt',
          '1893-Cleveland.txt',
          '1897-McKinley.txt',
          '1901-McKinley.txt',
          '1905-Roosevelt.txt',
          '1909-Taft.txt',
          '1913-Wilson.txt',
          '1917-Wilson.txt',
          '1921-Harding.txt',
          '1925-Coolidge.txt',
          '1929-Hoover.txt',
          '1933-Roosevelt.txt',
          '1937-Roosevelt.txt',
          '1941-Roosevelt.txt',
          '1945-Roosevelt.txt',
          '1949-Truman.txt',
          '1953-Eisenhower.txt',
          '1957-Eisenhower.txt',
          '1961-Kennedy.txt',
          '1965-Johnson.txt',
          '1969-Nixon.txt',
          '1973-Nixon.txt',
          '1977-Carter.txt',
          '1981-Reagan.txt',
          '1985-Reagan.txt',
          '1989-Bush.txt',
          '1993-Clinton.txt',
          '1997-Clinton.txt',
          '2001-Bush.txt',
          '2005-Bush.txt',
          '2009-Obama.txt']
```

1. Data cleaning / processing / language parsing

Raw format of the data

Resources: https://towardsdatascience.com/intro-to-nlp-using-inaugural-speeches-of-presidents-8c7ca32cbdfe (https://towardsdatascience.com/intro-to-nlp-using-inaugural-speeches-of-presidents-8c7ca32cbdfe)

In [59]:
```
names = inaugural.fileids()
print(len(names))
```

```
56
```

In [60]:
```
words=list(inaugural.words(fileids = names[55]))
words
```

```
Out[60]: ['My',
          'fellow',
          'citizens',
          ':',
          'I',
          'stand',
          'here',
          'today',
          'humbled',
          'by',
          'the',
          'task',
          'before',
          'us',
          ',',
          'grateful',
          'for',
          'the',
          'trust',
          'you',
          'have',
          'bestowed',
          ',',
          'mindful',
          'of',
          'the',
          'sacrifices',
          'borne',
          'by',
          'our',
          'ancestors',
          '.',
          'I',
          'thank',
          'President',
          'Bush',
          'for',
          'his',
          'service',
          'to',
          'our',
          'nation',
          ',',
          'as',
          'well',
          'as',
          'the',
          'generosity',
          'and',
          'cooperation',
          'he',
          'has',
          'shown',
          'throughout',
          'this',
          'transition',
          '.',
```

```
'Forty',
'-',
'four',
'Americans',
'have',
'now',
'taken',
'the',
'presidential',
'oath',
'.',
'The',
'words',
'have',
'been',
'spoken',
'during',
'rising',
'tides',
'of',
'prosperity',
'and',
'the',
'still',
'waters',
'of',
'peace',
'.',
'Yet',
',',
'every',
'so',
'often',
'the',
'oath',
'is',
'taken',
'amidst',
'gathering',
'clouds',
'and',
'raging',
'storms',
'.',
'At',
'these',
'moments',
',',
'America',
'has',
'carried',
'on',
'not',
'simply',
'because',
'of',
'the',
```

```
'skill',
'or',
'vision',
'of',
'those',
'in',
'high',
'office',
',',
'but',
'because',
'We',
'the',
'People',
'have',
'remained',
'faithful',
'to',
'the',
'ideals',
'of',
'our',
'forbearers',
',',
'and',
'true',
'to',
'our',
'founding',
'documents',
'.',
'So',
'it',
'has',
'been',
'.',
'So',
'it',
'must',
'be',
'with',
'this',
'generation',
'of',
'Americans',
'.',
'That',
'we',
'are',
'in',
'the',
'midst',
'of',
'crisis',
'is',
'now',
'well',
```

