Illuminating Mainstream Media Political Bias through Text Mining

Aaron Carr, Azucena Faus, and Dave Friesen - ADS-509-01-SU23

In [1]:
```
__author__ = 'Aaron Carr, Azucena Faus, Dave Friesen'
__email__ = 'acarr@sandiego.edu, afaus@sandiego.edu, dfriesen@sandiego.edu'
__version__ = '1.0'
__date__ = 'June 2023'
```

Setup

In [2]:
```
# Import basic and data access libraries
import numpy as np
import pandas as pd
from profiler import profile, profile_cat

# Import pre-processing, model and performance evaluation libraries
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from nltk.tokenize import sent_tokenize, word_tokenize
from sklearn.decomposition import LatentDirichletAllocation
from model_process import ModelProcess

# Import lexicons
#import nltk
#nltk.download('opinion_lexicon')
from nltk.corpus import opinion_lexicon

# Import visualization libraries
from matplotlib import pyplot as plt
%matplotlib inline
import seaborn as sns
from wordcloud import WordCloud

# Import utility libraries
from collections import Counter, defaultdict
from tqdm import tqdm; tqdm.pandas()
```

In [3]:
```
# Set basic np, pd, and plt output defaults (keeping this code 'clean')
%run -i 'defaults.py'
```

Data Ingestion

In [4]:
```
# Instantiate and confirm master dataframe
master_df = pd.read_csv('../data/master.csv')
master_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4509 entries, 0 to 4508
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   source_name   4509 non-null   object
```

```
 1   author         4472 non-null   object
 2   title          4509 non-null   object
 3   url            4509 non-null   object
 4   publish_date   4509 non-null   object
 5   content        1158 non-null   object
 6   article_text   4508 non-null   object
dtypes: object(7)
memory usage: 246.7+ KB
```

In [5]:

```
# "Blanket" label data based on purported source leaning
target_cls_col = 'lean'

def assign_lean(source_name):
    if source_name == 'Breitbart News' or source_name == 'Fox News':
        return 'right'
    elif source_name == 'CNN' or source_name == 'The Washington Post':
        return 'left'
    else:
        return np.nan
master_df[target_cls_col] = master_df['source_name'].apply(lambda x: assign_lean(x))
```

Tokenization and Cleaning

In [6]:

```
# Stopword removal function, with related initialization
from nltk.corpus import stopwords
sw = stopwords.words('english')
def remove_stop(tokens):
    filtered_tokens = [word for word in tokens if word not in sw]
    return(filtered_tokens)

# Token join back to string
def join_tokens(tokens):
    return ' '.join(tokens)

# Tokenizing function
def tokenize(text):
    return(text.split())  # Tokenize on white space

# Emoji-to-text conversion function
import emoji
def convert_emojis(text):
#    return emoji.demojize(text)
    return emoji.demojize(text).replace('_', ' ')

# Contains-emojis function, with related initialization
all_language_emojis = set()
for country in emoji.EMOJI_DATA :
    for em in emoji.EMOJI_DATA[country]:
        all_language_emojis.add(em)
def contains_emoji(s):
    s = str(s)
    emojis = [ch for ch in s if ch in all_language_emojis]
    return(len(emojis) > 0)

# Punctuation removal function, with related initialization
from string import punctuation
tw_punct = set(punctuation + '’') - {'#'}
def remove_punct(text, punct_set=tw_punct):
    return(''.join([ch for ch in text if ch not in punct_set]))

# Preparation (pipeline) function
def prepare(text, pipeline):
    tokens = str(text)
```

```
    for transform in pipeline:
        tokens = transform(tokens)
    return(tokens)
```

In [7]:

```
# Set pipeline
pipeline = [str.lower, remove_punct, convert_emojis, tokenize, remove_stop]

# Clean and tokenize master dataframe
master_df['article_tokens'] = master_df['article_text'].progress_apply(lambda x: prepare(x
master_df['article_text_tokenized'] = master_df['article_tokens'].progress_apply(lambda x:
print(master_df['article_tokens'])
print(master_df['article_text_tokenized'])
```

```
100%|██████████████████████████████████| 4509/4509 [00:09<00:00, 493.88it/s]
100%|██████████████████████████████████| 4509/4509 [00:00<00:00, 86571.74it/s]
0       [travelers, alabama, driving, interstate, 65, ...
1       [federal, prosecutor, may, nearing, decision, ...
2       [federal, appeals, court, tuesday, cleared, wa...
3       [speaking, orlando, november, 2015, republican...
4                                                   [nan]
                              ...
4504    [germanys, populist, alternative, germany, afd...
4505    [president, bidens, justice, department, seemi...
4506    [incumbent, turkish, president, recep, tayyip,...
4507    [throughout, month, may, farleft, cnn, attract...
4508    [disney, known, fighting, antigrooming, legisl...
Name: article_tokens, Length: 4509, dtype: object
0       travelers alabama driving interstate 65 partie...
1       federal prosecutor may nearing decision whethe...
2       federal appeals court tuesday cleared way drug...
3       speaking orlando november 2015 republican pres...
4                                                     nan
                              ...
4504    germanys populist alternative germany afd surg...
4505    president bidens justice department seemingly ...
4506    incumbent turkish president recep tayyip erdog...
4507    throughout month may farleft cnn attracted mea...
4508    disney known fighting antigrooming legislation...
Name: article_text_tokenized, Length: 4509, dtype: object
```

Descriptive Stats

In [8]:

```
# Descriptive stats function
def descriptive_stats(tokens, num_tokens=5, verbose=False):
    num_tokens = len(tokens)
    num_unique_tokens = len(set(tokens))  # set() creates unordered set of unique elements
    num_characters = sum(len(token) for token in tokens)  # Finds characters sans spaces
    lexical_diversity = num_unique_tokens / num_tokens

    if verbose:
        print(f'There are {num_tokens} tokens in the data.')
        print(f'There are {num_unique_tokens} unique tokens in the data.')
        print(f'There are {num_characters} characters in the data.')
        print(f'The lexical diversity is {lexical_diversity:.3f} in the data.')

    return([num_tokens, num_unique_tokens, lexical_diversity, num_characters])
```

In [9]:

```
# Descriptive stats across all sources
descriptive_stats([token for sublist in master_df['article_tokens'] for token in sublist])
```

Out[9]: [1977106, 84569, 0.0427741355294051, 12724251]

In [10]:
```
# Standard dataframe profile for confirmation
profile(master_df)
```

100%|██████████| 4509/4509 [00:00<00:00, 1305454.32it/s]

	Dtype	count	unique	na	na%	mean	std	min	max	skew(>=3)	<v0.01	V
source_name	object	4509.0	4.0									
author	object	4472.0	956.0	37.0	0.8							
title	object	4509.0	4509.0									
url	object	4509.0	4509.0									
publish_date	object	4509.0	4487.0									
content	object	1158.0	1158.0	3351.0	74.3							
article_text	object	4508.0	4508.0	1.0								
lean	object	4509.0	2.0									
article_tokens	object	1977106.0	84569.0									
article_text_tokenized	object	4509.0	4509.0									

In [11]:
```
# Descriptive stats aggregating function
def aggregate_and_describe(group):
    aggregate_tokens = [token for sublist in group['article_tokens'].tolist() for token in
    return descriptive_stats(aggregate_tokens)

# Aggregate descriptive stats by source; convert to dataframe; sort and output
grouped_stats = master_df.groupby('source_name').apply(aggregate_and_describe)
grouped_stats_df = pd.DataFrame(grouped_stats.tolist(), index=grouped_stats.index,
                                columns=['num_tokens', 'num_unique_tokens', 'lexical_diver
grouped_stats_df = grouped_stats_df.sort_index(ascending=False)
print(grouped_stats_df)
```

source_name	num_tokens	num_unique_tokens	lexical_diversity	num_characters
The Washington Post	366707	32341	0.09	2370171
Fox News	828739	47097	0.06	5326935
CNN	409422	34724	0.08	2628951
Breitbart News	372238	36815	0.10	2398194

Word Cloud

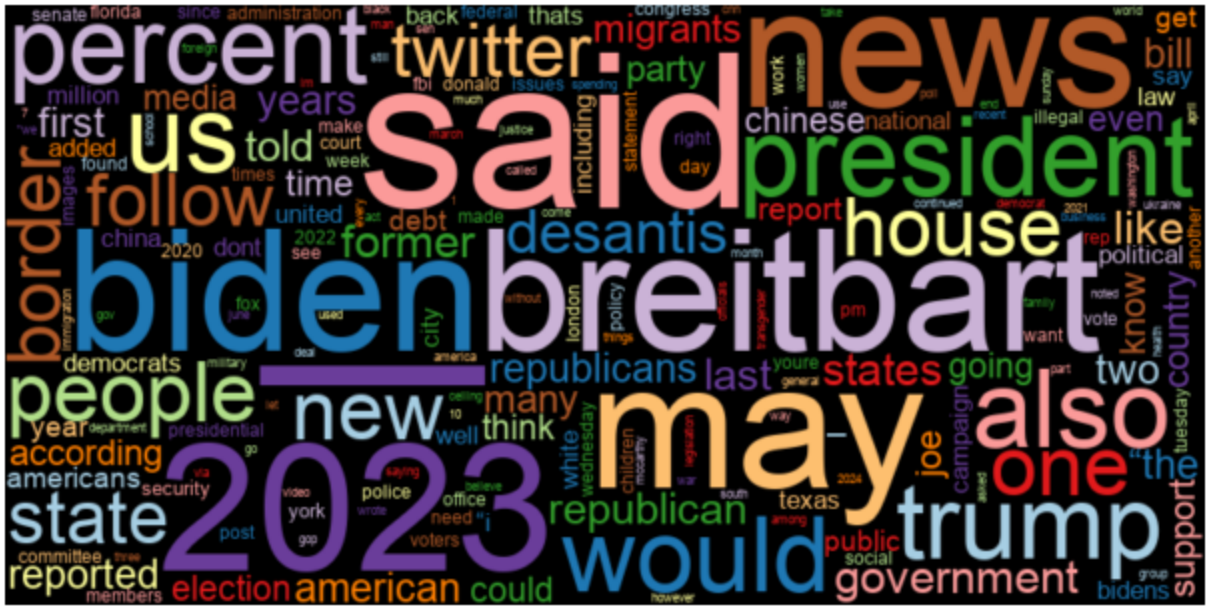
In [12]:
```
# Word cloud function
def wordcloud(word_freq, title=None, max_words=200, stopwords=None):
    wc = WordCloud(font_path='/Library/Fonts/Arial.ttf',
                   width=800, height=400,
                   background_color= "black", colormap="Paired",
                   max_font_size=150, max_words=max_words)

    # Convert data frame into dict
    if type(word_freq) == pd.Series:
        counter = Counter(word_freq.fillna(0).to_dict())
    else:
```

```
        counter = word_freq

    # filter stop words in frequency counter
    if stopwords is not None:
        counter = {token:freq for (token, freq) in counter.items()
                              if token not in stopwords}
    wc.generate_from_frequencies(counter)

    plt.title(title)

    plt.imshow(wc, interpolation='bilinear')
    plt.axis("off")

    plt.show()

# Word count function counter
def count_words(df, column='article_tokens', preprocess=None, min_freq=2):
    # Process tokens and update counter
    def update(doc):
        tokens = doc if preprocess is None else preprocess(doc)
        counter.update(tokens)

    # Create counter and run through all data
    counter = Counter()
    df[column].map(update)

    # Transform counter into data frame
    freq_df = pd.DataFrame.from_dict(counter, orient='index', columns=['freq'])
    freq_df = freq_df.query('freq >= @min_freq')
    freq_df.index.name = 'token'

    return freq_df.sort_values('freq', ascending=False)
```

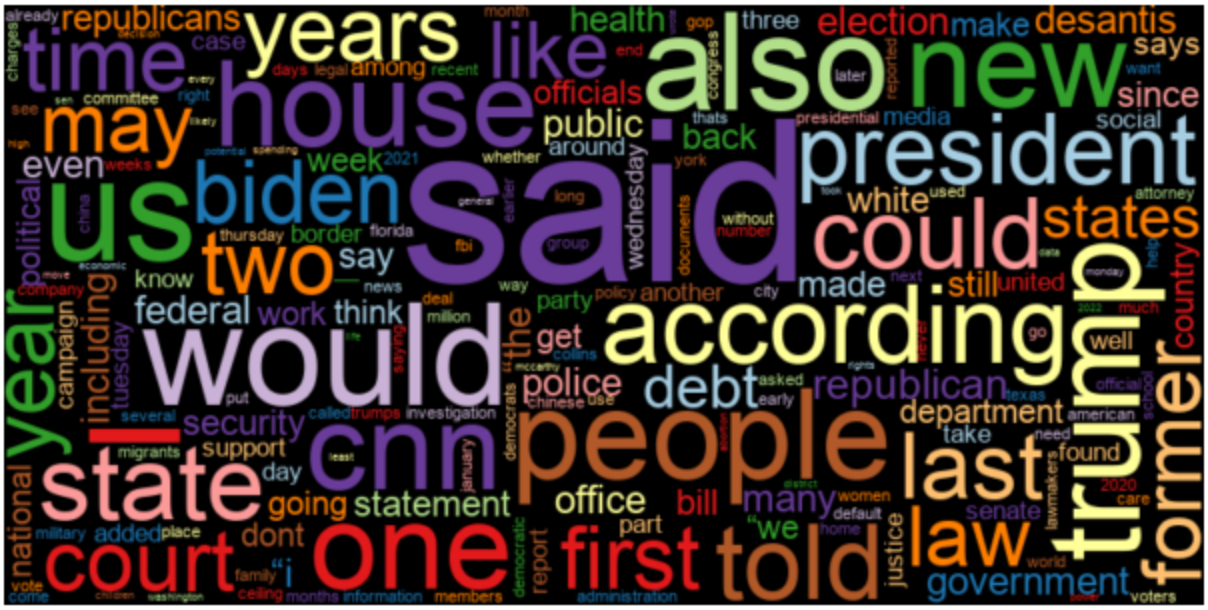
In [13]:

```
# Iterate and produce word cloud by source
for name, group in master_df.groupby('source_name'):
    print(f"Wordcloud for source: {name}")
    wordcloud(count_words(group)['freq'].to_dict())
```

Wordcloud for source: Breitbart News

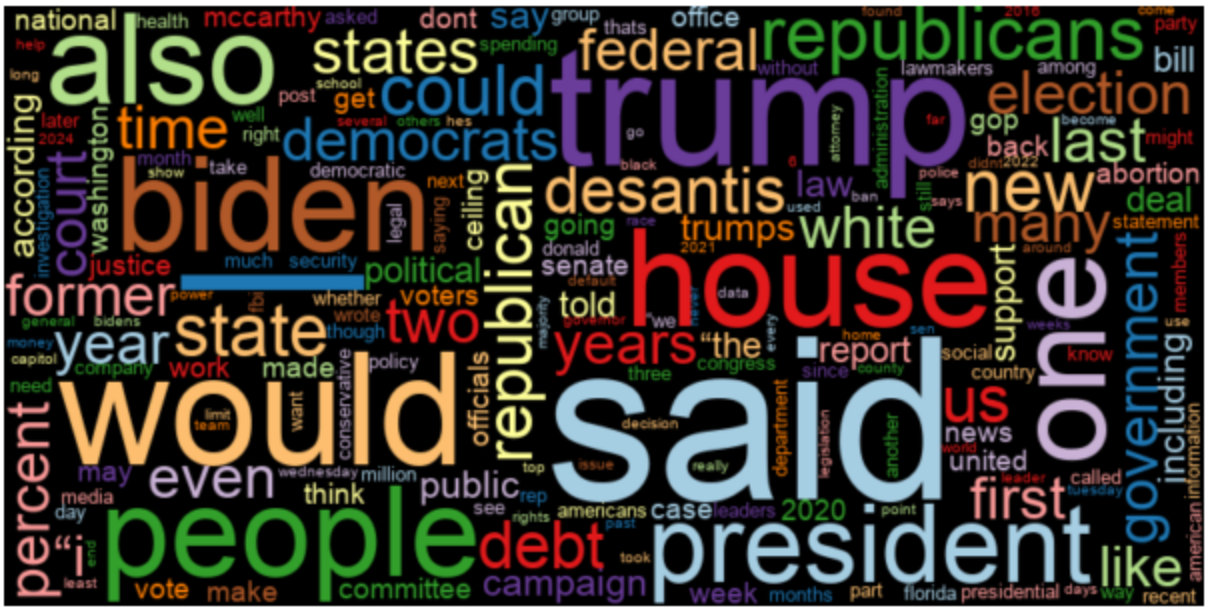


Wordcloud for source: CNN



Wordcloud for source: Fox News

Wordcloud for source: The Washington Post



Partition Data

In [14]:
```
# Set splits
train_ratio = 0.7; val_ratio = 0.20; test_ratio = 0.10

# Split and profile
train_df, test_df = train_test_split(master_df, test_size=1-train_ratio,
                                     random_state=42, stratify=master_df[target_cls_col])
val_df, test_df = train_test_split(test_df, test_size=test_ratio/(test_ratio+val_ratio),
                                   random_state=42, stratify=test_df[target_cls_col])
profile_cat(train_df, [target_cls_col])
```

```
lean -
right 71.17
 left 28.83
```

Topic Modeling

In [15]:
```
# Topic summarization function, from BTAP repo
def display_topics(model, features, no_top_words=5):
    for topic, words in enumerate(model.components_):
        total = words.sum()
        largest = words.argsort()[::-1]  # invert sort order

        print('\nTopic %02d' % topic, end=':')

        out = []
        for i in range(0, no_top_words):
            out.append(' %s (%2.2f)' % (features[largest[i]], abs(words[largest[i]]*100.0/
        print(';'.join(out), end='')
```

In [16]:
```
# Model topics by source
for source_name, group in train_df.groupby('source_name'):
    print(f'Topic modeling for source: {source_name}')

    # Transform article tokens into bag-of-words document-term sparse matrix
    count_vectorizer = CountVectorizer(min_df=0.05, max_df=0.75)
    count_vectors = count_vectorizer.fit_transform(group['article_text_tokenized'])
#    print('Vector shape:', count_vectors.shape)

    lda_model = LatentDirichletAllocation(n_components=5, random_state=42)
    W_lda_matrix = lda_model.fit_transform(count_vectors)
    H_lda_matrix = lda_model.components_

    display_topics(lda_model, count_vectorizer.get_feature_names_out())
    print('\n')
```

```
Topic modeling for source: Breitbart News

Topic 00: percent (5.31); desantis (2.53); trump (2.50); news (1.30); president (1.29)
Topic 01: biden (2.85); border (1.88); house (1.82); migrants (1.29); debt (1.23)
Topic 02: chinese (0.98); people (0.97); government (0.94); china (0.94); may (0.89)
Topic 03: 2023 (1.45); women (1.06); children (1.05); may (1.02); news (0.95)
Topic 04: trump (2.19); president (1.46); think (1.27); thats (1.14); people (1.03)

Topic modeling for source: CNN

Topic 00: people (1.18); health (0.99); new (0.98); one (0.77); like (0.73)
Topic 01: us (1.97); government (0.81); china (0.71); chinese (0.65); security (0.64)
Topic 02: police (1.67); according (1.33); cnn (1.17); told (1.02); people (0.79)
Topic 03: trump (2.55); desantis (1.70); former (1.29); president (1.16); court (1.08)
Topic 04: house (2.23); debt (1.78); would (1.68); biden (1.12); bill (1.12)
```

```
Topic modeling for source: Fox News

Topic 00: ai (1.14); people (1.09); also (0.85); us (0.80); like (0.77)
Topic 01: trump (2.19); president (1.75); desantis (1.59); former (1.31); campaign (1.06)
Topic 02: biden (2.87); house (2.38); president (1.46); debt (1.14); fbi (1.12)
Topic 03: border (2.02); state (1.98); school (1.41); law (1.38); migrants (1.21)
Topic 04: police (1.77); according (0.92); told (0.90); two (0.79); one (0.78)

Topic modeling for source: The Washington Post

Topic 00: state (1.91); abortion (1.39); republicans (0.94); bill (0.84); ban (0.81)
Topic 01: trump (1.31); court (0.92); election (0.76); case (0.74); justice (0.72)
Topic 02: trump (3.52); desantis (1.91); president (1.06); trumps (0.85); election (0.83)
Topic 03: people (1.09); states (0.68); new (0.63); us (0.56); health (0.54)
Topic 04: house (2.33); debt (2.05); biden (1.77); republicans (1.29); mccarthy (1.19)
```