```
 'understood',
 '.',
 'Our',
 'nation',
 'is',
 'at',
 'war',
 ',',
 'against',
 'a',
 'far',
 '-',
 'reaching',
 'network',
 'of',
 'violence',
 'and',
 'hatred',
 '.',
 'Our',
 'economy',
 'is',
 'badly',
 'weakened',
 ',',
 'a',
 'consequence',
 'of',
 'greed',
 'and',
 'irresponsibility',
 'on',
 'the',
 'part',
 'of',
 'some',
 ',',
 'but',
 'also',
 'our',
 'collective',
 'failure',
 'to',
 'make',
 'hard',
 'choices',
 'and',
 'prepare',
 'the',
 'nation',
 'for',
 'a',
 'new',
 'age',
 '.',
 'Homes',
 'have',
```

```
'been',
'lost',
';',
'jobs',
'shed',
';',
'businesses',
'shuttered',
'.',
'Our',
'health',
'care',
'is',
'too',
'costly',
';',
'our',
'schools',
'fail',
'too',
'many',
';',
'and',
'each',
'day',
'brings',
'further',
'evidence',
'that',
'the',
'ways',
'we',
'use',
'energy',
'strengthen',
'our',
'adversaries',
'and',
'threaten',
'our',
'planet',
'.',
'These',
'are',
'the',
'indicators',
'of',
'crisis',
',',
'subject',
'to',
'data',
'and',
'statistics',
'.',
'Less',
'measurable',
```

```
                'but',
                'no',
                'less',
                'profound',
                'is',
                'a',
                'sapping',
                'of',
                'confidence',
                'across',
                'our',
                'land',
                '--',
                'a',
                'nagging',
                'fear',
                'that',
                'America',
                "'",
                's',
                'decline',
                'is',
                'inevitable',
                ',',
                'that',
                'the',
                'next',
                'generation',
                'must',
                'lower',
                'its',
                'sights',
                '.',
                'Today',
                'I',
                'say',
                'to',
                'you',
                'that',
                'the',
                'challenges',
                'we',
                'face',
                'are',
                'real',
                '.',
                'They',
                'are',
                'serious',
                'and',
                'they',
                'are',
                'many',
                '.',
                'They',
                'will',
                'not',
```

```
'be',
'met',
'easily',
'or',
'in',
'a',
'short',
'span',
'of',
'time',
'.',
'But',
'know',
'this',
',',
'America',
'--',
'they',
'will',
'be',
'met',
'.',
'On',
'this',
'day',
',',
'we',
'gather',
'because',
'we',
'have',
'chosen',
'hope',
'over',
'fear',
',',
'unity',
'of',
'purpose',
'over',
'conflict',
'and',
'discord',
'.',
'On',
'this',
'day',
',',
'we',
'come',
'to',
'proclaim',
'an',
'end',
'to',
'the',
'petty',
```

```
'grievances',
'and',
'false',
'promises',
',',
'the',
'recriminations',
'and',
'worn',
'-',
'out',
'dogmas',
'that',
'for',
'far',
'too',
'long',
'have',
'strangled',
'our',
'politics',
'.',
'We',
'remain',
'a',
'young',
'nation',
',',
'but',
'in',
'the',
'words',
'of',
'Scripture',
',',
'the',
'time',
'has',
'come',
'to',
'set',
'aside',
'childish',
'things',
'.',
'The',
'time',
'has',
'come',
'to',
'reaffirm',
'our',
'enduring',
'spirit',
';',
'to',
'choose',
```

```
'our',
'better',
'history',
';',
'to',
'carry',
'forward',
'that',
'precious',
'gift',
',',
'that',
'noble',
'idea',
',',
'passed',
'on',
'from',
'generation',
'to',
'generation',
':',
'the',
'God',
'-',
'given',
'promise',
'that',
'all',
'are',
'equal',
',',
'all',
'are',
'free',
',',
'and',
'all',
'deserve',
'a',
'chance',
'to',
'pursue',
'their',
'full',
'measure',
'of',
'happiness',
'.',
'In',
'reaffirming',
'the',
'greatness',
'of',
'our',
'nation',
',',
```

```
'we',
'understand',
'that',
'greatness',
'is',
'never',
'a',
'given',
'.',
'It',
'must',
'be',
'earned',
'.',
'Our',
'journey',
'has',
'never',
'been',
'one',
'of',
'shortcuts',
'or',
'settling',
'for',
'less',
'.',
'It',
'has',
'not',
'been',
'the',
'path',
'for',
'the',
'faint',
'-',
'hearted',
'--',
'for',
'those',
'who',
'prefer',
'leisure',
'over',
'work',
',',
'or',
'seek',
'only',
'the',
'pleasures',
'of',
'riches',
'and',
'fame',
'.',
```

```
                'Rather',
                ',',
                'it',
                'has',
                'been',
                'the',
                'risk',
                '-',
                'takers',
                ',',
                'the',
                'doers',
                ',',
                'the',
                'makers',
                'of',
                'things',
                "'",
                'some',
                'celebrated',
                'but',
                'more',
                'often',
                'men',
                'and',
                'women',
                'obscure',
                'in',
                'their',
                'labor',
                ',',
                'who',
                'have',
                'carried',
                'us',
                'up',
                'the',
                'long',
                ',',
                'rugged',
                'path',
                'towards',
                'prosperity',
                'and',
                'freedom',
                '.',
                'For',
                'us',
                ',',
                'they',
                'packed',
                'up',
                'their',
                'few',
                'worldly',
                'possessions',
                'and',
```

```
 'traveled',
 'across',
 'oceans',
 'in',
 'search',
 'of',
 'a',
 'new',
 'life',
 '.',
 'For',
 'us',
 ',',
 'they',
 'toiled',
 'in',
 'sweatshops',
 'and',
 'settled',
 'the',
 'West',
 ';',
 'endured',
 'the',
 'lash',
 'of',
 'the',
 'whip',
 'and',
 'plowed',
 'the',
 'hard',
 'earth',
 '.',
 'For',
 'us',
 ',',
 'they',
 'fought',
 'and',
 'died',
 ',',
 'in',
 'places',
 'like',
 'Concord',
 'and',
 'Gettysburg',
 ';',
 'Normandy',
 'and',
 'Khe',
 'Sahn',
 '.',
 'Time',
 'and',
 'again',
```

```
'these',
'men',
'and',
'women',
'struggled',
'and',
'sacrificed',
'and',
'worked',
'till',
'their',
'hands',
'were',
'raw',
'so',
'that',
'we',
'might',
'live',
'a',
'better',
'life',
'.',
'They',
'saw',
'America',
'as',
'bigger',
'than',
'the',
'sum',
'of',
'our',
'individual',
'ambitions',
';',
'greater',
'than',
'all',
'the',
'differences',
'of',
'birth',
'or',
'wealth',
'or',
'faction',
'.',
'This',
'is',
'the',
'journey',
'we',
'continue',
'today',
'.',
'We',
```

```
'remain',
'the',
'most',
'prosperous',
',',
'powerful',
'nation',
'on',
'Earth',
'.',
'Our',
'workers',
'are',
'no',
'less',
'productive',
'than',
'when',
'this',
'crisis',
'began',
'.',
'Our',
'minds',
'are',
'no',
'less',
'inventive',
',',
'our',
'goods',
'and',
'services',
'no',
'less',
'needed',
'than',
'they',
'were',
'last',
'week',
'or',
'last',
'month',
'or',
'last',
'year',
'.',
'Our',
'capacity',
'remains',
'undiminished',
'.',
'But',
'our',
'time',
'of',
```

```
'standing',
'pat',
',',
'of',
'protecting',
'narrow',
'interests',
'and',
'putting',
'off',
'unpleasant',
'decisions',
'--',
'that',
'time',
'has',
'surely',
'passed',
'.',
'Starting',
'today',
',',
'we',
'must',
'pick',
'ourselves',
'up',
',',
'dust',
'ourselves',
'off',
',',
'and',
'begin',
'again',
'the',
'work',
'of',
'remaking',
'America',
'.',
'For',
'everywhere',
'we',
'look',
',',
'there',
'is',
'work',
'to',
'be',
'done',
'.',
'The',
'state',
'of',
'our',
```