Text Summarization and Sentiment Analysis

In [17]:

```
# NLTK opinion lexicon
positive_words = set(opinion_lexicon.positive())
negative_words = set(opinion_lexicon.negative())
```

In [18]:

```
# List of "assumed" political phrases
political_phrases = ['gun rights', 'voting rights', 'climate change', 'immigration reform'
                     'tax cuts', 'universal healthcare']
```

In [19]:

```
# Group train_df by 'source_name' for source-level comparison
grouped_df = train_df.groupby('source_name')

# Create dictionaries for scores
political_phrase_scores = {}
sentiment_scores = {}

# Iterate over sources and calc TF-IDF scores vs. political phrases
for source, group in tqdm(grouped_df):
    tfidf_vectorizer = TfidfVectorizer(ngram_range=(1, 3))
    tfidf_vectors = tfidf_vectorizer.fit_transform(group['article_text_tokenized'])

    # Calc TF-IDF sum (scores) where political phrases found
    scores = {}
    sentiment = defaultdict(lambda: defaultdict(int))

    # Iterate over political phrases
    for phrase in political_phrases:
        try:
            index = tfidf_vectorizer.get_feature_names_out().tolist().index(phrase)  # try
            scores[phrase] = tfidf_vectors[:, index].sum()  # and sum related score
        except ValueError:
            pass  # didn't find political phrase

        # Iterate over each article in the group to calc sentiment
        for text in group['article_text_tokenized']:
            # Tokenize text into sentences because we're calc'ing sentiment on phrase-rele
            sentences = sent_tokenize(text)

            # Check each sentence if it contains the political phrase
            for sentence in sentences:
                if phrase in sentence:
                    # [Tokenize the sentence into words
                    tokens = word_tokenize(sentence)
```

```
                    # Count positive and negative words
                    for word in tokens:
                        if word in positive_words:
                            sentiment[phrase]['positive'] += 1
                        elif word in negative_words:
                            sentiment[phrase]['negative'] += 1

    # Add the scores to the dictionary
    political_phrase_scores[source] = scores
    sentiment_scores[source] = dict(sentiment)
```

```
100%|██████████| 4/4 [00:23<00:00,  5.84s/it]
```

In [20]:

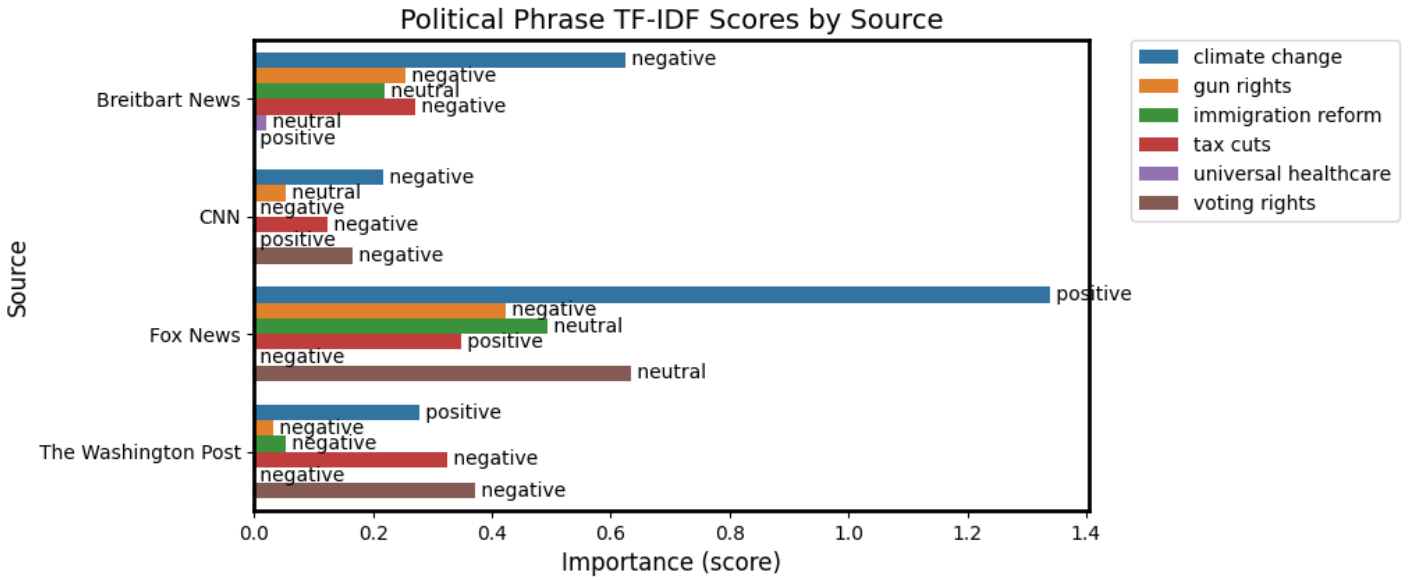
```
# Calc aggregate scores against which to compare "hits" above
all_scores = np.asarray(tfidf_vectors.sum(axis=0)).flatten()
mean_score = np.mean(all_scores)
median_score = np.median(all_scores)

results_df = pd.DataFrame()

# Iterate over sources and political phrase TF-IDF scores and show results
for source_name in political_phrase_scores:
    print(f'\nScores for {source_name}:')
    phrase_scores = political_phrase_scores[source_name]
    sentiment = sentiment_scores[source_name]

    # . . . by political phrase
    results = []
    for phrase in political_phrases:
        score = phrase_scores.get(phrase, 0)
        relative_to_mean = score / mean_score if mean_score != 0 else 0
        relative_to_median = score / median_score if median_score != 0 else 0

        # Categorize based on relative_to_median (otherwise arbitrary)
        if relative_to_median > 10:
            category = 'high'
        elif 5 < relative_to_median <= 10:
            category = 'medium'
        else:
            category = 'low'

        sentiment_phrase = sentiment.get(phrase, {'positive': 0, 'negative': 0})
        results.append({
            'source_name': source_name,
            'phrase': phrase,
            'score': score,
            'relative_to_mean': relative_to_mean,
            'relative_to_median': relative_to_median,
            'category': category,
            'p_sentiment': sentiment_phrase['positive'],
            'n_sentiment': sentiment_phrase['negative'],
            'sentiment': sentiment_phrase['positive'] + (sentiment_phrase['negative'] * -1
        })

    for result in results:
        result['sentiment_label'] = 'positive' if result['sentiment'] > 0 else 'negative'

    # Sort results by score
    results.sort(key=lambda x: x['score'], reverse=True)

    # Print sorted results
    for result in results:
        print(f"{result['phrase']}:  {result['category']} importance ",
              f"{result['sentiment_label']} sentiment")
```

```
    results_df = pd.concat([results_df, pd.DataFrame(results)])
```

```
Scores for Breitbart News:
climate change:  high importance  negative sentiment
tax cuts:  high importance  positive sentiment
gun rights:  high importance  negative sentiment
immigration reform:  high importance  positive sentiment
universal healthcare:  low importance  negative sentiment
voting rights:  low importance  neutral sentiment

Scores for CNN:
climate change:  high importance  negative sentiment
voting rights:  medium importance  negative sentiment
tax cuts:  medium importance  negative sentiment
gun rights:  low importance  negative sentiment
immigration reform:  low importance  neutral sentiment
universal healthcare:  low importance  neutral sentiment

Scores for Fox News:
climate change:  high importance  negative sentiment
voting rights:  high importance  positive sentiment
immigration reform:  high importance  positive sentiment
gun rights:  high importance  negative sentiment
tax cuts:  high importance  negative sentiment
universal healthcare:  low importance  neutral sentiment

Scores for The Washington Post:
voting rights:  high importance  negative sentiment
tax cuts:  high importance  negative sentiment
climate change:  high importance  negative sentiment
immigration reform:  low importance  positive sentiment
gun rights:  low importance  negative sentiment
universal healthcare:  low importance  neutral sentiment
```

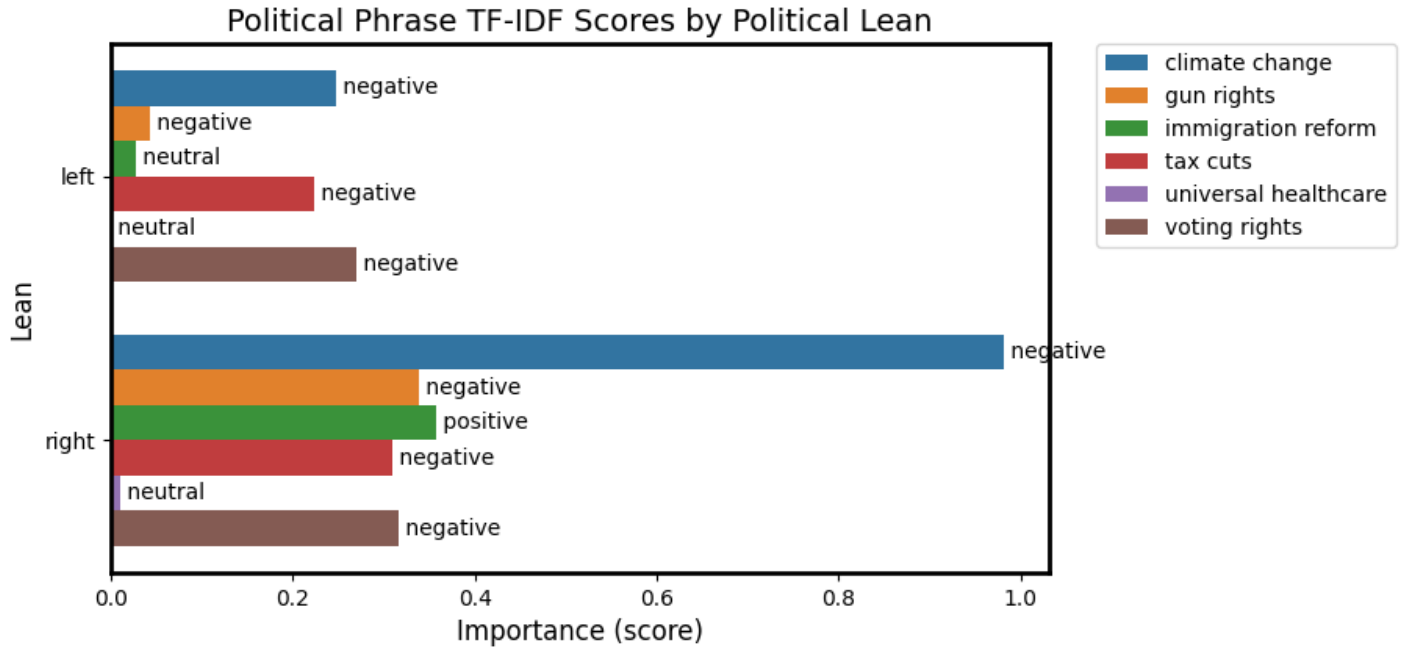
In [21]:

```
# Sort DataFrame by 'source' and 'phrase' to match order of bars in plot
sorted_df = results_df.sort_values(['source_name', 'phrase'])

# Create barplot
fig, ax = plt.subplots()
sns.barplot(data=sorted_df, y='source_name', x='score',
            ax=ax, hue='phrase', errorbar=None)

# Iterate over bars and dataframe rows to add sentiment
for p, (_, row) in zip(ax.patches, sorted_df.iterrows()):
    plt.text(p.get_width(), p.get_y() + p.get_height()/2,
             f' {row["sentiment_label"]}',
             ha='left', va='center')

ax.set_title('Political Phrase TF-IDF Scores by Source')
ax.set_xlabel('Importance (score)')
ax.set_ylabel('Source')

plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)

plt.show()
```



In [22]:

```
# Sort DataFrame by political 'lean' and 'phrase' to match order of bars in plot
lean_df = train_df[['source_name', 'lean']].drop_duplicates()
results_df = pd.merge(results_df, lean_df, on='source_name', how='left')
sorted_df = results_df.sort_values(['lean', 'phrase'])

# Create barplot
fig, ax = plt.subplots()
sns.barplot(data=sorted_df, y='lean', x='score',
            ax=ax, hue='phrase', errorbar=None)

# Iterate over bars and dataframe rows to add sentiment
for p, (_, row) in zip(ax.patches, sorted_df.iterrows()):
    plt.text(p.get_width(), p.get_y() + p.get_height()/2,
             f' {row["sentiment_label"]}',
             ha='left', va='center')

ax.set_title('Political Phrase TF-IDF Scores by Political Lean')
ax.set_xlabel('Importance (score)')
ax.set_ylabel('Lean')

plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)

plt.show()
```



In [ ]:

PreProcess and EDA

Azucena Faus

Setup

Import libraries:

In [ ]:

```
import pyLDAvis

pyLDAvis.enable_notebook()
from tqdm.auto import tqdm

import pyLDAvis.lda_model
import pyLDAvis.gensim_models

import numpy as np
import pandas as pd
import pymysql as mysql
import matplotlib.pyplot as plt
import os
import shutil
import re
import logging
import time
import zipfile
import requests
from bs4 import BeautifulSoup
import datetime
import re
import regex as rex
from collections import defaultdict, Counter
import random
import requests
from bs4 import BeautifulSoup
import datetime
import json
from wordcloud import WordCloud
from tabulate import tabulate
from sklearn.svm import SVC


import sqlite3
import nltk
from string import punctuation
from nltk.corpus import stopwords
import re
import emoji
from nltk.metrics import ConfusionMatrix
import itertools
import collections

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.decomposition import NMF
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, f1_score, recall_score, precision_recall_curve
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
## Depracated:
# from sklearn.metrics import plot_confusion_matrix
## New version:
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.model_selection import cross_val_score
from sklearn.pipeline import Pipeline
from sklearn.model_selection import GridSearchCV
from sklearn.svm import LinearSVC
from sklearn.decomposition import LatentDirichletAllocation

#import mysql.connector
```

```
# Set pandas global options
pd.options.display.max_rows = 17

import warnings
warnings.filterwarnings('ignore')
warnings.simplefilter('ignore')
```

Functions:

Data pre-processing:

In [ ]:

```
punctuation = set(punctuation) # speeds up comparison
tw_punct = punctuation - {"#"}


# Stopwords - added the 'nan' to this to remove nulls:
# next step could be to add pronouns like she/her, he/him, etc.

sw = stopwords.words("english")
sw = sw + ['nan']
sw = sw + ['said'] + ['news'] + ['us'] + ['reuters'] + ['ap'] + ['fox'] + ['cnn'] + ['breitbart']

# Two useful regex
whitespace_pattern = re.compile(r"\s+")
hashtag_pattern = re.compile(r"^#[0-9a-zA-Z]+")

def remove_stop(tokens) :
    # modify this function to remove stopwords

    return[t for t in tokens if t not in sw]

def remove_punctuation(text, punct_set=tw_punct) :
    return("".join([ch for ch in text if ch not in punct_set]))

def tokenize(text) :
    """ Splitting on whitespace rather than the book's tokenize function. That
        function will drop tokens like '#hashtag' or '2A', which we need for Twitter. """

    return([item.lower() for item in whitespace_pattern.split(text)])

def remove_url(text):
    return(re.sub(r'http\S+', '', text))

def remove_messy(text): # remove words that give away the source
    text1=re.sub(r'cnn', '', text)
    text2=re.sub(r'fox', '', text1)
    text3=re.sub(r' – ', '', text2)
    text4=re.sub(r'breitbart', '', text3)
    return(re.sub(r'\\n', '', text4))

# two pipelines to either tokenize or simply remove punctuation
# and lowercase as we will need to extract feature words:

full_pipeline = [str.lower, remove_url, remove_messy, remove_punctuation, tokenize, remove_stop]
first_pipeline = [str.lower, remove_url, remove_messy, remove_punctuation]


def prepare(text, pipeline) :
    tokens = str(text)

    for transform in pipeline :
        tokens = transform(tokens)

    return(tokens)
```

Feature extraction Function:

In [ ]:

```
def conv_features(text,fw) :
     feature_set=dict()
     for word in text.split():
          if word in fw:
               feature_set[word]=True
     return(feature_set)
```

EDA functions:

Get Patterns, Counts, WordCloud:

In [ ]:
```
def get_patterns(text_analyze, num_words, T):
    if(len(text_analyze)==0):
        raise ValueError("Can't work with empty text object")
    total_tokens = 1
    unique_tokens = 0
    avg_token_len = 0.0
    lexical_diversityP = 0.0
    top_words = []

    # Only applying the token_normal, which takes only alphanumeric values
    # to twitter data:
    if T ==1:
        text_analyze=token_normal(text_analyze)

    total_tokens = len(text_analyze)
    unique_tokens = len(set(text_analyze))
    lexical_diversityP = unique_tokens/total_tokens
    avg_token_len = np.mean([len(ta) for ta in text_analyze])

    top_words_1 = collections.Counter(text_analyze)
    top_words = top_words_1.most_common(num_words)

    results={'tokens': total_tokens,
             'unique_tokens': unique_tokens,
             'avg_token_length': avg_token_len,
             'lexical_diversity': lexical_diversityP,
             'top_words': top_words}
    return(results)
```

In [ ]:
```
def wordcloud(word_freq, title=None, max_words=200, stopwords=None):

    wc = WordCloud(width=800, height=400,
                   background_color= "black", colormap="Paired",
                   max_font_size=150, max_words=max_words)

    # convert data frame into dict
    if type(word_freq) == pd.Series:
        counter = Counter(word_freq.fillna(0).to_dict())
    else:
        counter = word_freq

    # filter stop words in frequency counter
    if stopwords is not None:

        counter = {token:freq for (token, freq) in counter.items()
                              if token not in stopwords}
    wc.generate_from_frequencies(counter)

    plt.title(title)

    plt.imshow(wc, interpolation='bilinear')
    plt.axis("off")

# Here, we only apply splitting to the lyrics data due to the difference
# in dataframe/data ingestion between twitter and lycis data:

#def count_words(df, column='tokens', preprocess=None, min_freq=2, split=0):
def count_words(x, preprocess=None, min_freq=2, split=0):

    # process tokens and update counter
    def update(doc):
        tokens = doc if preprocess is None else preprocess(doc)
        counter.update(tokens)

    # create counter and run through all data
    #counter = collections.Counter()
    #top_words_1 = collections.Counter(text_analyze)
    #top_words = top_words_1.most_common(num_words)
    if split == 0:
        counter = collections.Counter(x)
```

```
    else:
        counter = collections.Counter(x.split())

    #df[column].map(update)

    # transform counter into data frame
    freq_df = pd.DataFrame.from_dict(counter, orient='index', columns=['freq'])
    freq_df = freq_df.query('freq >= @min_freq')
    freq_df.index.name = 'token'

    return freq_df.sort_values('freq', ascending=False)
```

In [ ]:
```
def display_topics(model, features, no_top_words=5):
    for topic, words in enumerate(model.components_):
        total = words.sum()
        largest = words.argsort()[::-1] # invert sort order
        print("\nTopic %02d" % topic)
        for i in range(0, no_top_words):
            print("  %s (%2.2f)" % (features[largest[i]], abs(words[largest[i]]*100.0/total)))
```

Load and Preprocess Data:

In [ ]:
```
api_data_complete_df=pd.read_csv('../data/master.csv')
```

Apply tokenization/clean data columns:

In [ ]:
```
api_data_complete_df=api_record_df_latest.copy()

# Tokenize text:

api_data_complete_df['tokens']= api_data_complete_df['article_text'].apply(prepare,
                                                    pipeline=full_pipeline)

# Clean data into lowercase/no punctuation:

api_data_complete_df['cleaner_text']= api_data_complete_df['article_text'].apply(prepare,
                                                    pipeline=first_pipeline)
```

In [ ]:
```
api_data_complete_df['word_count_tokens'] = api_data_complete_df['tokens'].apply(lambda x: len(str(x).split
api_data_complete_df['word_count'] = api_data_complete_df['article_text'].apply(lambda x: len(str(x).split(
```

In [ ]:
```
# Removing Reuters, ms, and Associated Press authors from data:

api_data_complete_df2 = api_data_complete_df[~api_data_complete_df['author'].isin(['msn', 'Associated Press

display(api_data_complete_df2.head())
```

EDA:

Word Clouds

In [ ]:
```
# Separate Left and Right Lean articles to analyze as groups:

Left_lean_articles_df=api_data_complete_df2[api_data_complete_df2['Political_Lean']=="Left"]
Right_lean_articles_df=api_data_complete_df2[api_data_complete_df2['Political_Lean']=="Right"]
```

In [ ]:
```
# Generate list from Left an Right dataframes:

Left_text=[token for sublist in
           Left_lean_articles_df['tokens']
           for token in sublist]
Right_text=[token for sublist in
            Right_lean_articles_df['tokens']
            for token in sublist]
```

In [ ]:
```
#Left_text
```

In [ ]:
```
# Left lean vs Right lean word clouds:

Left_Lean_counts = collections.Counter(Left_text)


Right_Lean_counts = collections.Counter(Right_text)



print("\nLeft Leaning Article's top 5 words:\n")
for HT, count in Left_Lean_counts.most_common(5):
    print(f"{HT}: {count}")

print("\nRight Leaning Article's top 5 words:\n")
for HT, count in Right_Lean_counts.most_common(5):
    print(f"{HT}: {count}")
```

```
Left Leaning Article's top 5 words:

would: 3591
trump: 3442
also: 2774
house: 2566
people: 2507

Right Leaning Article's top 5 words:

biden: 4453
president: 3767
also: 3621
would: 3512
people: 3375
```

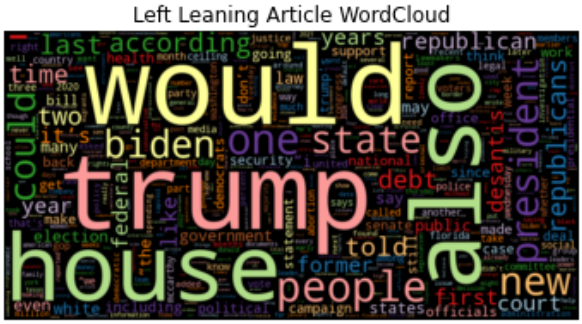
In [ ]:
```
Left_Lean_counts = count_words(Left_text, split=0)
Right_Lean_counts = count_words(Right_text, split=0)
display(Left_Lean_counts)
```

token	freq
would	3591
trump	3442
also	2774
house	2566
people	2507
...	...
paints	2
houghton	2
912	2
k5	2
andreeva	2

27475 rows × 1 columns

In [ ]:
```
#her_df=lyrics_data_df.loc[0,:]
#robyn_df=lyrics_data_df.loc[1,:]
```

In [ ]:
```
wordcloud(Left_Lean_counts['freq'], title="Left Leaning Article WordCloud", max_words=500)
```



In [ ]:
```
wordcloud(Right_Lean_counts['freq'], title="Right Leaning Article WordCloud", max_words=500)
```

Word Clouds reveal that both Left and Right leaning articles tend to have a high word count for their opposition; Left focusing on Trump while the Right focuses on the current president Biden.

Topic Modeling NMF:

In [ ]:
```
Left_tfidf_topic = TfidfVectorizer(stop_words=list(sw), min_df=5, max_df=0.7, ngram_range=(1,2))
Left_topic_modeling_input = Left_tfidf_topic.fit_transform(Left_lean_articles_df['cleaner_text'])
```

In [ ]:
```
Right_tfidf_topic = TfidfVectorizer(stop_words=list(sw), min_df=5, max_df=0.7, ngram_range=(1,2))
Right_topic_modeling_input = Right_tfidf_topic.fit_transform(Right_lean_articles_df['cleaner_text'])
```

In [ ]:
```
nmf_text_model_newsL = NMF(n_components=3, random_state=314)
Left_text_matrix = nmf_text_model_newsL.fit_transform(Left_topic_modeling_input)
HLeft_text_matrix = nmf_text_model_newsL.components_
```

In [ ]:
```
nmf_text_model_newsR = NMF(n_components=3, random_state=314)
Right_text_matrix = nmf_text_model_newsR.fit_transform(Right_topic_modeling_input)
HRight_text_matrix = nmf_text_model_newsR.components_
```

Display Topics for Left vs Right:

In [ ]:
```
display_topics(nmf_text_model_newsL, Left_tfidf_topic.get_feature_names_out())
```

```
Topic 00
  trump (1.75)
  desantis (1.14)
  president (0.32)
  campaign (0.32)
  former (0.31)

Topic 01
  debt (0.97)
  house (0.63)
  mccarthy (0.61)
  biden (0.60)
  debt ceiling (0.57)

Topic 02
  court (0.21)
  state (0.21)
```

```
  police (0.15)
  people (0.15)
  abortion (0.14)
```

In [ ]:
```
display_topics(nmf_text_model_newsR, Right_tfidf_topic.get_feature_names_out())
```

```
Topic 00
  biden (0.22)
  debt (0.17)
  house (0.16)
  ai (0.14)
  bill (0.13)

Topic 01
  border (1.72)
  migrants (1.12)
  title (0.71)
  title 42 (0.65)
  42 (0.64)

Topic 02
  desantis (1.29)
  trump (1.24)
  percent (0.59)
  president (0.41)
  florida (0.39)
```

Topic Modeling using NMF reveals that the top three topics depending on political lean are:

Left:

- Trump/Desantis relationship
- US debt ceiling
- Abortion Laws

Right:

- Bills to change Bill of Rights and regulate AI
- Title 42/Migrants
- Relationship DeSantis and Trump

Topic Modeling LDA:

In [ ]:
```
Lcount_text_vectorizer = CountVectorizer(stop_words=list(sw), min_df=5, max_df=0.7)
Left_count_text_vectors = Lcount_text_vectorizer.fit_transform(Left_lean_articles_df['cleaner_text'])
Left_count_text_vectors.shape

Rcount_text_vectorizer = CountVectorizer(stop_words=list(sw), min_df=5, max_df=0.7)
Right_count_text_vectors = Rcount_text_vectorizer.fit_transform(Right_lean_articles_df['cleaner_text'])
Right_count_text_vectors.shape
```

```
(2758, 13927)
```

In [ ]:
```
Left_lda_para_model = LatentDirichletAllocation(n_components = 4, random_state=42)
W_lda_para_matrix_Left = Left_lda_para_model.fit_transform(Left_count_text_vectors)
H_lda_para_matrix_Left = Left_lda_para_model.components_
```

In [ ]:
```
Right_lda_para_model = LatentDirichletAllocation(n_components = 4, random_state=42)
W_lda_para_matrix_Right = Right_lda_para_model.fit_transform(Right_count_text_vectors)
H_lda_para_matrix_Right = Right_lda_para_model.components_
```

In [ ]:
```
display_topics(Left_lda_para_model, Lcount_text_vectorizer.get_feature_names_out())
```

```
Topic 00
  people (0.77)
  health (0.58)
```

```
  state (0.57)
  new (0.48)
  year (0.42)

Topic 01
  house (1.17)
  biden (1.06)
  debt (1.04)
  could (0.60)
  mccarthy (0.57)

Topic 02
  trump (2.13)
  desantis (0.78)
  president (0.76)
  election (0.63)
  state (0.58)

Topic 03
  court (0.62)
  people (0.45)
  police (0.39)
  according (0.37)
  new (0.33)
```

In [ ]:
```
display_topics(Right_lda_para_model, Rcount_text_vectorizer.get_feature_names_out())
```

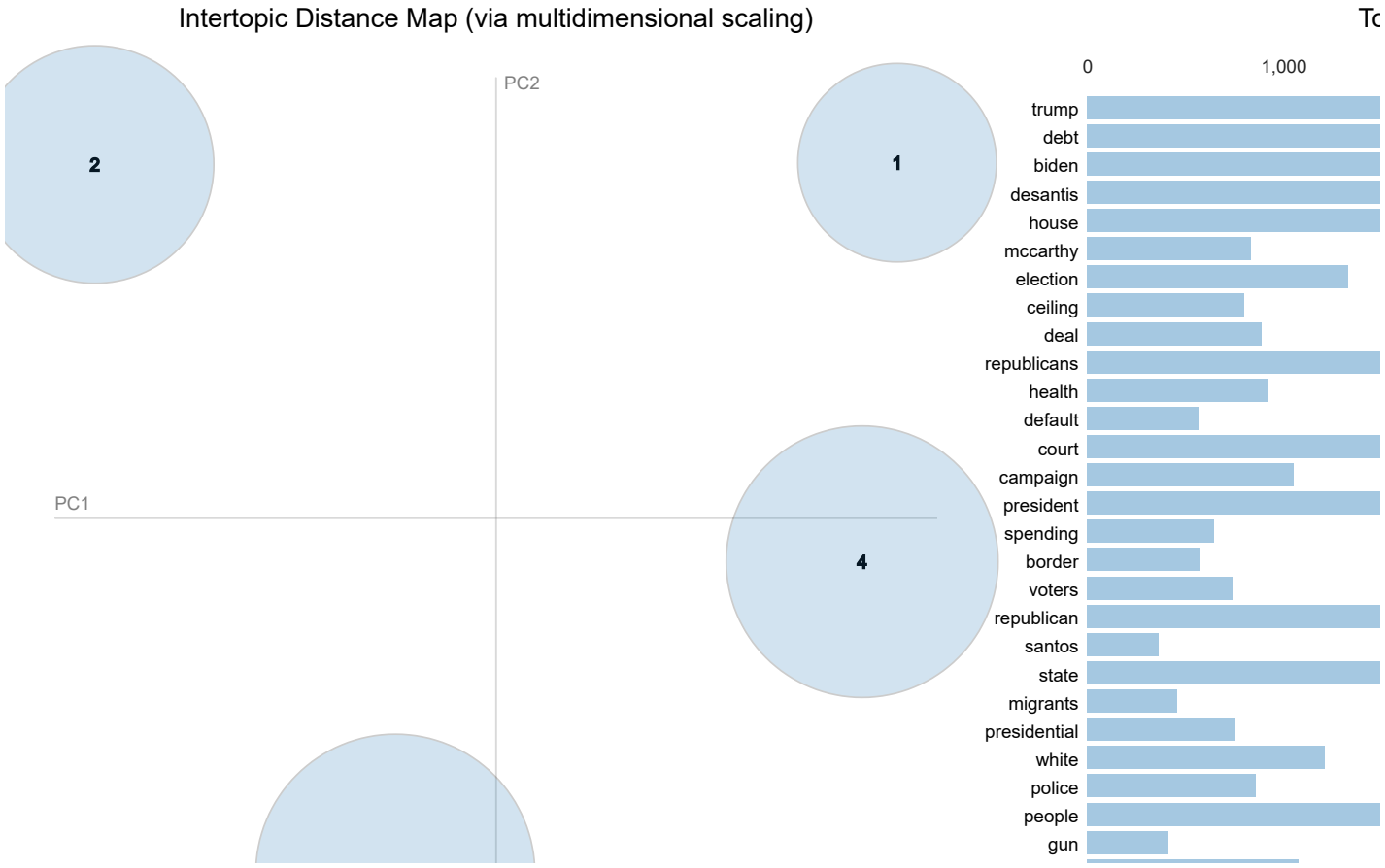
```
Topic 00
  also (0.44)
  people (0.44)
  would (0.42)
  one (0.39)
  school (0.37)

Topic 01
  trump (1.18)
  president (0.98)
  biden (0.80)
  desantis (0.71)
  former (0.51)

Topic 02
  police (0.79)
  biden (0.46)
  told (0.45)
  fbi (0.45)
  according (0.42)

Topic 03
  border (1.66)
  migrants (0.96)
  biden (0.92)
  texas (0.62)
  administration (0.62)
```

In [ ]:
```
lda_display_left = pyLDAvis.lda_model.prepare(Left_lda_para_model, Left_count_text_vectors, Lcount_text_vec
pyLDAvis.display(lda_display_left)
```

Selected Topic: 0 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric:(2)

λ = 1



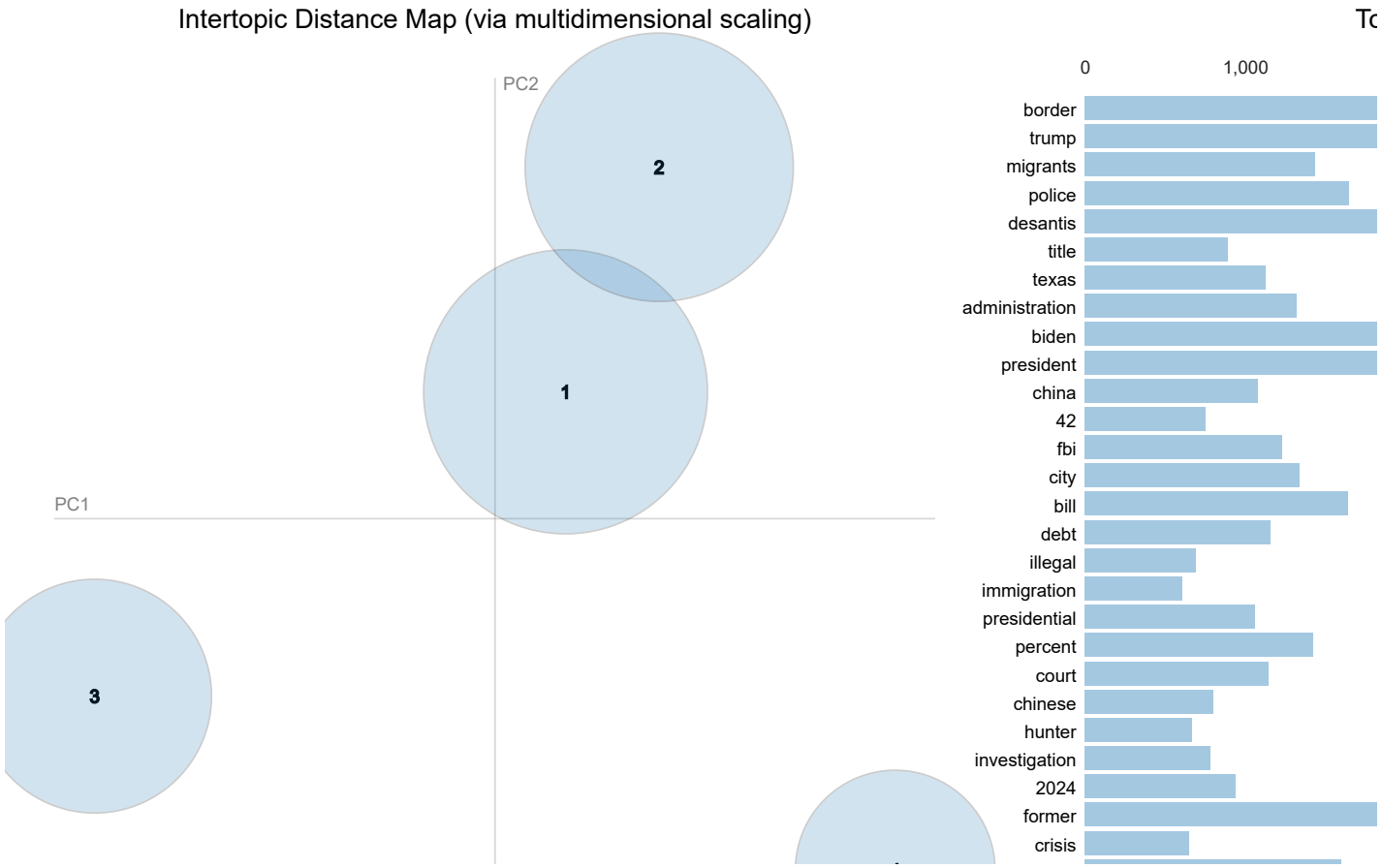
In [ ]:

```
lda_display_right = pyLDAvis.lda_model.prepare(Right_lda_para_model, Right_count_text_vectors, Rcount_text_
pyLDAvis.display(lda_display_right)
```

Selected Topic: 0 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric:(2)

λ = 1



Topic Modeling using LDA reveals that the top three topics depending on political lean are:

Left:

- Healthcare
- McCarthy/Biden Debt Ceiling
- Trump/DeSantis relationship
- Justice system, police

Right:

- Unclear/school-related (the visualization shows this topic overlaps with the following one a bit)
- President, Biden, Trump, politics
- Justice system, fbi
- Immigration

---

509 Final Project

The notebook is for Exploratory Data Analysis (EDA), text data preprocessing, modeling, and evaluation.

Globally import libraries

In [1]:
```
from bs4 import BeautifulSoup
from collections import defaultdict, Counter
import datetime as dt
import emoji
import itertools
import json
import logging
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
import pickle
import pymysql as mysql
import random
import re
import regex as rex
import requests
import shutil
from string import punctuation
import time
from tqdm import tqdm
import zipfile

import nltk
from nltk.corpus import stopwords
import spacy

from skopt import BayesSearchCV
from skopt.space import Real, Categorical, Integer

from sklearn.feature_extraction.text import TfidfTransformer, \
CountVectorizer, TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.pipeline import make_pipeline, Pipeline
from sklearn import metrics
from sklearn.metrics import make_scorer, f1_score, classification_report, \
confusion_matrix, ConfusionMatrixDisplay, RocCurveDisplay

import textacy.preprocessing as tprep
from textacy.extract import keyword_in_context

# Set pandas global options
```

```
pd.options.display.max_rows = 17
pd.options.display.precision = 4
np.set_printoptions(suppress=True, precision=4)

%matplotlib inline
```

Upload data from CSV

In [2]:
```
'''Dir nav citation:
https://softhints.com/python-change-directory-parent/'''
curr_dir = os.path.abspath(os.curdir)
print(curr_dir)
os.chdir("..")
up1_dir = os.path.abspath(os.curdir)
print(up1_dir)
```

```
C:\Users\acarr\Documents\GitHub\ADS509_Final_project\deliverables
C:\Users\acarr\Documents\GitHub\ADS509_Final_project
```

In [3]:
```
# change `data_location` to the location of the folder on your machine.
data_location = 'data'

file_in_name01 = 'master.csv'
file_in_name02 = 'master_business_TheHill.csv'

file_in_path01 = os.path.join(up1_dir, data_location, file_in_name01)
file_in_path02 = os.path.join(curr_dir, file_in_name02)

print(f'CSV file 1 in path: {file_in_path01}')
print(f'CSV file 2 in path: {file_in_path02}')
```

```
CSV file 1 in path: C:\Users\acarr\Documents\GitHub\ADS509_Final_project\data\maste
r.csv
CSV file 2 in path: C:\Users\acarr\Documents\GitHub\ADS509_Final_project\deliverable
s\master_business_TheHill.csv
```

Review dataframe

In [4]:
```
slct_tbl_full_df01 = pd.read_csv(file_in_path01)
print(f'Dataframe shape: {slct_tbl_full_df01.shape}')
display(slct_tbl_full_df01.head())
```

```
Dataframe shape: (4509, 7)
```

	source_name	author	title	url	publis
0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/...	20 30T16:
1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0...	20 30T19:
2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/...	20 30T23:
3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0...	20 30T18:
4	The Washington Post	NaN	The revolt of Christian home-schoolers...	https://www.washingtonpost.com/education/inter...	20 30T18:

Exploratory Data Analysis (EDA)

Count missing `article_text` feature

The majority of null values appear in the `content` column. There are also several in `author` and one in `article_text`. Neither `content` nor `author` will be used for current modeling efforts, therefore they are not a factor. The one instance with missing article text will be removed.

In [5]:
```
count_nan = slct_tbl_full_df01.isnull().sum()

# printing the number of values present
# in the column
print('Number of NaN values present: ' + str(count_nan))
```

```
Number of NaN values present: source_name        0
author            37
title              0
url                0
publish_date       0
content         3351
article_text       1
dtype: int64
```

Count blank `article_text` feature

In [6]:
```
print(len(slct_tbl_full_df01[slct_tbl_full_df01['article_text']=='']))
display(slct_tbl_full_df01[slct_tbl_full_df01['article_text']=='].head(20))
```

0

source_name	author	title	url	publish_date	content	article_text

Remove missing `article_text` row(s)

In [7]:
```
'''Drop missing citation:
https://pandas.pydata.org/pandas-docs/stable/reference
/api/pandas.DataFrame.dropna.html#pandas.DataFrame.dropna'''
slct_tbl_full_df02 = slct_tbl_full_df01.dropna(subset=['article_text'])
print(f'Dataframe shape: {slct_tbl_full_df02.shape}')
display(slct_tbl_full_df02.head())
```

```
Dataframe shape: (4508, 7)
```

	source_name	author	title	url	pub
0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/...	30T
1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0...	30T
2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/...	30T
3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0...	30T
5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0...	30T

Count characters and words for initial review

In [8]:
```
tqdm.pandas(ncols=50)  # can use tqdm_gui, optional kwargs, etc
# Now you can use `progress_apply` instead of `apply`

# Raw text character and word counts
slct_tbl_full_df02['char_cnt'] = slct_tbl_full_df02['article_text']\
.progress_apply(len)
slct_tbl_full_df02['word_cnt'] = slct_tbl_full_df02['article_text']\
.progress_apply(lambda x: len(x.split()))
display(slct_tbl_full_df02.head())
```

```
100%|██████| 4508/4508 [00:00<00:00, 644002.81it/s]
C:\Users\acarr\AppData\Local\Temp\ipykernel_21224\2936833956.py:5: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  slct_tbl_full_df02['char_cnt'] = slct_tbl_full_df02['article_text']\
100%|██████| 4508/4508 [00:00<00:00, 20923.57it/s]
C:\Users\acarr\AppData\Local\Temp\ipykernel_21224\2936833956.py:7: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  slct_tbl_full_df02['word_cnt'] = slct_tbl_full_df02['article_text']\
```

	source_name	author	title	url	pub
0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/...	30T
1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0...	30T
2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/...	30T
3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0...	30T
5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0...	30T

Descriptive statistics

Stats are displayed for both categorical and numerical columns. As expected "Fox News" is the most frequent value in `source_name` as the most articles were collected from that news site. The inclusion of "Associated Press" as the mode for `author` identified it as a potential source for skew in the final results, as AP was rated as a "center" source in the AllSide Media Bias Chart. As a result, all articles with an `author` value of "Associated Press" were removed; similarily, articles by "msn" and "Reuters" were also removed.
For the numerical values, there was a very large range for both character and word counts (80,454 and 14,306, respectively), but also a large delta between the 75% percentile and max (74,920.5 and 13,433, respectively), indicating a distribution with a very long right tail with a very small amount of some extremely long (outlier) articles. As a result, the standard deviation was also quite large relative to the mean. For the current analyses, no additional efforts will be performed to account for outliers, but this will be an examination factor for future expansion/comparative studies.

In [9]:
```
slct_tbl_full_df02[['source_name',
                    'author',
                    'publish_date',
                    'article_text']].describe(include="O").T
```

Out[9]:

	count	unique	top	freq
source_name	4508	4	Fox News	2192
author	4472	956	Associated Press	450
publish_date	4508	4486	2023-05-13T11:00:00Z	3
article_text	4508	4508	Travelers in Alabama driving on Interstate 65 ...	1

In [10]:
```
slct_tbl_full_df02.describe().T
```

Out[10]:

	count	mean	std	min	25%	50%	75%	max
char_cnt	4508.0	4655.5011	3137.3650	131.0	2832.0	3951.5	5664.5	80585.0
word_cnt	4508.0	731.0315	518.5765	16.0	432.0	607.0	889.0	14322.0

Display Source counts

In [11]:
```
slct_tbl_full_df02['source_name'].value_counts()
```

Out[11]:
```
Fox News               2192
Breitbart News         1017
CNN                     773
The Washington Post     526
Name: source_name, dtype: int64
```

Examine inclusion of "centrist" sources indicated by `author` feature

In [12]:
```
slct_tbl_full_df02a = slct_tbl_full_df02[slct_tbl_full_df02['author']\
                                         .isin(['msn',
                                                'Associated Press',
                                                'Reuters'])]

display(slct_tbl_full_df02a[slct_tbl_full_df02a['author']=='msn'])

display(slct_tbl_full_df02a.groupby(by=['source_name', 'author']).count())
```

	source_name	author	title	url	pu
17	The Washington Post	msn	State Dept seeks to expand space diplomacy...	https://www.washingtonpost.com/technology/2023...	30
18	The Washington Post	msn	SHOCK IN RUSSIAN CAPITAL	https://www.washingtonpost.com/world/2023/05/3...	30
22	The Washington Post	msn	Debate over whether AI will destroy us is divi...	https://www.washingtonpost.com/technology/2023...	20
81	The Washington Post	msn	Corporate bankruptcies creeping up as pressure...	https://www.washingtonpost.com/business/2023/0...	23
84	The Washington Post	msn	The looming existential crisis for cable news...	https://www.washingtonpost.com/media/2023/05/2...	23
...	...	...	...	...	
492	The Washington Post	msn	Biden shows growing appetite to cross Putin's ...	https://www.washingtonpost.com/national-securi...	01
502	The Washington Post	msn	Behind-the-scenes videos of Tucker Carlson wer...	https://www.washingtonpost.com/media/2023/06/0...	02
503	The Washington Post	msn	Georgia probe of Trump broadens to activities ...	https://www.washingtonpost.com/nation/2023/06/...	02
506	The Washington Post	msn	DRAMA: Couple, both nurses, save man's life mi...	https://www.washingtonpost.com/lifestyle/2023/...	02

	source_name	author	title	url	pu
509	The Washington Post	msn	'DRAG RACE' queen says cancellation of militar...	https://www.washingtonpost.com/nation/2023/06/...	02

25 rows × 9 columns

		title	url	publish_date	content	article_text	char_cnt	word_cnt
source_name	**author**							
CNN	**Reuters**	6	6	6	1	6	6	6
Fox News	**Associated Press**	450	450	450	73	450	450	450
	Reuters	1	1	1	0	1	1	1
The Washington Post	**msn**	25	25	25	25	25	25	25

In [13]:
```
counter = Counter(slct_tbl_full_df02['author'])

word_cutoff = 5
con_feature_words = set()

for word, count in counter.items():
    if count > word_cutoff:
        con_feature_words.add(word)

print(f'''With a word cutoff of {word_cutoff}, we have
{len(con_feature_words)} words as features in the model.''')
print(con_feature_words)
```

```
With a word cutoff of 5, we have
151 words as features in the model.
{nan, 'Paul Bois, Paul Bois', 'Michael Ruiz', 'Hannah Bleau, Hannah Bleau', 'Peter C
addle, Peter Caddle', 'Kristine Parks', 'Gabriel Hays', 'Brianna Herlihy', 'Lindsay
Kornick', 'Anders Hagstrom', 'Peter Aitken', 'Audrey Conklin', 'Azi Paybarah', 'Nick
Gilbertson, Nick Gilbertson', 'Brie Stimson', 'Kurt Knutsson, CyberGuy Report', 'Ale
xandra Meeks', 'Hannah Knowles', 'Stephen Collinson', 'Jeffrey Clark', 'Hannah Ray L
ambert', 'Jeff Stein', 'Jacob Bliss, Jacob Bliss', 'Bob Price, Bob Price', 'Madeline
Coggins', 'Andrew Miller', 'Ashley Carnahan', 'Kurt Zindulka, Kurt Zindulka', 'Jeff
Poor, Jeff Poor', 'Sean Lyngaas', 'Paulina Dedaj', 'Joe Schoffstall', 'Spencer S. Hs
u', 'Elizabeth Heckman', 'Zachary B. Wolf', 'Kevin Liptak', 'Rebecca Rosenberg', 'Th
omas Catenacci', 'Greg Norman', 'Matthew Boyle, Matthew Boyle', 'Chad Pergram', 'War
ner Todd Huston, Warner Todd Huston', 'Danielle Wallace', 'Aaron Kliegman', 'Pam Ke
y, Pam Key', 'Haley Chi-Sing', 'Houston Keene', 'Kyle Morris', 'Timothy Nerozzi', 'E
laine Mallon, Elaine Mallon', 'Amy B Wang', 'Mariana Alfaro', 'Amy Furr, Amy Furr',
'Adam Sabes', 'Ian Hanchett, Ian Hanchett', 'John Hayward, John Hayward', 'Paul Wald
man', 'Angelica Stabile', 'Wendell Husebø, Wendell Husebø', 'Sean Moran, Sean Mora
n', 'Ryan Morik', 'Joseph Wulfsohn', 'Jordan Dixon-Hamilton, Jordan Dixon-Hamilton',
'Paul Steinhauser', 'John Wagner', 'Maeve Reston', 'Hannah Rabinowitz', 'Patrick Hau
f', 'Nicole Goodkind', 'Stephen Sorace', 'Andrea Vacchiano', 'Julia Musto', 'Melissa
Rudy', 'Hanna Panreck', 'Elizabeth Elkind', 'msn', 'Charles Creitz', 'Oliver Darcy',
'Nadeen Ebrahim', 'Frances Martel, Frances Martel', 'Alisha Ebrahimji', 'Kassy Dillo
n', 'Jennifer Rubin', 'Oliver JJ Lane, Oliver JJ Lane', 'Lucas Nolan, Lucas Nolan',
'Aubrie Spady', 'Lawrence Richard', 'Jessica Chasmar', 'Fox News Staff', 'Christian
K. Caruzo, Christian K. Caruzo', 'Bailee Hill', 'Brian Flood', 'AWR Hawkins, AWR Haw
kins', 'Philip Bump', 'Brian Fung', 'Kendall Tietz', 'Breitbart London, Breitbart Lo
ndon', 'Kerry Byrne', 'Ashley Oliver, Ashley Oliver', 'Katherine Hamilton, Katherine
Hamilton', 'Greg Gutfeld', 'Fox News', 'John Binder, John Binder', 'Yael Halon', 'Mi
chael Lee', 'Adam Shaw', 'Emma Colton', 'Alexander Hall', 'Reuters', 'Aaron Blake',
'Chris Eberhart', 'Alana Mastrangelo, Alana Mastrangelo', 'Glenn Kessler', 'Howard K
urtz', 'Jon Brown', 'Steve Contorno', 'Robert Barnes', 'Devlin Barrett', 'Ariane de
Vogue', 'Eric Bradner', 'Associated Press', 'Deirdre Reilly', 'David Ng, David Ng',
'Tierney Sneed', 'Taylor Penley', 'Matt Egan', 'Brooke Singman', 'Paul Kane', 'Caitl
in McFall', 'Chantz Martin', 'Tony Romm', 'Louis Casiano', 'Joshua Klein, Joshua Kle
in', 'Allum Bokhari, Allum Bokhari', 'Bradford Betz', 'Landon Mion', 'Dylan Gwinn, D
ylan Gwinn', 'Kristina Wong, Kristina Wong', 'Chris Pandolfo', 'Joshua Nelson', 'Tam
i Luhby', 'Brandon Gillespie', 'Sarah Rumpf-Whitten', 'Peter Kasperowicz', 'Greg Weh
ner', 'Ryan Gaydos', 'Neil Munro, Neil Munro', 'Joel B. Pollak, Joel B. Pollak', 'Si
mon Kent, Simon Kent', 'John Nolte, John Nolte', 'Hannah Grossman'}
```

Assign class based on `source_name` and AllSides Media Bias Chart

In [14]:

```
slct_tbl_full_df03 = slct_tbl_full_df02[~slct_tbl_full_df02['author']\
                                        .isin(['msn',
                                               'Associated Press',
                                               'Reuters'])]
slct_tbl_full_df03 = slct_tbl_full_df03.reset_index()
slct_tbl_full_df03['political_lean'] = 'right'
print(slct_tbl_full_df03.shape)
display(slct_tbl_full_df03.head())
slct_tbl_full_df03.loc[(slct_tbl_full_df03['source_name'] \
                        == 'The Washington Post') \
                       | (slct_tbl_full_df03['source_name'] \
                          == 'CNN'), 'political_lean'] = 'left'
```

```
display(slct_tbl_full_df03.head())

display(slct_tbl_full_df03['political_lean'].value_counts())
```

```
(4026, 11)
```

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

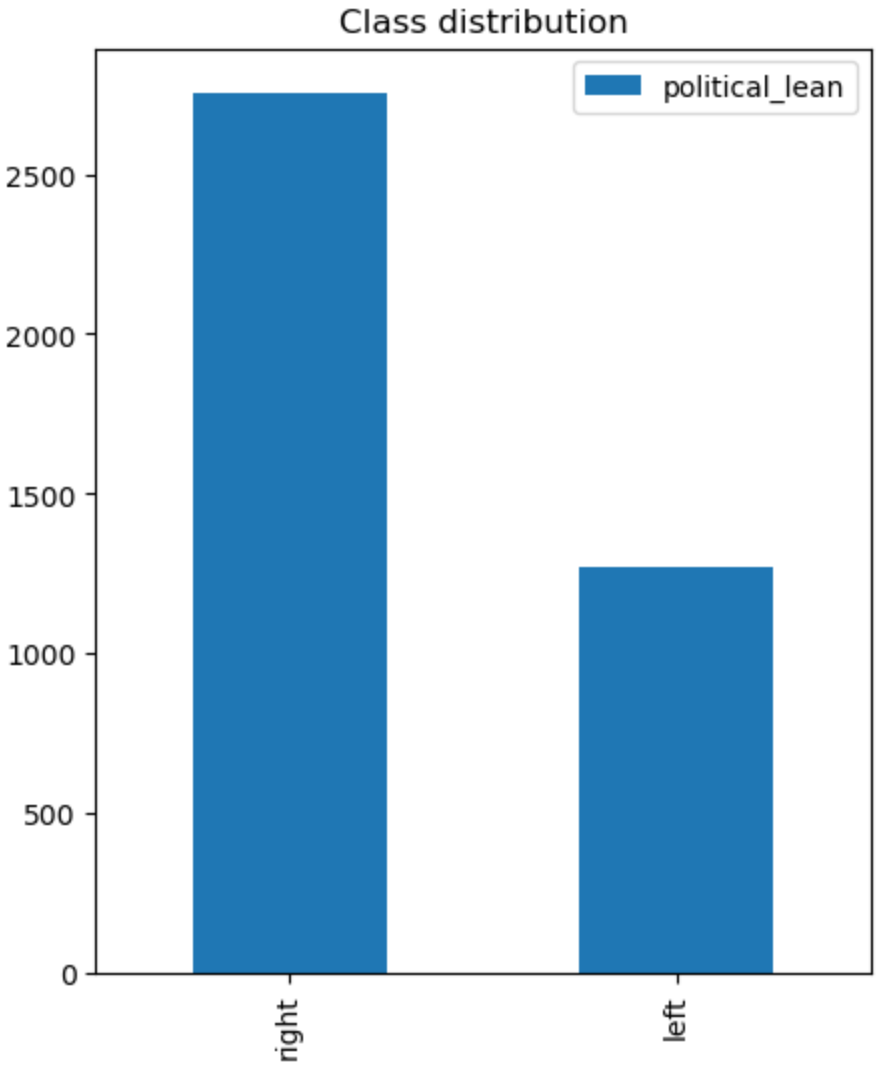
```
right    2758
left     1268
```

Visualize class distribution

There is definitely an imbalance in the number of instances in each class. This is due to Fox News being the most prolific source, whether because they put out a lot more articles or their sites were more consistently available for scraping. This imbalance is not considered extreme and will not be adjusted for within the scope of the current study.

In [15]:
```
slct_tbl_full_df03['political_lean'].value_counts().plot(kind="bar",
                                                         legend=True,
                                                         figsize=(5,6),
                                                        title='Class distribution')
```

Out[15]:
```
<Axes: title={'center': 'Class distribution'}>
```

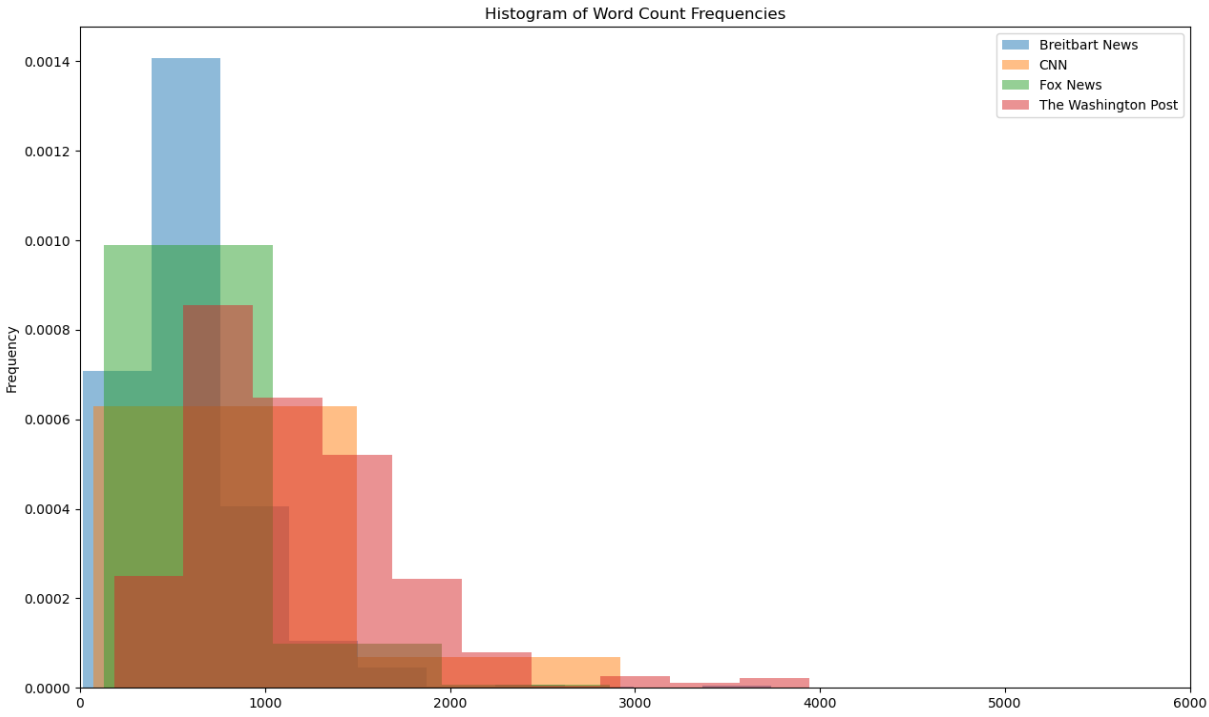


Plot word counts

All sources seem to have very similar consolidation of most frequent word counts between 0 and 2,000. However, the two "left" sources (CNN and The Washington Post) seem to be the significant source of the outliers, with a small amount of articles each that have extremely large word counts (*note:* the x-axis range was truncated at 6,000 to make it more readabla--as noted above, there were some articles with word counts greater than 14,000). Given the simialirties between the sources within each class, the differences may correlate to intentional word limitation based on perceived audience desires, but in the very least do add evidence that the sources have been grouped together appropriately.