```
 'economy',
 'calls',
 'for',
 'action',
 ',',
 'bold',
 'and',
 'swift',
 ',',
 'and',
 'we',
 'will',
 'act',
 '--',
 'not',
 'only',
 'to',
 'create',
 'new',
 'jobs',
 ',',
 'but',
 'to',
 'lay',
 'a',
 'new',
 'foundation',
 'for',
 'growth',
 '.',
 'We',
 'will',
 'build',
 'the',
 'roads',
 'and',
 'bridges',
 ',',
 'the',
 'electric',
 'grids',
 'and',
 'digital',
 'lines',
 'that',
 'feed',
 'our',
 'commerce',
 'and',
 'bind',
 'us',
 'together',
 '.',
 'We',
 'will',
 'restore',
 'science',
```

```
                'to',
                'its',
                'rightful',
                'place',
                ',',
                'and',
                'wield',
                'technology',
                "'",
                's',
                'wonders',
                'to',
                'raise',
                'health',
                'care',
                "'",
                's',
                'quality',
                'and',
                'lower',
                'its',
                'cost',
                '.',
                'We',
                'will',
                'harness',
                'the',
                'sun',
                'and',
                'the',
                'winds',
                'and',
                'the',
                'soil',
                'to',
                'fuel',
                'our',
                'cars',
                'and',
                'run',
                'our',
                'factories',
                '.',
                'And',
                'we',
                'will',
                'transform',
                'our',
                'schools',
                'and',
                'colleges',
                'and',
                'universities',
                'to',
                'meet',
                'the',
                'demands',
```

```
 'of',
 'a',
 'new',
 'age',
 '.',
 'All',
 'this',
 'we',
 'can',
 'do',
 '.',
 'All',
 'this',
 'we',
 'will',
 'do',
 '.',
 'Now',
 ',',
 'there',
 'are',
 'some',
 'who',
 'question',
 'the',
 'scale',
 'of',
 'our',
 'ambitions',
 '--',
 'who',
 ...]
```

In [61]:
```
nltk.FreqDist(words)
```

Out[61]:
```
FreqDist({',': 130, 'the': 126, '.': 108, 'and': 105, 'of': 82, 'to': 66, 'ou
r': 58, 'we': 50, 'that': 48, 'a': 47, ...})
```

In [62]:
```
stop_words = stopwords.words('english')
add_to_stop_words=[',', '.', '-', ';', ':', '--', "'", '(',')']
stop_words.extend(add_to_stop_words)
stop_words=set(stop_words)
```

In [63]:
```
stop_words
```

```
Out[63]: {"'",
          '(',
          ')',
          ',',
          '-',
          '--',
          '.',
          ':',
          ';',
          'a',
          'about',
          'above',
          'after',
          'again',
          'against',
          'ain',
          'all',
          'am',
          'an',
          'and',
          'any',
          'are',
          'aren',
          "aren't",
          'as',
          'at',
          'be',
          'because',
          'been',
          'before',
          'being',
          'below',
          'between',
          'both',
          'but',
          'by',
          'can',
          'couldn',
          "couldn't",
          'd',
          'did',
          'didn',
          "didn't",
          'do',
          'does',
          'doesn',
          "doesn't",
          'doing',
          'don',
          "don't",
          'down',
          'during',
          'each',
          'few',
          'for',
          'from',
          'further',
```

```
                'had',
                'hadn',
                "hadn't",
                'has',
                'hasn',
                "hasn't",
                'have',
                'haven',
                "haven't",
                'having',
                'he',
                'her',
                'here',
                'hers',
                'herself',
                'him',
                'himself',
                'his',
                'how',
                'i',
                'if',
                'in',
                'into',
                'is',
                'isn',
                "isn't",
                'it',
                "it's",
                'its',
                'itself',
                'just',
                'll',
                'm',
                'ma',
                'me',
                'mightn',
                "mightn't",
                'more',
                'most',
                'mustn',
                "mustn't",
                'my',
                'myself',
                'needn',
                "needn't",
                'no',
                'nor',
                'not',
                'now',
                'o',
                'of',
                'off',
                'on',
                'once',
                'only',
                'or',
                'other',
```