In [16]:
```
slct_tbl_full_df03.groupby('source_name')['word_cnt'].plot(kind="hist",
                                                           density=True,
                                                           alpha=0.5,
                                                           legend=True,
                                                           figsize=(15,9),
                                    title='Histogram of Word Count Frequencies',
                                                           xlim=(0,6000))
```

Out[16]:
```
source_name
Breitbart News         Axes(0.125,0.11;0.775x0.77)
CNN                    Axes(0.125,0.11;0.775x0.77)
Fox News               Axes(0.125,0.11;0.775x0.77)
The Washington Post    Axes(0.125,0.11;0.775x0.77)
Name: word_cnt, dtype: object
```



Data preprocessing

In [17]:
```
def uniq_tok(df_col=None):
    '''Display all unique tokens across all instances'''
    df_cols1 = pd.Series(df_col)

    all_tokens_lst01 = []

    [all_tokens_lst01.append(f) for f in df_cols1]
    all_tokens_lst01 = list(itertools.chain.from_iterable(all_tokens_lst01))
    all_tokens_set01 = set(all_tokens_lst01)
    print(len(sorted(all_tokens_set01)))
    print(sorted(all_tokens_set01))
```

In [18]:
```
slct_tbl_full_df04 = slct_tbl_full_df03.copy()
```

Case-loading

In [19]:
```
slct_tbl_full_df03['lower'] = slct_tbl_full_df03['article_text']\
.apply(str.lower)

print(slct_tbl_full_df03.shape)
display(slct_tbl_full_df03.head())
```

```
(4026, 12)
```

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

Text normalization

Create function

In [20]:
```
def normalize(text):
    text = tprep.normalize.hyphenated_words(text)
    text = tprep.normalize.quotation_marks(text)
    text = tprep.normalize.unicode(text)
    text = tprep.remove.accents(text)
    return text
```

Call function

In [21]:
```
slct_tbl_full_df03['norm'] = slct_tbl_full_df03['lower'].apply(normalize)

print(slct_tbl_full_df03.shape)
display(slct_tbl_full_df03.head())
```

```
for c in range(0,1):
    try:
        print(slct_tbl_full_df03['norm'][c], '\n')
    except:
        print(f'Skip {c}')
```

```
(4026, 13)
```

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

travelers in alabama driving on interstate 65 to parties and barbecues on memorial d
ay might have seen messages on digital road signs honoring veterans who died fightin
g for the united states. but that's not what some drivers near clanton, ala., saw on
monday. instead, motorists reported seeing a sign that was apparently hacked to disp
lay the words "reclaim america," a white nationalist slogan, and "patriot front us,"
referencing the white supremacist group that was involved in the deadly 2017 unite t
he right rally in charlottesville. "how does this come about?" wrote sarah hughes, a
motorist who captured photos of the sign and posted them on twitter. "weird as hel
l." a contractor's portable message board was hacked on i-65 in chilton county, al
a., on monday afternoon, john mcwilliams, a spokesman for the alabama department of
transportation (aldot) west central region, told the washington post in a statement.
"a citizen alerted a nearby state trooper about the message, who then contacted aldo
t," mcwilliams said tuesday. "aldot personnel immediately responded and turned the m
essage board off. no other message boards on i-65 were affected." mcwilliams added
that aldot is investigating how the white supremacist language appeared on the sign
near clanton, about 40 miles northwest of montgomery, ala. officials have given no i
mmediate indication of who is responsible for apparently hacking the interstate sig
n. the news was first reported by al.com. hughes told the post that she was driving
home to birmingham from a weekend at alabama's gulf coast when she saw the white sup
remacist messages that have recently popped up around her home city from supporters
of patriot front. "when i saw it, i thought, 'oh, it's the same guys,' " said hughe
s, a 31-year-old attorney. "i was kind of shocked." the hacked alabama road sign com
es at a time when president biden has declared white supremacy "the most dangerous t
errorist threat" to the country. during his commencement address at howard universit
y this month, biden told the graduating class at the historically black university t
hat he pledged "to stand up against the poison of white supremacy, as i did in my in
augural address — to single it out as the most dangerous terrorist threat to our hom
eland is white supremacy." "i don't have to tell you that progress toward justice of
ten meets ferocious pushback from the oldest and most sinister of forces," biden sai
d in the may 13 address, after quoting donald trump's equivocating response to the 2
017 rally in charlottesville that killed 32-year-old heather heyer and injured 19 ot
hers. "that's because hate never goes away." biden calls white supremacy greatest te
rrorism threat as 2024 race heats up the southern poverty law center (splc) tracked
at least 13 hate groups in alabama in 2021, including the proud boys. the discussion
surrounding white supremacists and white nationalists in alabama intensified this mo
nth after sen. tommy tuberville (r-ala.) said that people identified as "white extre
mists" and white nationalists should be allowed to serve in the u.s. armed forces. w
hen asked by a reporter with wbhm in birmingham whether white nationalists should be
allowed to serve in the military, tuberville replied, "well, they call them that. i
call them americans." after tuberville was criticized, a spokesman told the post tha
t the senator "resents the implication that the people in our military are anything
but patriots and heroes." gop senator says of white nationalists in the military, 'i
call them americans' patriot front, the white supremacist group whose name was displ
ayed on the interstate sign, is a texas-based hate group that broke off from vanguar
d america and formed after the charlottesville rally, the splc says. its members hav
e chanted "reclaim america" at rallies in coeur d'alene, idaho, washington and bosto
n in recent years, according to news reports. patriot front is responsible for "the
vast majority of white supremacist propaganda distributed in the united states" sinc
e 2019, according to the anti-defamation league. it's not the first time that langua
ge promoting patriot front has made its way into a public space in alabama. in july,
graffiti beneath a birmingham bridge appeared with "patriot front us" spray-painted
in red and blue letters, al.com reported. other patriot front graffiti has also been
spotted in birmingham, a city with a population that's nearly 70 percent black, acco
rding to u.s. census data. a photo posted to twitter this month showed more patriot
front graffiti along the red mountain expressway in birmingham with the words, "we d
are defend our rights." the patriot front graffiti was later removed, but the messag

e left sydney duncan, the attorney director for the magic legal center in birmingha
m, saddened that hate had become so public in some parts of alabama. "white supremac
y is alive and well," duncan wrote. hughes said she was traveling north to birmingh
am when she pulled over on i-65 to take photos of the messages on the sign. she had
seen confederate monuments and flags on that drive before, but that kind of messagin
g on government-owned property was different, she said. a police officer who was alr
eady at the scene waved at her to keep driving, hughes added. when she returned hom
e, hughes said she felt compelled to share the images due to the ongoing conversatio
n happening among birmingham residents about the promotion of patriot front in publi
c spaces. "some people might perceive this as upsetting and scary, and a sign of the
worsening of our country," she said. "but if this is their strategy, then i'm not re
ally impressed." she added, "they're a dying breed." toluse olorunnipa and azi payba
rah contributed to this report.

Remove special characters

Create function

In [22]:
```
rex_sep = rex.compile(r' ')
rex_ucode = rex.compile(r'[\\]u20*')

'''re.sub lambda citation:
https://chat.openai.com/share/402ec66e-2802-4cda-af8c-6f9f5b097d85
'''
sep_lst = []
ucode_lst = []
# Add leading and trailing space to URLs
def rex_replace(text):
    #txt = str(text)
    #print(lambda x: x.replace(' ', ' '))
    #sep_lst.append(rex_sep.findall(txt))
    #ucode_lst.append(rex_ucode.findall(txt))
    text = text.replace(r' ', ' ').replace(r'-', ' ')\
    .replace(r'\n', ' ').replace('\u2063', ' ').replace('\u2066', ' ')\
    .replace('\u2069', ' ').replace('\u200b', ' ').replace('\u200d', ' ')
    #txt = txt
    #text = text.replace(r'200b', 'd171c')
    #text = rex_ucode.sub('', text)
    return text
```

Call function

In [23]:
```
slct_tbl_full_df03['replace'] = slct_tbl_full_df03['norm'].apply(rex_replace)

#print(ucode_lst)
#print(sep_lst)

print(slct_tbl_full_df03.shape)
display(slct_tbl_full_df03.head())
```

(4026, 14)

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

'''Complex citation (add lambda): https://chat.openai.com/share/a135754c-c38c-47ea-8f83-54d41d5397ab ''' slct_tbl_full_df03['replace'] = slct_tbl_full_df03['norm'].apply(lambda x: x.replace(' ', ' ').replace(r'\n', ' ').replace('\u2063', ' ').replace('\u2066', ' ').replace('\u2069', ' ').replace('\u200b', ' ').replace('\u200d', ' '))

URL RegEx find

Create function

In [24]:
```
rex_url_c = rex.compile(r'http[s]?:[\/]+[\S]*\s')

'''re.sub lambda citation:
https://chat.openai.com/share/402ec66e-2802-4cda-af8c-6f9f5b097d85
'''
# Add leading and trailing space to URLs
def rex_url(text):
```

```
    text = rex_url_c.sub(lambda match: ' ' + match.group(0) + ' ', text)
    return text
```

Call function

In [25]:
```
slct_tbl_full_df03['rex_urls'] = slct_tbl_full_df03['replace'].apply(rex_url)

print(slct_tbl_full_df03.shape)
display(slct_tbl_full_df03.head())
```

```
(4026, 15)
```

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

Separate emojis as individual tokens

Create function

In [26]:
```
def emoji_split(text):
    return("".join([' ' + c + ' ' if emoji.is_emoji(c) else c for c in text]))
```

Call function

In [27]:
```
slct_tbl_full_df03['emoji_split'] = slct_tbl_full_df03['rex_urls']\
.apply(emoji_split)

print(slct_tbl_full_df03.shape)
display(slct_tbl_full_df03.head())

for c in range(0,1):
    try:
        print(slct_tbl_full_df03['emoji_split'][c], '\n')
    except:
        print(f'Skip {c}')
```

(4026, 16)

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

travelers in alabama driving on interstate 65 to parties and barbecues on memorial d
ay might have seen messages on digital road signs honoring veterans who died fightin
g for the united states. but that's not what some drivers near clanton, ala., saw on
monday. instead, motorists reported seeing a sign that was apparently hacked to disp
lay the words "reclaim america," a white nationalist slogan, and "patriot front us,"
referencing the white supremacist group that was involved in the deadly 2017 unite t
he right rally in charlottesville. "how does this come about?" wrote sarah hughes, a
motorist who captured photos of the sign and posted them on twitter. "weird as hel
l." a contractor's portable message board was hacked on i 65 in chilton county, al
a., on monday afternoon, john mcwilliams, a spokesman for the alabama department of
transportation (aldot) west central region, told the washington post in a statement.
"a citizen alerted a nearby state trooper about the message, who then contacted aldo
t," mcwilliams said tuesday. "aldot personnel immediately responded and turned the m
essage board off. no other message boards on i 65 were affected." mcwilliams added
that aldot is investigating how the white supremacist language appeared on the sign
near clanton, about 40 miles northwest of montgomery, ala. officials have given no i
mmediate indication of who is responsible for apparently hacking the interstate sig
n. the news was first reported by al.com. hughes told the post that she was driving
home to birmingham from a weekend at alabama's gulf coast when she saw the white sup
remacist messages that have recently popped up around her home city from supporters
of patriot front. "when i saw it, i thought, 'oh, it's the same guys,' " said hughe
s, a 31 year old attorney. "i was kind of shocked." the hacked alabama road sign com
es at a time when president biden has declared white supremacy "the most dangerous t
errorist threat" to the country. during his commencement address at howard universit
y this month, biden told the graduating class at the historically black university t
hat he pledged "to stand up against the poison of white supremacy, as i did in my in
augural address — to single it out as the most dangerous terrorist threat to our hom
eland is white supremacy." "i don't have to tell you that progress toward justice of
ten meets ferocious pushback from the oldest and most sinister of forces," biden sai
d in the may 13 address, after quoting donald trump's equivocating response to the 2
017 rally in charlottesville that killed 32 year old heather heyer and injured 19 ot
hers. "that's because hate never goes away." biden calls white supremacy greatest te
rrorism threat as 2024 race heats up the southern poverty law center (splc) tracked
at least 13 hate groups in alabama in 2021, including the proud boys. the discussion
surrounding white supremacists and white nationalists in alabama intensified this mo
nth after sen. tommy tuberville (r ala.) said that people identified as "white extre
mists" and white nationalists should be allowed to serve in the u.s. armed forces. w
hen asked by a reporter with wbhm in birmingham whether white nationalists should be
allowed to serve in the military, tuberville replied, "well, they call them that. i
call them americans." after tuberville was criticized, a spokesman told the post tha
t the senator "resents the implication that the people in our military are anything
but patriots and heroes." gop senator says of white nationalists in the military, 'i
call them americans' patriot front, the white supremacist group whose name was displ
ayed on the interstate sign, is a texas based hate group that broke off from vanguar
d america and formed after the charlottesville rally, the splc says. its members hav
e chanted "reclaim america" at rallies in coeur d'alene, idaho, washington and bosto
n in recent years, according to news reports. patriot front is responsible for "the
vast majority of white supremacist propaganda distributed in the united states" sinc
e 2019, according to the anti defamation league. it's not the first time that langua
ge promoting patriot front has made its way into a public space in alabama. in july,
graffiti beneath a birmingham bridge appeared with "patriot front us" spray painted
in red and blue letters, al.com reported. other patriot front graffiti has also been
spotted in birmingham, a city with a population that's nearly 70 percent black, acco
rding to u.s. census data. a photo posted to twitter this month showed more patriot
front graffiti along the red mountain expressway in birmingham with the words, "we d
are defend our rights." the patriot front graffiti was later removed, but the messag

```
e left sydney duncan, the attorney director for the magic legal center in birmingha
m, saddened that hate had become so public in some parts of alabama. "white supremac
y is alive and well," duncan wrote.  hughes said she was traveling north to birmingh
am when she pulled over on i 65 to take photos of the messages on the sign. she had
seen confederate monuments and flags on that drive before, but that kind of messagin
g on government owned property was different, she said. a police officer who was alr
eady at the scene waved at her to keep driving, hughes added. when she returned hom
e, hughes said she felt compelled to share the images due to the ongoing conversatio
n happening among birmingham residents about the promotion of patriot front in publi
c spaces. "some people might perceive this as upsetting and scary, and a sign of the
worsening of our country," she said. "but if this is their strategy, then i'm not re
ally impressed." she added, "they're a dying breed." toluse olorunnipa and azi payba
rah contributed to this report.
```

Lemmatization using spaCY

In [28]:
```
nlp_trans = spacy.load('en_core_web_sm')

def lemma(text):
    trans_txt = nlp_trans(text)
    tokens = [t.lemma_ for t in trans_txt]
    return tokens
```

slct_tbl_full_df03['lemma'] = slct_tbl_full_df03['replace'].progress_apply(lemma)

print(slct_tbl_full_df03.shape) display(slct_tbl_full_df03.head())

for c in range(0,1): try: print(slct_tbl_full_df03['lemma'][c], '\n') except: print(f'Skip {c}')

Display globally unique tokens on lemmas

uniq_tok(df_col=slct_tbl_full_df03['lemma'])

Split text

Apply

In [29]:
```
slct_tbl_full_df03['split'] = slct_tbl_full_df03['emoji_split']\
.apply(str.split)

print(slct_tbl_full_df03.shape)
display(slct_tbl_full_df03.head())

for c in range(0,1):
    try:
        print(slct_tbl_full_df03['split'][c], '\n')
    except:
        print(f'Skip {c}')
```

```
(4026, 17)
```