```
                'our',
                'ours',
                'ourselves',
                'out',
                'over',
                'own',
                're',
                's',
                'same',
                'shan',
                "shan't",
                'she',
                "she's",
                'should',
                "should've",
                'shouldn',
                "shouldn't",
                'so',
                'some',
                'such',
                't',
                'than',
                'that',
                "that'll",
                'the',
                'their',
                'theirs',
                'them',
                'themselves',
                'then',
                'there',
                'these',
                'they',
                'this',
                'those',
                'through',
                'to',
                'too',
                'under',
                'until',
                'up',
                've',
                'very',
                'was',
                'wasn',
                "wasn't",
                'we',
                'were',
                'weren',
                "weren't",
                'what',
                'when',
                'where',
                'which',
                'while',
                'who',
                'whom',
```

```
'why',
'will',
'with',
'won',
"won't",
'wouldn',
"wouldn't",
'y',
'you',
"you'd",
"you'll",
"you're",
"you've",
'your',
'yours',
'yourself',
'yourselves'}
```

In [64]:
```
for i in stop_words:
    if i in words:
        while i in words:
            words.remove(i)
```

In [65]:
```
nltk.FreqDist(words)
```

Out[65]:
```
FreqDist({'us': 23, 'nation': 12, 'We': 12, 'new': 11, 'America': 10, 'The':
9, 'Our': 9, 'every': 8, 'must': 8, 'For': 8, ...})
```

In [66]:
```
def text_cleaner(text):
    # Visual inspection identifies a form of punctuation spaCy does not
    # recognize: the double dash '--'.  Better get rid of it now!
    text = re.sub(r'--',' ',text)
    text = re.sub("[\[].*?[\]]", "", text)
    text = ' '.join(text.split())
    return text
```

In [68]:
```
#Presidential parties from Washington to Obama
#link for all presidential parties: https://www.presidentsusa.net/partyofpresi
dents.html
#dem_list - democrat
#rep_list - repbulican
#fed_list - federalist
#demrep_list - democratic-republican
#whig_list - whig


dem_list = ['2009-Obama.txt','1997-Clinton.txt','1993-Clinton.txt','1977-Carte
r.txt','1965-Johnson.txt', '1885-Cleveland.txt', '1893-Cleveland.txt',
            '1853-Pierce.txt', '1857-Buchanan.txt', '1829-Jackson.txt', '1833-
Jackson.txt', '1837-VanBuren.txt', '1845-Polk.txt',
            '1961-Kennedy.txt','1949-Truman.txt','1945-Roosevelt.txt','1941-Ro
osevelt.txt','1937-Roosevelt.txt','1933-Roosevelt.txt']


rep_list = ['1953-Eisenhower.txt', '1957-Eisenhower.txt', '1969-Nixon.txt', '1
973-Nixon.txt', '1981-Reagan.txt', '1985-Reagan.txt', '1989-Bush.txt','2001-Bu
sh.txt', '2005-Bush.txt'
            '1861-Lincoln.txt', '1865-Lincoln.txt', '1869-Grant.txt', '1873-Gr
ant.txt', '1877-Hayes.txt', '1881-Garfield.txt', '1889-Harrison.txt',
            '1897-McKinley.txt', '1901-McKinley.txt', '1905-Roosevelt.txt', '1
909-Taft.txt', '1921-Harding.txt', '1925-Coolidge.txt', '1929-Hoover.txt']

fed_list = ['1789-Washington.txt','1793-Washington.txt','1797-Adams.txt']


demrep_list = ['1801-Jefferson.txt', '1805-Jefferson.txt', '1809-Madison.txt',
'1813-Madison.txt',
             '1817-Monroe.txt','1821-Monroe.txt','1825-Adams.txt']


whig_list = [ '1841-Harrison.txt', '1849-Taylor.txt']