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

['travelers', 'in', 'alabama', 'driving', 'on', 'interstate', '65', 'to', 'parties',
'and', 'barbecues', 'on', 'memorial', 'day', 'might', 'have', 'seen', 'messages', 'o
n', 'digital', 'road', 'signs', 'honoring', 'veterans', 'who', 'died', 'fighting',
'for', 'the', 'united', 'states.', 'but', "that's", 'not', 'what', 'some', 'driver
s', 'near', 'clanton,', 'ala.,', 'saw', 'on', 'monday.', 'instead,', 'motorists', 'r
eported', 'seeing', 'a', 'sign', 'that', 'was', 'apparently', 'hacked', 'to', 'displ
ay', 'the', 'words', '"reclaim', 'america,"', 'a', 'white', 'nationalist', 'sloga
n,', 'and', '"patriot', 'front', 'us,"', 'referencing', 'the', 'white', 'supremacis
t', 'group', 'that', 'was', 'involved', 'in', 'the', 'deadly', '2017', 'unite', 'th
e', 'right', 'rally', 'in', 'charlottesville.', '"how', 'does', 'this', 'come', 'abo
ut?"', 'wrote', 'sarah', 'hughes,', 'a', 'motorist', 'who', 'captured', 'photos', 'o
f', 'the', 'sign', 'and', 'posted', 'them', 'on', 'twitter.', '"weird', 'as', 'hel
l."', 'a', "contractor's", 'portable', 'message', 'board', 'was', 'hacked', 'on',
'i', '65', 'in', 'chilton', 'county,', 'ala.,', 'on', 'monday', 'afternoon,', 'joh
n', 'mcwilliams,', 'a', 'spokesman', 'for', 'the', 'alabama', 'department', 'of', 't
ransportation', '(aldot)', 'west', 'central', 'region,', 'told', 'the', 'washingto
n', 'post', 'in', 'a', 'statement.', '"a', 'citizen', 'alerted', 'a', 'nearby', 'sta
te', 'trooper', 'about', 'the', 'message,', 'who', 'then', 'contacted', 'aldot,"',
'mcwilliams', 'said', 'tuesday.', '"aldot', 'personnel', 'immediately', 'responded',
'and', 'turned', 'the', 'message', 'board', 'off.', 'no', 'other', 'message', 'board
s', 'on', 'i', '65', 'were', 'affected."', 'mcwilliams', 'added', 'that', 'aldot',
'is', 'investigating', 'how', 'the', 'white', 'supremacist', 'language', 'appeared',
'on', 'the', 'sign', 'near', 'clanton,', 'about', '40', 'miles', 'northwest', 'of',
'montgomery,', 'ala.', 'officials', 'have', 'given', 'no', 'immediate', 'indicatio
n', 'of', 'who', 'is', 'responsible', 'for', 'apparently', 'hacking', 'the', 'inters
tate', 'sign.', 'the', 'news', 'was', 'first', 'reported', 'by', 'al.com.', 'hughe
s', 'told', 'the', 'post', 'that', 'she', 'was', 'driving', 'home', 'to', 'birmingha
m', 'from', 'a', 'weekend', 'at', "alabama's", 'gulf', 'coast', 'when', 'she', 'sa
w', 'the', 'white', 'supremacist', 'messages', 'that', 'have', 'recently', 'popped',
'up', 'around', 'her', 'home', 'city', 'from', 'supporters', 'of', 'patriot', 'fron
t.', '"when', 'i', 'saw', 'it,', 'i', 'thought,', "'oh,", "it's", 'the', 'same', "gu
ys,'", '"', 'said', 'hughes,', 'a', '31', 'year', 'old', 'attorney.', '"i', 'was',
'kind', 'of', 'shocked."', 'the', 'hacked', 'alabama', 'road', 'sign', 'comes', 'a
t', 'a', 'time', 'when', 'president', 'biden', 'has', 'declared', 'white', 'supremac
y', '"the', 'most', 'dangerous', 'terrorist', 'threat"', 'to', 'the', 'country.', 'd
uring', 'his', 'commencement', 'address', 'at', 'howard', 'university', 'this', 'mon
th,', 'biden', 'told', 'the', 'graduating', 'class', 'at', 'the', 'historically', 'b
lack', 'university', 'that', 'he', 'pledged', '"to', 'stand', 'up', 'against', 'th
e', 'poison', 'of', 'white', 'supremacy,', 'as', 'i', 'did', 'in', 'my', 'inaugura
l', 'address', '—', 'to', 'single', 'it', 'out', 'as', 'the', 'most', 'dangerous',
'terrorist', 'threat', 'to', 'our', 'homeland', 'is', 'white', 'supremacy."', '"i',
"don't", 'have', 'to', 'tell', 'you', 'that', 'progress', 'toward', 'justice', 'ofte
n', 'meets', 'ferocious', 'pushback', 'from', 'the', 'oldest', 'and', 'most', 'sinis
ter', 'of', 'forces,"', 'biden', 'said', 'in', 'the', 'may', '13', 'address,', 'afte
r', 'quoting', 'donald', "trump's", 'equivocating', 'response', 'to', 'the', '2017',
'rally', 'in', 'charlottesville', 'that', 'killed', '32', 'year', 'old', 'heather',
'heyer', 'and', 'injured', '19', 'others.', '"that\'s', 'because', 'hate', 'never',
'goes', 'away."', 'biden', 'calls', 'white', 'supremacy', 'greatest', 'terrorism',
'threat', 'as', '2024', 'race', 'heats', 'up', 'the', 'southern', 'poverty', 'law',
'center', '(splc)', 'tracked', 'at', 'least', '13', 'hate', 'groups', 'in', 'alabam
a', 'in', '2021,', 'including', 'the', 'proud', 'boys.', 'the', 'discussion', 'surro
unding', 'white', 'supremacists', 'and', 'white', 'nationalists', 'in', 'alabama',
'intensified', 'this', 'month', 'after', 'sen.', 'tommy', 'tuberville', '(r', 'al
a.)', 'said', 'that', 'people', 'identified', 'as', '"white', 'extremists"', 'and',
'white', 'nationalists', 'should', 'be', 'allowed', 'to', 'serve', 'in', 'the', 'u.
s.', 'armed', 'forces.', 'when', 'asked', 'by', 'a', 'reporter', 'with', 'wbhm', 'i

```
n', 'birmingham', 'whether', 'white', 'nationalists', 'should', 'be', 'allowed', 't
o', 'serve', 'in', 'the', 'military,', 'tuberville', 'replied,', '"well,', 'they',
'call', 'them', 'that.', 'i', 'call', 'them', 'americans."', 'after', 'tuberville',
'was', 'criticized,', 'a', 'spokesman', 'told', 'the', 'post', 'that', 'the', 'senat
or', '"resents', 'the', 'implication', 'that', 'the', 'people', 'in', 'our', 'milita
ry', 'are', 'anything', 'but', 'patriots', 'and', 'heroes."', 'gop', 'senator', 'say
s', 'of', 'white', 'nationalists', 'in', 'the', 'military,', "'i", 'call', 'them',
"americans'", 'patriot', 'front,', 'the', 'white', 'supremacist', 'group', 'whose',
'name', 'was', 'displayed', 'on', 'the', 'interstate', 'sign,', 'is', 'a', 'texas',
'based', 'hate', 'group', 'that', 'broke', 'off', 'from', 'vanguard', 'america', 'an
d', 'formed', 'after', 'the', 'charlottesville', 'rally,', 'the', 'splc', 'says.',
'its', 'members', 'have', 'chanted', '"reclaim', 'america"', 'at', 'rallies', 'in',
'coeur', "d'alene,", 'idaho,', 'washington', 'and', 'boston', 'in', 'recent', 'year
s,', 'according', 'to', 'news', 'reports.', 'patriot', 'front', 'is', 'responsible',
'for', '"the', 'vast', 'majority', 'of', 'white', 'supremacist', 'propaganda', 'dist
ributed', 'in', 'the', 'united', 'states"', 'since', '2019,', 'according', 'to', 'th
e', 'anti', 'defamation', 'league.', "it's", 'not', 'the', 'first', 'time', 'that',
'language', 'promoting', 'patriot', 'front', 'has', 'made', 'its', 'way', 'into',
'a', 'public', 'space', 'in', 'alabama.', 'in', 'july,', 'graffiti', 'beneath', 'a',
'birmingham', 'bridge', 'appeared', 'with', '"patriot', 'front', 'us"', 'spray', 'pa
inted', 'in', 'red', 'and', 'blue', 'letters,', 'al.com', 'reported.', 'other', 'pat
riot', 'front', 'graffiti', 'has', 'also', 'been', 'spotted', 'in', 'birmingham,',
'a', 'city', 'with', 'a', 'population', "that's", 'nearly', '70', 'percent', 'blac
k,', 'according', 'to', 'u.s.', 'census', 'data.', 'a', 'photo', 'posted', 'to', 'tw
itter', 'this', 'month', 'showed', 'more', 'patriot', 'front', 'graffiti', 'along',
'the', 'red', 'mountain', 'expressway', 'in', 'birmingham', 'with', 'the', 'words,',
'"we', 'dare', 'defend', 'our', 'rights."', 'the', 'patriot', 'front', 'graffiti',
'was', 'later', 'removed,', 'but', 'the', 'message', 'left', 'sydney', 'duncan,', 't
he', 'attorney', 'director', 'for', 'the', 'magic', 'legal', 'center', 'in', 'birmin
gham,', 'saddened', 'that', 'hate', 'had', 'become', 'so', 'public', 'in', 'some',
'parts', 'of', 'alabama.', '"white', 'supremacy', 'is', 'alive', 'and', 'well,"', 'd
uncan', 'wrote.', 'hughes', 'said', 'she', 'was', 'traveling', 'north', 'to', 'birmi
ngham', 'when', 'she', 'pulled', 'over', 'on', 'i', '65', 'to', 'take', 'photos', 'o
f', 'the', 'messages', 'on', 'the', 'sign.', 'she', 'had', 'seen', 'confederate', 'm
onuments', 'and', 'flags', 'on', 'that', 'drive', 'before,', 'but', 'that', 'kind',
'of', 'messaging', 'on', 'government', 'owned', 'property', 'was', 'different,', 'sh
e', 'said.', 'a', 'police', 'officer', 'who', 'was', 'already', 'at', 'the', 'scen
e', 'waved', 'at', 'her', 'to', 'keep', 'driving,', 'hughes', 'added.', 'when', 'sh
e', 'returned', 'home,', 'hughes', 'said', 'she', 'felt', 'compelled', 'to', 'shar
e', 'the', 'images', 'due', 'to', 'the', 'ongoing', 'conversation', 'happening', 'am
ong', 'birmingham', 'residents', 'about', 'the', 'promotion', 'of', 'patriot', 'fron
t', 'in', 'public', 'spaces.', '"some', 'people', 'might', 'perceive', 'this', 'as',
'upsetting', 'and', 'scary,', 'and', 'a', 'sign', 'of', 'the', 'worsening', 'of', 'o
ur', 'country,"', 'she', 'said.', '"but', 'if', 'this', 'is', 'their', 'strategy,',
'then', "i'm", 'not', 'really', 'impressed."', 'she', 'added,', '"they\'re', 'a', 'd
ying', 'breed."', 'toluse', 'olorunnipa', 'and', 'azi', 'paybarah', 'contributed',
'to', 'this', 'report.']
```

Display globally unique tokens on first split

In [30]:
```
#uniq_tok(df_col=slct_tbl_full_df03['split'])
```

Remove stop words

In [31]:

```
sw = stopwords.words("english")

# Add additional stop words
sw.extend(['',
           '',
           'arent',
           'cannot',
           'cant',
           'couldnt',
           'couldve',
           'didnt',
           'doesnt',
           'dont',
           'hadnt',
           'hasnt',
           'havent',
           'hes',
           'im',
           "i'm",
           'isnt',
           'it’s',
           'ive',
           'of',
           'mightnt',
           'mustnt',
           'neednt',
           'shant',
           'shes',
           'shouldnt',
           'shouldve',
           'thatll',
           'theyll',
           'theyve',
           'wasnt',
           'werent',
           'whats',
           'weve',
           'wont',
           'wouldnt',
           'wouldve',
           'yall',
           'youd',
           'youll',
           'youre',
           'youve',
           "we'll",
           "you’re",
           "you’ve",
           "you’ll",
           "you’d",
           "she’s",
           "it’s",
           "that’ll",
           "don’t",
           "should’ve",
```

```
            "aren't",
            "couldn't",
            "didn't",
            "doesn't",
            "hadn't",
            "hasn't",
            "haven't",
            "isn't",
            "mightn't",
            "mustn't",
            "needn't",
            "shan't",
            "shouldn't",
            "wasn't",
            "weren't",
            "won't",
            "wouldn't",
            "i'm",
            "we'll",
            'said',
            'told',
            'according',
            'fox',
            'news',
            'cnn',
            'breitbart',
            'reuters'])

print(sw)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "yo
u've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'h
is', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itse
lf', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'who
m', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were',
'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing',
'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of',
'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'durin
g', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'o
n', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'wh
en', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'ot
her', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'to
o', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've",
'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "cou
ldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'h
aven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'n
eedn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'were
n', "weren't", 'won', "won't", 'wouldn', "wouldn't", '', '', 'arent', 'cannot', 'can
t', 'couldnt', 'couldve', 'didnt', 'doesnt', 'dont', 'hadnt', 'hasnt', 'havent', 'he
s', 'im', "i'm", 'isnt', 'it’s', 'ive', 'of', 'mightnt', 'mustnt', 'neednt', 'shan
t', 'shes', 'shouldnt', 'shouldve', 'thatll', 'theyll', 'theyve', 'wasnt', 'werent',
'whats', 'weve', 'wont', 'wouldnt', 'wouldve', 'yall', 'youd', 'youll', 'youre', 'yo
uve', "we'll", 'you’re', 'you’ve', 'you’ll', 'you’d', 'she’s', 'it’s', 'that’ll', 'd
on’t', 'should’ve', 'aren’t', 'couldn’t', 'didn’t', 'doesn’t', 'hadn’t', 'hasn’t',
'haven’t', 'isn’t', 'mightn’t', 'mustn’t', 'needn’t', 'shan’t', 'shouldn’t', 'was
n’t', 'weren’t', 'won’t', 'wouldn’t', 'i’m', 'we’ll', 'said', 'told', 'according',
'fox', 'news', 'cnn', 'breitbart', 'reuters']
```

Create function

In [32]:
```
def sw_remover(tokens):
    return [t for t in tokens if t.lower() not in sw]
```

Call function

In [33]:
```
slct_tbl_full_df03['no_sw'] = slct_tbl_full_df03['split'].apply(sw_remover)

print(slct_tbl_full_df03.shape)
display(slct_tbl_full_df03.head())

for c in range(0,1):
    print(slct_tbl_full_df03['no_sw'][c])
```

```
(4026, 18)
```

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

['travelers', 'alabama', 'driving', 'interstate', '65', 'parties', 'barbecues', 'mem
orial', 'day', 'might', 'seen', 'messages', 'digital', 'road', 'signs', 'honoring',
'veterans', 'died', 'fighting', 'united', 'states.', "that's", 'drivers', 'near', 'c
lanton,', 'ala.,', 'saw', 'monday.', 'instead,', 'motorists', 'reported', 'seeing',
'sign', 'apparently', 'hacked', 'display', 'words', '"reclaim', 'america,"', 'whit
e', 'nationalist', 'slogan,', '"patriot', 'front', 'us,"', 'referencing', 'white',
'supremacist', 'group', 'involved', 'deadly', '2017', 'unite', 'right', 'rally', 'ch
arlottesville.', '"how', 'come', 'about?"', 'wrote', 'sarah', 'hughes,', 'motorist',
'captured', 'photos', 'sign', 'posted', 'twitter.', '"weird', 'hell."', "contracto
r's", 'portable', 'message', 'board', 'hacked', '65', 'chilton', 'county,', 'ala.,',
'monday', 'afternoon,', 'john', 'mcwilliams,', 'spokesman', 'alabama', 'department',
'transportation', '(aldot)', 'west', 'central', 'region,', 'washington', 'post', 'st
atement.', '"a', 'citizen', 'alerted', 'nearby', 'state', 'trooper', 'message,', 'co
ntacted', 'aldot,"', 'mcwilliams', 'tuesday.', '"aldot', 'personnel', 'immediately',
'responded', 'turned', 'message', 'board', 'off.', 'message', 'boards', '65', 'affec
ted."', 'mcwilliams', 'added', 'aldot', 'investigating', 'white', 'supremacist', 'la
nguage', 'appeared', 'sign', 'near', 'clanton,', '40', 'miles', 'northwest', 'montgo
mery,', 'ala.', 'officials', 'given', 'immediate', 'indication', 'responsible', 'app
arently', 'hacking', 'interstate', 'sign.', 'first', 'reported', 'al.com.', 'hughe
s', 'post', 'driving', 'home', 'birmingham', 'weekend', "alabama's", 'gulf', 'coas
t', 'saw', 'white', 'supremacist', 'messages', 'recently', 'popped', 'around', 'hom
e', 'city', 'supporters', 'patriot', 'front.', '"when', 'saw', 'it,', 'thought,',
"'oh,", "guys,'", '"', 'hughes,', '31', 'year', 'old', 'attorney.', '"i', 'kind', 's
hocked."', 'hacked', 'alabama', 'road', 'sign', 'comes', 'time', 'president', 'bide
n', 'declared', 'white', 'supremacy', '"the', 'dangerous', 'terrorist', 'threat"',
'country.', 'commencement', 'address', 'howard', 'university', 'month,', 'biden', 'g
raduating', 'class', 'historically', 'black', 'university', 'pledged', '"to', 'stan
d', 'poison', 'white', 'supremacy,', 'inaugural', 'address', '—', 'single', 'dangero
us', 'terrorist', 'threat', 'homeland', 'white', 'supremacy."', '"i', 'tell', 'progr
ess', 'toward', 'justice', 'often', 'meets', 'ferocious', 'pushback', 'oldest', 'sin
ister', 'forces,"', 'biden', 'may', '13', 'address,', 'quoting', 'donald', "trum
p's", 'equivocating', 'response', '2017', 'rally', 'charlottesville', 'killed', '3
2', 'year', 'old', 'heather', 'heyer', 'injured', '19', 'others.', '"that\'s', 'hat
e', 'never', 'goes', 'away."', 'biden', 'calls', 'white', 'supremacy', 'greatest',
'terrorism', 'threat', '2024', 'race', 'heats', 'southern', 'poverty', 'law', 'cente
r', '(splc)', 'tracked', 'least', '13', 'hate', 'groups', 'alabama', '2021,', 'inclu
ding', 'proud', 'boys.', 'discussion', 'surrounding', 'white', 'supremacists', 'whit
e', 'nationalists', 'alabama', 'intensified', 'month', 'sen.', 'tommy', 'tubervill
e', '(r', 'ala.)', 'people', 'identified', '"white', 'extremists"', 'white', 'nation
alists', 'allowed', 'serve', 'u.s.', 'armed', 'forces.', 'asked', 'reporter', 'wbh
m', 'birmingham', 'whether', 'white', 'nationalists', 'allowed', 'serve', 'militar
y,', 'tuberville', 'replied,', '"well,', 'call', 'that.', 'call', 'americans."', 'tu
berville', 'criticized,', 'spokesman', 'post', 'senator', '"resents', 'implication',
'people', 'military', 'anything', 'patriots', 'heroes."', 'gop', 'senator', 'says',
'white', 'nationalists', 'military,', "'i", 'call', "americans'", 'patriot', 'fron
t,', 'white', 'supremacist', 'group', 'whose', 'name', 'displayed', 'interstate', 's
ign,', 'texas', 'based', 'hate', 'group', 'broke', 'vanguard', 'america', 'formed',
'charlottesville', 'rally,', 'splc', 'says.', 'members', 'chanted', '"reclaim', 'ame
rica"', 'rallies', 'coeur', "d'alene,", 'idaho,', 'washington', 'boston', 'recent',
'years,', 'reports.', 'patriot', 'front', 'responsible', '"the', 'vast', 'majority',
'white', 'supremacist', 'propaganda', 'distributed', 'united', 'states"', 'since',
'2019,', 'anti', 'defamation', 'league.', 'first', 'time', 'language', 'promoting',
'patriot', 'front', 'made', 'way', 'public', 'space', 'alabama.', 'july,', 'graffit
i', 'beneath', 'birmingham', 'bridge', 'appeared', '"patriot', 'front', 'us"', 'spra
y', 'painted', 'red', 'blue', 'letters,', 'al.com', 'reported.', 'patriot', 'front',
'graffiti', 'also', 'spotted', 'birmingham,', 'city', 'population', "that's", 'nearl

```
y', '70', 'percent', 'black,', 'u.s.', 'census', 'data.', 'photo', 'posted', 'twitte
r', 'month', 'showed', 'patriot', 'front', 'graffiti', 'along', 'red', 'mountain',
'expressway', 'birmingham', 'words,', '"we', 'dare', 'defend', 'rights."', 'patrio
t', 'front', 'graffiti', 'later', 'removed,', 'message', 'left', 'sydney', 'dunca
n,', 'attorney', 'director', 'magic', 'legal', 'center', 'birmingham,', 'saddened',
'hate', 'become', 'public', 'parts', 'alabama.', '"white', 'supremacy', 'alive', 'we
ll,"', 'duncan', 'wrote.', 'hughes', 'traveling', 'north', 'birmingham', 'pulled',
'65', 'take', 'photos', 'messages', 'sign.', 'seen', 'confederate', 'monuments', 'fl
ags', 'drive', 'before,', 'kind', 'messaging', 'government', 'owned', 'property', 'd
ifferent,', 'said.', 'police', 'officer', 'already', 'scene', 'waved', 'keep', 'driv
ing,', 'hughes', 'added.', 'returned', 'home,', 'hughes', 'felt', 'compelled', 'shar
e', 'images', 'due', 'ongoing', 'conversation', 'happening', 'among', 'birmingham',
'residents', 'promotion', 'patriot', 'front', 'public', 'spaces.', '"some', 'peopl
e', 'might', 'perceive', 'upsetting', 'scary,', 'sign', 'worsening', 'country,"', 's
aid.', '"but', 'strategy,', 'really', 'impressed."', 'added,', '"they\'re', 'dying',
```

Display no stop words

In [34]:
```
#uniq_tok(df_col=slct_tbl_full_df03['no_sw'])
```

Rejoin semi_processed tokens

In [35]:
```
slct_tbl_full_df03['no_sw_join'] = slct_tbl_full_df03['no_sw'].apply(" ".join)

print(slct_tbl_full_df03.shape)
display(slct_tbl_full_df03.head())

for c in range(0,1):
    print(slct_tbl_full_df03['no_sw_join'][c])
```

```
(4026, 19)
```