#John Tyler - no address following Harrison's death
#Millard Fillmore - no address following Taylor's death
#Andrew Johnson - no address following Lincoln's death
```

Using textstat

https://pypi.org/project/textstat/ (https://pypi.org/project/textstat/)

In [69]:
```
import textstat
```

In [70]:
```
# make lists with the text and features
speeches = []
raw_text = []
clean_text = []
reading_ease = []
smog_index = []
flesch_kincaid_grade = []
coleman_liau_index = []
readability = []
chall_readability = []
diffwords = []
linsear_write_formula = []
gunning_fog = []
text_standard = []
party1 = []
twords = []
for p in inaugural.fileids():
    speeches.append(p)
    x = inaugural.raw(p)
    raw_text.append(x)
    clean = text_cleaner(x)
    clean_text.append(clean)

    ease = textstat.flesch_reading_ease(x)
    reading_ease.append(ease)
    smog = textstat.smog_index(x)
    smog_index.append(smog)
    fk_grade = textstat.flesch_kincaid_grade(x)
    flesch_kincaid_grade.append(fk_grade)
    liau = textstat.coleman_liau_index(x)
    coleman_liau_index.append(liau)
    read = textstat.automated_readability_index(x)
    readability.append(read)
    read2 = textstat.dale_chall_readability_score(x)
    chall_readability.append(read2)
    words = textstat.difficult_words(x)
    diffwords.append(words)
    write = textstat.linsear_write_formula(x)
    linsear_write_formula.append(write)
    fog = textstat.gunning_fog(x)
    gunning_fog.append(fog)
    standard = textstat.text_standard(x)
    text_standard.append(standard)

    token_words = nltk.word_tokenize(clean)
    twords.append(token_words)

    if p in dem_list:
        party1.append(1)  #democrat
    elif p in rep_list:
        party1.append(2)  #republican
    elif p in fed_list:
        party1.append(3)  #federalist
    elif p in demrep_list:
        party1.append(4)  #democrat-republican
    elif p in whig_list:
```

```
        party1.append(5)  #whig
    else:
        party1.append(-1)    #unknown
```

In [71]:
```
# make dataframe of the lists of features
sp = pd.DataFrame()
sp['speeches'] = speeches
sp['raw_text'] = raw_text
sp['clean_text'] = clean_text
sp['reading_ease'] = reading_ease
sp['smog_index'] = smog_index
sp['flesch_kincaid_grade'] = flesch_kincaid_grade
sp['coleman_liau_index'] = coleman_liau_index
sp['readability'] = readability
sp['chall_readability'] = chall_readability
sp['diffwords'] = diffwords
sp['linsear_write_formula'] = linsear_write_formula
sp['gunning_fog'] = gunning_fog
sp['text_standard'] = text_standard
sp['party'] = party1
sp['tokens'] = twords
```

In [72]:
```
pd.set_option('display.max_columns', None)
```

In [73]:
```
sp.head(5)
```

Out[73]:

	speeches	raw_text	clean_text	reading_ease	smog_index	flesch_kinca
0	1789-Washington.txt	Fellow-Citizens of the Senate and of the House...	Fellow-Citizens of the Senate and of the House...	-9.13	24.5	34.3
1	1793-Washington.txt	Fellow citizens, I am again called upon by the...	Fellow citizens, I am again called upon by the...	25.80	19.9	20.8
2	1797-Adams.txt	When it was first perceived, in early times, t...	When it was first perceived, in early times, t...	-26.58	26.2	41.0
3	1801-Jefferson.txt	Friends and Fellow Citizens:\n\nCalled upon to...	Friends and Fellow Citizens: Called upon to un...	31.32	17.5	20.8
4	1805-Jefferson.txt	Proceeding, fellow citizens, to that qualifica...	Proceeding, fellow citizens, to that qualifica...	8.99	21.0	29.4

WordNetLemmatizer

https://pythonprogramming.net/lemmatizing-nltk-tutorial/ (https://pythonprogramming.net/lemmatizing-nltk-tutorial/)

In [74]:
```
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
```

In [75]:
```
# this picks the most common words in each speech:
allwords = []
def bag_of_words(text):