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

travelers alabama driving interstate 65 parties barbecues memorial day might seen me ssages digital road signs honoring veterans died fighting united states. that's driv ers near clanton, ala., saw monday. instead, motorists reported seeing sign apparent ly hacked display words "reclaim america," white nationalist slogan, "patriot front us," referencing white supremacist group involved deadly 2017 unite right rally char lottesville. "how come about?" wrote sarah hughes, motorist captured photos sign pos ted twitter. "weird hell." contractor's portable message board hacked 65 chilton cou nty, ala., monday afternoon, john mcwilliams, spokesman alabama department transport ation (aldot) west central region, washington post statement. "a citizen alerted nea rby state trooper message, contacted aldot," mcwilliams tuesday. "aldot personnel im mediately responded turned message board off. message boards 65 affected." mcwilliam s added aldot investigating white supremacist language appeared sign near clanton, 4 0 miles northwest montgomery, ala. officials given immediate indication responsible apparently hacking interstate sign. first reported al.com. hughes post driving home birmingham weekend alabama's gulf coast saw white supremacist messages recently popp ed around home city supporters patriot front. "when saw it, thought, 'oh, guys,' " h ughes, 31 year old attorney. "i kind shocked." hacked alabama road sign comes time p resident biden declared white supremacy "the dangerous terrorist threat" country. co mmencement address howard university month, biden graduating class historically blac k university pledged "to stand poison white supremacy, inaugural address — single da ngerous terrorist threat homeland white supremacy." "i tell progress toward justice often meets ferocious pushback oldest sinister forces," biden may 13 address, quotin g donald trump's equivocating response 2017 rally charlottesville killed 32 year old heather heyer injured 19 others. "that's hate never goes away." biden calls white su premacy greatest terrorism threat 2024 race heats southern poverty law center (splc) tracked least 13 hate groups alabama 2021, including proud boys. discussion surround ing white supremacists white nationalists alabama intensified month sen. tommy tuber ville (r ala.) people identified "white extremists" white nationalists allowed serve u.s. armed forces. asked reporter wbhm birmingham whether white nationalists allowed serve military, tuberville replied, "well, call that. call americans." tuberville cr iticized, spokesman post senator "resents implication people military anything patri ots heroes." gop senator says white nationalists military, 'i call americans' patrio t front, white supremacist group whose name displayed interstate sign, texas based h ate group broke vanguard america formed charlottesville rally, splc says. members ch anted "reclaim america" rallies coeur d'alene, idaho, washington boston recent year s, reports. patriot front responsible "the vast majority white supremacist propagand a distributed united states" since 2019, anti defamation league. first time language promoting patriot front made way public space alabama. july, graffiti beneath birmin gham bridge appeared "patriot front us" spray painted red blue letters, al.com repor ted. patriot front graffiti also spotted birmingham, city population that's nearly 7 0 percent black, u.s. census data. photo posted twitter month showed patriot front g raffiti along red mountain expressway birmingham words, "we dare defend rights." pat riot front graffiti later removed, message left sydney duncan, attorney director mag ic legal center birmingham, saddened hate become public parts alabama. "white suprem acy alive well," duncan wrote. hughes traveling north birmingham pulled 65 take phot os messages sign. seen confederate monuments flags drive before, kind messaging gove rnment owned property different, said. police officer already scene waved keep drivi ng, hughes added. returned home, hughes felt compelled share images due ongoing conv ersation happening among birmingham residents promotion patriot front public spaces. "some people might perceive upsetting scary, sign worsening country," said. "but str ategy, really impressed." added, "they're dying breed." toluse olorunnipa azi paybar

Remove punctuation

In [36]:
```
punctuation = set(punctuation) # speeds up comparison
#print(punctuation)

# Add special hyphen mark
tw_punct = punctuation - {"#"}
#print(tw_punct)

# Remove hash and at symbols for later capture of hashtag info
tw_punct = tw_punct - {"@"}
tw_punct = tw_punct - {"-"}
#tw_punct = tw_punct - {"/"}
tw_punct.add("’")
tw_punct.add("‘")
tw_punct.add("”")
tw_punct.add("“")
tw_punct.add("…")
tw_punct.add("—")
tw_punct.add("...")
tw_punct.add("€")
tw_punct.add("±")
tw_punct.add("£")
tw_punct.add("¡")
tw_punct.add("§")
tw_punct.add("◉")

print(tw_punct)
```

```
{'[', ',', '§', '^', '◉', '/', '?', '`', '...', ';', ')', '>', '‘', '€', ']', '…',
'.', '+', '¡', '*', "'", '(', '%', '”', '!', '’', '<', '=', '{', '\\', '|', '}',
'±', '$', '£', '"', ':', '_', '“', '—', '~', '&'}
```

Create function

In [37]:
```
def remove_punctuation(text, punct_set=tw_punct):
    return("".join([ch for ch in text if ch not in punct_set]))
```

Call function

In [38]:
```
slct_tbl_full_df03['no_sw_join_no_punc'] = slct_tbl_full_df03['no_sw_join']\
.apply(remove_punctuation, punct_set=tw_punct)

print(slct_tbl_full_df03.shape)
display(slct_tbl_full_df03.head())

for c in range(0,1):
    try:
        print(slct_tbl_full_df03['no_sw_join_no_punc'][c], '\n')
    except:
        print(f'\nerror on {c}\n')
```

```
(4026, 20)
```

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

travelers alabama driving interstate 65 parties barbecues memorial day might seen me ssages digital road signs honoring veterans died fighting united states thats driver s near clanton ala saw monday instead motorists reported seeing sign apparently hack ed display words reclaim america white nationalist slogan patriot front us referenci ng white supremacist group involved deadly 2017 unite right rally charlottesville ho w come about wrote sarah hughes motorist captured photos sign posted twitter weird h ell contractors portable message board hacked 65 chilton county ala monday afternoon john mcwilliams spokesman alabama department transportation aldot west central regio n washington post statement a citizen alerted nearby state trooper message contacted aldot mcwilliams tuesday aldot personnel immediately responded turned message board off message boards 65 affected mcwilliams added aldot investigating white supremacis t language appeared sign near clanton 40 miles northwest montgomery ala officials gi ven immediate indication responsible apparently hacking interstate sign first report ed alcom hughes post driving home birmingham weekend alabamas gulf coast saw white s upremacist messages recently popped around home city supporters patriot front when s aw it thought oh guys hughes 31 year old attorney i kind shocked hacked alabama roa d sign comes time president biden declared white supremacy the dangerous terrorist t hreat country commencement address howard university month biden graduating class hi storically black university pledged to stand poison white supremacy inaugural addres s single dangerous terrorist threat homeland white supremacy i tell progress toward justice often meets ferocious pushback oldest sinister forces biden may 13 address q uoting donald trumps equivocating response 2017 rally charlottesville killed 32 year old heather heyer injured 19 others thats hate never goes away biden calls white sup remacy greatest terrorism threat 2024 race heats southern poverty law center splc tr acked least 13 hate groups alabama 2021 including proud boys discussion surrounding white supremacists white nationalists alabama intensified month sen tommy tuberville r ala people identified white extremists white nationalists allowed serve us armed f orces asked reporter wbhm birmingham whether white nationalists allowed serve milita ry tuberville replied well call that call americans tuberville criticized spokesman post senator resents implication people military anything patriots heroes gop senato r says white nationalists military i call americans patriot front white supremacist group whose name displayed interstate sign texas based hate group broke vanguard ame rica formed charlottesville rally splc says members chanted reclaim america rallies coeur dalene idaho washington boston recent years reports patriot front responsible the vast majority white supremacist propaganda distributed united states since 2019 anti defamation league first time language promoting patriot front made way public s pace alabama july graffiti beneath birmingham bridge appeared patriot front us spray painted red blue letters alcom reported patriot front graffiti also spotted birmingh am city population thats nearly 70 percent black us census data photo posted twitter month showed patriot front graffiti along red mountain expressway birmingham words w e dare defend rights patriot front graffiti later removed message left sydney duncan attorney director magic legal center birmingham saddened hate become public parts al abama white supremacy alive well duncan wrote hughes traveling north birmingham pull ed 65 take photos messages sign seen confederate monuments flags drive before kind m essaging government owned property different said police officer already scene waved keep driving hughes added returned home hughes felt compelled share images due ongoi ng conversation happening among birmingham residents promotion patriot front public spaces some people might perceive upsetting scary sign worsening country said but st rategy really impressed added theyre dying breed toluse olorunnipa azi paybarah cont ributed report

Tokenize

In [39]:
```
slct_tbl_full_df03['no_sw_join_no_punc_tok'] \
= slct_tbl_full_df03['no_sw_join_no_punc'].apply(str.split)

print(slct_tbl_full_df03.shape)
display(slct_tbl_full_df03.head())

for c in range(0,1):
    print(slct_tbl_full_df03['no_sw_join_no_punc_tok'][c], '\n')
```

```
(4026, 21)
```

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

5 rows × 21 columns

['travelers', 'alabama', 'driving', 'interstate', '65', 'parties', 'barbecues', 'mem
orial', 'day', 'might', 'seen', 'messages', 'digital', 'road', 'signs', 'honoring',
'veterans', 'died', 'fighting', 'united', 'states', 'thats', 'drivers', 'near', 'cla
nton', 'ala', 'saw', 'monday', 'instead', 'motorists', 'reported', 'seeing', 'sign',
'apparently', 'hacked', 'display', 'words', 'reclaim', 'america', 'white', 'national
ist', 'slogan', 'patriot', 'front', 'us', 'referencing', 'white', 'supremacist', 'gr
oup', 'involved', 'deadly', '2017', 'unite', 'right', 'rally', 'charlottesville', 'h
ow', 'come', 'about', 'wrote', 'sarah', 'hughes', 'motorist', 'captured', 'photos',
'sign', 'posted', 'twitter', 'weird', 'hell', 'contractors', 'portable', 'message',
'board', 'hacked', '65', 'chilton', 'county', 'ala', 'monday', 'afternoon', 'john',
'mcwilliams', 'spokesman', 'alabama', 'department', 'transportation', 'aldot', 'wes
t', 'central', 'region', 'washington', 'post', 'statement', 'a', 'citizen', 'alerte
d', 'nearby', 'state', 'trooper', 'message', 'contacted', 'aldot', 'mcwilliams', 'tu
esday', 'aldot', 'personnel', 'immediately', 'responded', 'turned', 'message', 'boar
d', 'off', 'message', 'boards', '65', 'affected', 'mcwilliams', 'added', 'aldot', 'i
nvestigating', 'white', 'supremacist', 'language', 'appeared', 'sign', 'near', 'clan
ton', '40', 'miles', 'northwest', 'montgomery', 'ala', 'officials', 'given', 'immedi
ate', 'indication', 'responsible', 'apparently', 'hacking', 'interstate', 'sign', 'f
irst', 'reported', 'alcom', 'hughes', 'post', 'driving', 'home', 'birmingham', 'week
end', 'alabamas', 'gulf', 'coast', 'saw', 'white', 'supremacist', 'messages', 'recen
tly', 'popped', 'around', 'home', 'city', 'supporters', 'patriot', 'front', 'when',
'saw', 'it', 'thought', 'oh', 'guys', 'hughes', '31', 'year', 'old', 'attorney',
'i', 'kind', 'shocked', 'hacked', 'alabama', 'road', 'sign', 'comes', 'time', 'presi
dent', 'biden', 'declared', 'white', 'supremacy', 'the', 'dangerous', 'terrorist',
'threat', 'country', 'commencement', 'address', 'howard', 'university', 'month', 'bi
den', 'graduating', 'class', 'historically', 'black', 'university', 'pledged', 'to',
'stand', 'poison', 'white', 'supremacy', 'inaugural', 'address', 'single', 'dangerou
s', 'terrorist', 'threat', 'homeland', 'white', 'supremacy', 'i', 'tell', 'progres
s', 'toward', 'justice', 'often', 'meets', 'ferocious', 'pushback', 'oldest', 'sinis
ter', 'forces', 'biden', 'may', '13', 'address', 'quoting', 'donald', 'trumps', 'equ
ivocating', 'response', '2017', 'rally', 'charlottesville', 'killed', '32', 'year',
'old', 'heather', 'heyer', 'injured', '19', 'others', 'thats', 'hate', 'never', 'goe
s', 'away', 'biden', 'calls', 'white', 'supremacy', 'greatest', 'terrorism', 'threa
t', '2024', 'race', 'heats', 'southern', 'poverty', 'law', 'center', 'splc', 'tracke
d', 'least', '13', 'hate', 'groups', 'alabama', '2021', 'including', 'proud', 'boy
s', 'discussion', 'surrounding', 'white', 'supremacists', 'white', 'nationalists',
'alabama', 'intensified', 'month', 'sen', 'tommy', 'tuberville', 'r', 'ala', 'peopl
e', 'identified', 'white', 'extremists', 'white', 'nationalists', 'allowed', 'serv
e', 'us', 'armed', 'forces', 'asked', 'reporter', 'wbhm', 'birmingham', 'whether',
'white', 'nationalists', 'allowed', 'serve', 'military', 'tuberville', 'replied', 'w
ell', 'call', 'that', 'call', 'americans', 'tuberville', 'criticized', 'spokesman',
'post', 'senator', 'resents', 'implication', 'people', 'military', 'anything', 'patr
iots', 'heroes', 'gop', 'senator', 'says', 'white', 'nationalists', 'military', 'i',
'call', 'americans', 'patriot', 'front', 'white', 'supremacist', 'group', 'whose',
'name', 'displayed', 'interstate', 'sign', 'texas', 'based', 'hate', 'group', 'brok
e', 'vanguard', 'america', 'formed', 'charlottesville', 'rally', 'splc', 'says', 'me
mbers', 'chanted', 'reclaim', 'america', 'rallies', 'coeur', 'dalene', 'idaho', 'was
hington', 'boston', 'recent', 'years', 'reports', 'patriot', 'front', 'responsible',
'the', 'vast', 'majority', 'white', 'supremacist', 'propaganda', 'distributed', 'uni
ted', 'states', 'since', '2019', 'anti', 'defamation', 'league', 'first', 'time', 'l
anguage', 'promoting', 'patriot', 'front', 'made', 'way', 'public', 'space', 'alabam
a', 'july', 'graffiti', 'beneath', 'birmingham', 'bridge', 'appeared', 'patriot', 'f
ront', 'us', 'spray', 'painted', 'red', 'blue', 'letters', 'alcom', 'reported', 'pat
riot', 'front', 'graffiti', 'also', 'spotted', 'birmingham', 'city', 'population',
'thats', 'nearly', '70', 'percent', 'black', 'us', 'census', 'data', 'photo', 'poste
d', 'twitter', 'month', 'showed', 'patriot', 'front', 'graffiti', 'along', 'red', 'm

```
ountain', 'expressway', 'birmingham', 'words', 'we', 'dare', 'defend', 'rights', 'pa
triot', 'front', 'graffiti', 'later', 'removed', 'message', 'left', 'sydney', 'dunca
n', 'attorney', 'director', 'magic', 'legal', 'center', 'birmingham', 'saddened', 'h
ate', 'become', 'public', 'parts', 'alabama', 'white', 'supremacy', 'alive', 'well',
'duncan', 'wrote', 'hughes', 'traveling', 'north', 'birmingham', 'pulled', '65', 'ta
ke', 'photos', 'messages', 'sign', 'seen', 'confederate', 'monuments', 'flags', 'dri
ve', 'before', 'kind', 'messaging', 'government', 'owned', 'property', 'different',
'said', 'police', 'officer', 'already', 'scene', 'waved', 'keep', 'driving', 'hughe
s', 'added', 'returned', 'home', 'hughes', 'felt', 'compelled', 'share', 'images',
'due', 'ongoing', 'conversation', 'happening', 'among', 'birmingham', 'residents',
'promotion', 'patriot', 'front', 'public', 'spaces', 'some', 'people', 'might', 'per
ceive', 'upsetting', 'scary', 'sign', 'worsening', 'country', 'said', 'but', 'strate
gy', 'really', 'impressed', 'added', 'theyre', 'dying', 'breed', 'toluse', 'olorunni
pa', 'azi', 'paybarah', 'contributed', 'report']
```

Display globally unique tokens on final tokens

In [40]:
```
#uniq_tok(df_col=slct_tbl_full_df03['no_sw_join_no_punc_tok'])
```

Pipeline consolidation

Pipeline function

In [41]:
```
def prepare(text, pipeline):
    '''Run a pipeline of text processing transformers'''
    tokens = str(text)

    # Pull key and val from trans dictionaries
    for transformer in pipeline:
        trans = list(transformer.keys())[0]
        args = list(transformer.values())[0]
        #print(trans)
        #print(args)
        if args == None:
            #print(1)
            tokens = trans(tokens)
        else:
            #print('check99', trans, args)
            tokens = trans(tokens, args)

    return(tokens)
```

article_text preprocessing _ w/o lemmatization

In [42]:
```
'''Set transformer pipeline 1:
Caseloading, normalization (using textacy), special ch removal,
split on whitespace, stop word removal, rejoin,
remove custom punctuation, tokenize
'''
transformers01 = [{str.lower: None},
                  {normalize: None},
```

```
                    {rex_replace: None},
                    {rex_url: None},
                    {emoji_split: None},
                    {str.split: None},
                    {sw_remover: None},
                    {" ".join: None},
                    {remove_punctuation: tw_punct},
                    {str.split: None},
                    {" ".join: None},
                   ]

# Apply transformers to pandas dataframe, w/ new col containing tokens
slct_tbl_full_df04['processed_text'] = slct_tbl_full_df04['article_text']\
.progress_apply(prepare, pipeline=transformers01)

slct_tbl_full_df04['processed_text_split'] = slct_tbl_full_df04['processed_text']\
.progress_apply(str.split)

slct_tbl_full_df04['num_tokens'] = slct_tbl_full_df04['processed_text_split']\
.map(len)

display(slct_tbl_full_df04.head())

# Review unique tokens across entire dataset
for c in range(0,1):
    try:
        print(slct_tbl_full_df04['processed_text'][c], '\n')
    except:
        print(f'Skip {c}')
```

```
100%|██████████| 4026/4026 [00:18<00:00, 219.33it/s]
100%|██████| 4026/4026 [00:00<00:00, 28206.68it/s]
```

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

travelers alabama driving interstate 65 parties barbecues memorial day might seen me ssages digital road signs honoring veterans died fighting united states thats driver s near clanton ala saw monday instead motorists reported seeing sign apparently hack ed display words reclaim america white nationalist slogan patriot front us referenci ng white supremacist group involved deadly 2017 unite right rally charlottesville ho w come about wrote sarah hughes motorist captured photos sign posted twitter weird h ell contractors portable message board hacked 65 chilton county ala monday afternoon john mcwilliams spokesman alabama department transportation aldot west central regio n washington post statement a citizen alerted nearby state trooper message contacted aldot mcwilliams tuesday aldot personnel immediately responded turned message board off message boards 65 affected mcwilliams added aldot investigating white supremacis t language appeared sign near clanton 40 miles northwest montgomery ala officials gi ven immediate indication responsible apparently hacking interstate sign first report ed alcom hughes post driving home birmingham weekend alabamas gulf coast saw white s upremacist messages recently popped around home city supporters patriot front when s aw it thought oh guys hughes 31 year old attorney i kind shocked hacked alabama road sign comes time president biden declared white supremacy the dangerous terrorist thr eat country commencement address howard university month biden graduating class hist orically black university pledged to stand poison white supremacy inaugural address single dangerous terrorist threat homeland white supremacy i tell progress toward ju stice often meets ferocious pushback oldest sinister forces biden may 13 address quo ting donald trumps equivocating response 2017 rally charlottesville killed 32 year o ld heather heyer injured 19 others thats hate never goes away biden calls white supr emacy greatest terrorism threat 2024 race heats southern poverty law center splc tra cked least 13 hate groups alabama 2021 including proud boys discussion surrounding w hite supremacists white nationalists alabama intensified month sen tommy tuberville r ala people identified white extremists white nationalists allowed serve us armed f orces asked reporter wbhm birmingham whether white nationalists allowed serve milita ry tuberville replied well call that call americans tuberville criticized spokesman post senator resents implication people military anything patriots heroes gop senato r says white nationalists military i call americans patriot front white supremacist group whose name displayed interstate sign texas based hate group broke vanguard ame rica formed charlottesville rally splc says members chanted reclaim america rallies coeur dalene idaho washington boston recent years reports patriot front responsible the vast majority white supremacist propaganda distributed united states since 2019 anti defamation league first time language promoting patriot front made way public s pace alabama july graffiti beneath birmingham bridge appeared patriot front us spray painted red blue letters alcom reported patriot front graffiti also spotted birmingh am city population thats nearly 70 percent black us census data photo posted twitter month showed patriot front graffiti along red mountain expressway birmingham words w e dare defend rights patriot front graffiti later removed message left sydney duncan attorney director magic legal center birmingham saddened hate become public parts al abama white supremacy alive well duncan wrote hughes traveling north birmingham pull ed 65 take photos messages sign seen confederate monuments flags drive before kind m essaging government owned property different said police officer already scene waved keep driving hughes added returned home hughes felt compelled share images due ongoi ng conversation happening among birmingham residents promotion patriot front public spaces some people might perceive upsetting scary sign worsening country said but st rategy really impressed added theyre dying breed toluse olorunnipa azi paybarah cont ributed report

Display globally unqiue tokens on final tokens

In [43]:
```
#uniq_tok(df_col=slct_tbl_full_df04['processed_text_split'])
```

article_text preprocessing w/ lemmatization

In [44]:
```
'''Set transformer pipeline 2:
Caseloading, normalization (using textacy), special ch removal,
lemmitization, stop word removal, rejoin,
remove custom punctuation, tokenize
'''
transformers02 = [{str.lower: None},
                  {normalize: None},
                  {rex_replace: None},
                  {lemma: None},
                  {" ".join: None},
                  {rex_url: None},
                  {emoji_split: None},
                  {str.split: None},
                  {sw_remover: None},
                  {" ".join: None},
                  {remove_punctuation: tw_punct},
                  {str.split: None},
                  {" ".join: None},
                 ]

# Apply transformers to pandas dataframe, w/ new col containing tokens
slct_tbl_full_df04['processed_lemmas'] = slct_tbl_full_df04['article_text']\
.progress_apply(prepare, pipeline=transformers02)

slct_tbl_full_df04['processed_lemmas_split'] = slct_tbl_full_df04['processed_lemmas
.progress_apply(str.split)

slct_tbl_full_df04['num_lemmas'] = slct_tbl_full_df04['processed_lemmas_split']\
.map(len)

display(slct_tbl_full_df04.head())

# Review unique tokens across entire dataset
for c in range(0,1):
    try:
        print(slct_tbl_full_df04['processed_lemmas'][c], '\n')
    except:
        print(f'Skip {c}')
```

```
100%|██████████| 4026/4026 [08:21<00:00,  8.02it/s]
100%|██████| 4026/4026 [00:00<00:00, 33122.67it/s]
```

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

traveler alabama drive interstate 65 party barbecue memorial day might see message d
igital road sign honor veteran die fight united states driver near clanton ala see m
onday instead motorist report see sign apparently hack display word reclaim america
white nationalist slogan patriot front reference white supremacist group involve dea
dly 2017 unite right rally charlottesville come write sarah hughes motorist capture
photo sign post twitter weird hell contractor s portable message board hack 65 chilt
on county ala monday afternoon john mcwilliam spokesman alabama department transport
ation aldot west central region tell washington post statement citizen alert nearby
state trooper message contact aldot mcwilliam say tuesday aldot personnel immediatel
y respond turn message board message board 65 affect mcwilliams add aldot investigat
e white supremacist language appear sign near clanton 40 mile northwest montgomery a
la official give immediate indication responsible apparently hack interstate sign fi
rst report alcom hughes tell post drive home birmingham weekend alabama s gulf coast
see white supremacist message recently pop around home city supporter patriot front
see think oh guy say hughes 31 year old attorney kind shocked hack alabama road sign
come time president biden declare white supremacy dangerous terrorist threat country
commencement address howard university month biden tell graduating class historicall
y black university pledge stand poison white supremacy inaugural address single dang
erous terrorist threat homeland white supremacy tell progress toward justice often m
eet ferocious pushback old sinister force biden say may 13 address quote donald trum
p s equivocate response 2017 rally charlottesville kill 32 year old heather heyer in
jure 19 hate never go away biden call white supremacy great terrorism threat 2024 ra
ce heat southern poverty law center splc track least 13 hate group alabama 2021 incl
ude proud boy discussion surround white supremacist white nationalist alabama intens
ify month sen tommy tuberville r ala say people identify white extremist white natio
nalist allow serve us armed force ask reporter wbhm birmingham whether white nationa
list allow serve military tuberville reply well call call americans tuberville criti
cize spokesman tell post senator resent implication people military anything patriot
hero gop senator say white nationalist military call americans patriot front white s
upremacist group whose name display interstate sign texas base hate group break vang
uard america form charlottesville rally splc say member chant reclaim america rally
coeur dalene idaho washington boston recent year accord report patriot front respons
ible vast majority white supremacist propaganda distribute united states since 2019
accord anti defamation league first time language promote patriot front make way pub
lic space alabama july graffiti beneath birmingham bridge appear patriot front spray
paint red blue letter alcom report patriot front graffiti also spot birmingham city
population nearly 70 percent black accord us census datum photo post twitter month s
how patriot front graffiti along red mountain expressway birmingham word dare defend
right patriot front graffiti later remove message leave sydney duncan attorney direc
tor magic legal center birmingham sadden hate become public part alabama white supre
macy alive well duncan write hughe say travel north birmingham pull 65 take photo me
ssage sign see confederate monument flag drive kind message government property diff
erent say police officer already scene wave keep drive hughes add return home hughes
say felt compel share image due ongoing conversation happen among birmingham residen
t promotion patriot front public space people might perceive upsetting scary sign wo
rsening country say strategy really impressed add die breed toluse olorunnipa azi pa
ybarah contribute report

Display globally unqiue tokens on final tokens

In [45]:
```
#uniq_tok(df_col=slct_tbl_full_df04['processed_lemmas_split'])
```

Calculate concentration ratio of each set of corpora

In [46]:
```
display(slct_tbl_full_df04['political_lean'].value_counts())

slct_tbl_full_df04_left = slct_tbl_full_df04[\
                                                slct_tbl_full_df04[\
                                                   'political_lean'] == 'left']

print(slct_tbl_full_df04_left.shape)
#display(slct_tbl_full_df04_left.head())

slct_tbl_full_df04_right = slct_tbl_full_df04[\
                                                 slct_tbl_full_df04[\
                                                   'political_lean'] == 'right']

print(slct_tbl_full_df04_right.shape)
#display(slct_tbl_full_df04_right.head())

slct_tbl_full_df04_left_s1 = list(itertools.chain.from_iterable(
    list(pd.Series(slct_tbl_full_df04_left['processed_text_split']))))
print(slct_tbl_full_df04_left_s1[:10])
slct_tbl_full_df04_right_s1 = list(itertools.chain.from_iterable(
    list(pd.Series(slct_tbl_full_df04_right['processed_text_split']))))
print(slct_tbl_full_df04_right_s1[:10])
```

```
right    2758
left     1268
Name: political_lean, dtype: int64
(1268, 17)
(2758, 17)
['travelers', 'alabama', 'driving', 'interstate', '65', 'parties', 'barbecues', 'mem
orial', 'day', 'might']
['family', 'jennifer', 'farber', 'dulos', 'released', 'statement', 'wednesday', 'mar
king', 'four', 'years']
```

In [47]:
```
def concen_ratio(artist_lst=[],
                 lsts=[]):
    lyr_corp_lst = []
    for l in lsts:
        print(type(l))
        lyr_corp_lst.append(' '.join(l))
    print(len(lyr_corp_lst))
    #print(lyr_corp_lst)

    cv = CountVectorizer(input='content',
                         encoding='utf-8',
                         stop_words=None,
                         token_pattern=r'\S+'
                        )

    lyr_tokens_fit = cv.fit(lyr_corp_lst)

    print(pd.Series(cv.get_feature_names_out()).sample(15))

    lyr_tokens_sm = cv.transform(lyr_corp_lst)
    display(lyr_tokens_sm)

    df = pd.DataFrame(lyr_tokens_sm.toarray(),
```

```
                    columns=cv.get_feature_names_out())
#display(df)

df02 = df.copy()
df02['r_sum'] = df02.sum(axis=1)
#display(df02)

'''Filter by frequency for all columns citation:
OpenAI. (2021). ChatGPT [Computer software]. https://openai.com/;
https://pandas.pydata.org/pandas-docs/stable/reference/api/
pandas.DataFrame.ge.html'''
condition = df.ge(5).all()
#print(condition)

# Get the list of columns that satisfy the condition
columns = condition[condition].index.tolist()
#print(columns)
columns.append('r_sum')
#print(columns)

#display(df02[columns])

df03 = df02[columns].copy()
display(df03)

'''Filter by frequency for all columns & add summary row citation:
OpenAI. (2021). ChatGPT [Computer software]. https://openai.com/'''
df04 = df03.apply(lambda x: x / df03.iloc[:,-1], axis=0)
#display(df04)

# Create new rows by dividing one artist row by the second artists row
new_row01 = df04.iloc[0] / df04.iloc[1]
new_row02 = df04.iloc[1] / df04.iloc[0]

# Append the new row to the DataFrame
df04 = df04.append(new_row01, ignore_index=True)
df04 = df04.append(new_row02, ignore_index=True)
display(df04)

# Transpose dataframe
df05 = df04.T
df05 = df05.reset_index()
df05.columns = ['token',
                'c1_concen',
                'c2_concen',
                'c1c2_concen_ratio',
                'c2c1_concen_ratio']
#print(df05)

'''Sort values citation:
https://pandas.pydata.org/pandas-docs/stable/reference/api
/pandas.DataFrame.sort_values.html'''
print(artist_lst[0])
display(df05[['token',
              'c1c2_concen_ratio']].sort_values(by='c1c2_concen_ratio',
                                                ascending=False).head(10))
```

```
    print(artist_lst[1])
    display(df05[['token',
                  'c2c1_concen_ratio']].sort_values(by='c2c1_concen_ratio',
                                                    ascending=False).head(10))

concen_ratio(artist_lst=['Left-Right Concentration Ratio',
                         'Right-Left Concentration Ratio'],
             lsts=[slct_tbl_full_df04_left_s1,
                   slct_tbl_full_df04_right_s1])
```

```
<class 'list'>
<class 'list'>
2
16213        distinctly
49659         theranoss
22971           haefele
19687       federations
51969           unknown
46495            solier
26885              ipos
36170    overperforming
40583           qureshi
48255         summarizes
39793         profoundly
19679                fed
14312           daylong
36296           oyelowo
19501          fascistic
dtype: object
<2x55719 sparse matrix of type '<class 'numpy.int64'>'
        with 79419 stored elements in Compressed Sparse Row format>
```

	#2	#metoo	0	07	1	10	100	1000	10000	100000	...	zero	zip	zone	zones
0	6	18	16	5	452	397	138	65	52	45	...	67	6	38	7
1	5	8	36	6	600	638	226	82	124	64	...	97	11	38	11

2 rows × 10959 columns

```
C:\Users\acarr\AppData\Local\Temp\ipykernel_21224\3142308763.py:59: FutureWarning: T
he frame.append method is deprecated and will be removed from pandas in a future ver
sion. Use pandas.concat instead.
  df04 = df04.append(new_row01, ignore_index=True)
C:\Users\acarr\AppData\Local\Temp\ipykernel_21224\3142308763.py:60: FutureWarning: T
he frame.append method is deprecated and will be removed from pandas in a future ver
sion. Use pandas.concat instead.
  df04 = df04.append(new_row02, ignore_index=True)
```

	#2	#metoo	0	07	1	10	100	1000	
0	7.9343e-06	2.3803e-05	2.1158e-05	6.6119e-06	0.0006	0.0005	0.0002	8.5955e-05	6.87
1	4.7075e-06	7.5320e-06	3.3894e-05	5.6490e-06	0.0006	0.0006	0.0002	7.7203e-05	1.16
2	1.6855e+00	3.1602e+00	6.2424e-01	1.1705e+00	1.0581	0.8740	0.8576	1.1134e+00	5.89
3	5.9331e-01	3.1643e-01	1.6019e+00	8.5437e-01	0.9451	1.1442	1.1660	8.9818e-01	1.697

4 rows × 10959 columns

`Left-Right Concentration Ratio`

	token	c1c2_concen_ratio
723	anonymity	17.5569
9885	thomass	16.2928
10775	willis	14.3264
6202	mehta	13.7334
3116	docket	13.2028
4350	ginni	12.3600
7422	pork	11.7046
10225	uncertainty	10.0479
4401	gorsuch	9.8319
4874	hush	9.8319

`Right-Left Concentration Ratio`

	token	c2c1_concen_ratio
1945	click	88.7452
2395	copyright	22.2745
6726	nyc	17.3246
8201	reparations	16.7313
2076	commenting	14.2394
755	ap	14.0812
765	app	13.0920
579	aliens	12.5782
1699	ccp	11.3915
5911	locker	10.9644

KWIC

In [48]:
```
def kwic(doc_series, keyword, window=35, print_samples=5):
    '''Search article text for keywords in context (KWIC)'''
    def add_kwic(text):
        kwic_list.extend(keyword_in_context(doc=text,
                                 keyword=keyword,
                                 ignore_case=True,
                                 window_width=window))
    kwic_list = []
    doc_series.map(add_kwic)

    if print_samples is None or print_samples==0:
        return kwic_list
    else:
        k = min(print_samples, len(kwic_list))
        print(f"{k} random samples out of {len(kwic_list)}" + \
              f"contexts for '{keyword}':")
        for sample in random.sample(list(kwic_list), k):
            print(re.sub(r'[\n\t]', ' ', sample[0])+' '+\
                  sample[1]+' '+\
                  re.sub(r'[\n\t]', ' ', sample[2]))
```

In [49]:
```
kwic(slct_tbl_full_df04['article_text'], 'ginni thomas')
```

```
5 random samples out of 33contexts for 'ginni thomas':
ss to be done.” The effort to keep  Ginni Thomas ’s name off paperwork makes the arr
o’s move to obscure the payment to  Ginni Thomas  came a little more than a year aft
nd The Polling Company, along with  Ginni Thomas ’s help, has been an invaluable res
ill a nonprofit he advises and pay  Ginni Thomas  $25,000. He also told her, “No men
ost, saying, “It is no secret that  Ginni Thomas  has a long history of working on i
```

Train/test split

In [50]:
```
slct_tbl_full_df04['stratifier'] = slct_tbl_full_df04['political_lean']\
.astype(str) + slct_tbl_full_df04['source_name'].astype(str)
slct_tbl_full_df04['stratifier'] = slct_tbl_full_df04['stratifier']\
.map(str.lower)
display(slct_tbl_full_df04.head())

y01a = ['stratifier']
slct_tbl_full_df04_y01_vc01a = slct_tbl_full_df04[y01a].to_numpy()
print(slct_tbl_full_df04_y01_vc01a.shape)

y01 = ['political_lean']
slct_tbl_full_df04_y01_vc01 = slct_tbl_full_df04[y01].to_numpy()
print(slct_tbl_full_df04_y01_vc01.shape)
```

	index	source_name	author	title	u
0	0	The Washington Post	NaN	Alabama Highway sign hacked with white suprema...	https://www.washingtonpost.com/nation/2023/05/
1	1	The Washington Post	Amber Phillips	Breaking down the GOP investigation into the B...	https://www.washingtonpost.com/politics/2023/0
2	2	The Washington Post	David Ovalle	Appeals court paves way for Purdue Pharma opio...	https://www.washingtonpost.com/health/2023/05/
3	3	The Washington Post	Philip Bump	Trump pledges to win an immigration fight he d...	https://www.washingtonpost.com/politics/2023/0
4	5	The Washington Post	Paul Waldman	Why fear of change will drive the GOP presiden...	https://www.washingtonpost.com/opinions/2023/0

```
(4026, 1)
(4026, 1)
```

In [51]:
```
nlm_train_x01, \
nlm_test_x01, \
nlm_train_y01, \
nlm_test_y01 = train_test_split(slct_tbl_full_df04['processed_text'],
                                slct_tbl_full_df04_y01_vc01,
                                test_size=.15,
                                random_state=1699,
                                stratify=slct_tbl_full_df04_y01_vc01a
                               )

nlm_train_y01 = nlm_train_y01.ravel()
nlm_test_y01 = nlm_test_y01.ravel()

print(f'{nlm_train_x01.shape}')
print(f'{nlm_train_y01.shape}')
print(f'\n{nlm_test_x01.shape}')
print(f'{nlm_test_y01.shape}')
```

```
(3422,)
(3422,)

(604,)
(604,)
```

In [52]:
```
lem_train_x01, \
lem_test_x01, \
lem_train_y01, \
lem_test_y01 = train_test_split(slct_tbl_full_df04['processed_lemmas'],
                                slct_tbl_full_df04_y01_vc01,
                                test_size=.15,
                                random_state=1699,
                                stratify=slct_tbl_full_df04_y01_vc01a
                               )

lem_train_y01 = lem_train_y01.ravel()
lem_test_y01 = lem_test_y01.ravel()

print(f'{lem_train_x01.shape}')
print(f'{lem_train_y01.shape}')
print(f'\n{lem_test_x01.shape}')
print(f'{lem_test_y01.shape}')
```

```
(3422,)
(3422,)

(604,)
(604,)
```

TF-IDF

In [53]:
```
print(slct_tbl_full_df04['processed_text'].shape)
print(slct_tbl_full_df04['processed_text'].head())
```

```
(4026,)
0    travelers alabama driving interstate 65 partie...
1    federal prosecutor may nearing decision whethe...
2    federal appeals court tuesday cleared way drug...
3    speaking orlando november 2015 republican pres...
4    look know countrys going wrong direction flori...
Name: processed_text, dtype: object
```

In [54]:
```
nlm_tfidf = TfidfVectorizer(encoding='utf-8',
                            analyzer='word',
                            stop_words=sw,
                            token_pattern=r'(?u)\b\w\w+\b',
                            ngram_range=(1,3),
                            max_df=.7,
                            min_df=5)

nlm_train_x01_mtx = nlm_tfidf.fit_transform(nlm_train_x01)
nlm_test_x01_mtx = nlm_tfidf.transform(nlm_test_x01)

display(nlm_train_x01_mtx)
display(nlm_test_x01_mtx)
```

```
<3422x49302 sparse matrix of type '<class 'numpy.float64'>'
        with 1275306 stored elements in Compressed Sparse Row format>
<604x49302 sparse matrix of type '<class 'numpy.float64'>'
        with 218436 stored elements in Compressed Sparse Row format>
```

In [55]:
```
lem_tfidf = TfidfVectorizer(encoding='utf-8',
                            analyzer='word',
                            stop_words=sw,
                            token_pattern=r'(?u)\b\w\w+\b',
                            ngram_range=(1,3),
                            max_df=.7,
                            min_df=5)

lem_train_x01_mtx = lem_tfidf.fit_transform(lem_train_x01)
lem_test_x01_mtx = lem_tfidf.transform(lem_test_x01)

display(lem_train_x01_mtx)
display(lem_test_x01_mtx)
```

```
<3422x54402 sparse matrix of type '<class 'numpy.float64'>'
        with 1316264 stored elements in Compressed Sparse Row format>
<604x54402 sparse matrix of type '<class 'numpy.float64'>'
        with 223719 stored elements in Compressed Sparse Row format>
```

In [56]:
```
def display_samp_dwm(sm=None,
                     vec=None,
                     n=(1,1),
                     rs_tup=(1,1)):
    mtx_df01 = pd.DataFrame(sm.toarray(),
                            columns=vec.get_feature_names_out())

    mtx_df01a = mtx_df01.sample(n=n[0],
                                random_state=rs_tup[0],
                                axis=1)

    mtx_df01b = mtx_df01a.sample(n=n[1],
                                 random_state=rs_tup[1],
                                 axis=0)

    display(mtx_df01b)
```

In [57]:
```
rs_tup=(1699,1699)

display_samp_dwm(sm=nlm_train_x01_mtx,
                 vec=nlm_tfidf,
                 n=(17,11),
                 rs_tup=(5,1699))

display_samp_dwm(sm=lem_train_x01_mtx,
                 vec=lem_tfidf,
                 n=(17,11),
                 rs_tup=(5,1699))
```

	banking committee	drug treatment	jake	million dollars	government service	among conservatives	humanitarian reasons significant	presid
1098	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2370	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1091	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3084	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2413	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
287	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
855	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2084	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1537	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
991	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2245	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

	release country	time republican	school shooter	blowout	australian government	zenny phuong	prepublication classification	cl republic
1098	0.0	0.0	0.0	0.0	0.0	0.0	0.0000	0.00
2370	0.0	0.0	0.0	0.0	0.0	0.0	0.0000	0.00
1091	0.0	0.0	0.0	0.0	0.0	0.0	0.0000	0.00
3084	0.0	0.0	0.0	0.0	0.0	0.0	0.0000	0.00
2413	0.0	0.0	0.0	0.0	0.0	0.0	0.0000	