    # Filter out punctuation and stop words.
    for word in text:
        word = word.lower()
        word = wordnet_lemmatizer.lemmatize(word)
        if word.isalnum() == True:
            if word not in stopwords.words('english'):
                allwords.append(word)
            else:
                continue
        else:
            continue

    # Return the most common words.
    return [item[0] for item in Counter(allwords).most_common(4000)]
```

In [76]:
```
# this goes through the whole speech dataframe to find the BOW
all_common_words = []
i = 0
for i in range(0,sp.shape[0]):
    z = bag_of_words(sp.tokens[i])
    all_common_words.append(z)
    z = []
    i =+ 1

# Can flatten list of lists    [[word, word],[word,word]]  ~  sum(list_of_list
s, [])
```

In [77]:
```
# this finds the common words among all speeches top words:

cw=[]

for i in range(0,sp.shape[0]):
    for word in all_common_words[i]:
        if word not in cw:
            cw.append(word)
        else:
            continue
```

In [78]:
```
print(len(cw))
```

```
5263
```

In [79]:
```
wordcount = pd.DataFrame(columns=cw)
wordcount['text_sentence'] = sp.clean_text
#wordcount['text_source'] = sp.party
wordcount.loc[:, cw] = 0

list_of_words = []
for i in range(0,sp.shape[0]):

    for word in sp.tokens[i]:
        word = word.lower()
        word = wordnet_lemmatizer.lemmatize(word)
        if word.isalnum() == True:
            if word not in stopwords.words('english'):
                if word in cw:
                    list_of_words.append(word)


    # Populate the row with word counts.
    for w in list_of_words:
        wordcount.loc[i, w] += 1

    # reset list again
    list_of_words = []
```

In [80]:
```
wordcount.head(10)
```

Out[80]:

	every	government	public	may	present	country	duty	ha	wa	one	hand	citizen	ou
0	9	9	6	6	5	5	5	5	4	4	4	4	4
1	0	1	0	1	1	1	0	1	0	0	0	1	0
2	5	18	6	13	2	10	3	7	8	1	0	3	0
3	2	13	4	8	0	4	2	4	1	6	0	5	1
4	4	3	14	10	3	5	8	5	3	1	1	10	0
5	2	0	6	1	1	5	3	4	0	2	0	0	0
6	5	3	2	2	0	6	1	6	7	2	1	3	0
7	14	21	8	10	4	11	9	19	6	2	3	9	6
8	13	14	5	15	3	8	6	35	16	6	0	14	3
9	5	21	9	3	1	10	9	27	9	4	3	3	0

In [81]:
```
sp.rename(columns={'party': 'political_party'}, inplace=True)

result = pd.concat([wordcount, sp], axis=1)
```

In [82]:
```
result.head(10)
```

Out[82]:

	every	government	public	may	present	country	duty	ha	wa	one	hand	citizen	ou
0	9	9	6	6	5	5	5	5	4	4	4	4	4
1	0	1	0	1	1	1	0	1	0	0	0	1	0
2	5	18	6	13	2	10	3	7	8	1	0	3	0
3	2	13	4	8	0	4	2	4	1	6	0	5	1
4	4	3	14	10	3	5	8	5	3	1	1	10	0
5	2	0	6	1	1	5	3	4	0	2	0	0	0
6	5	3	2	2	0	6	1	6	7	2	1	3	0
7	14	21	8	10	4	11	9	19	6	2	3	9	6

	every	government	public	may	present	country	duty	ha	wa	one	hand	citizen	ou
8	13	14	5	15	3	8	6	35	16	6	0	14	3
9	5	21	9	3	1	10	9	27	9	4	3	3	0

In [83]:
```
result.corr()
```

Out[83]: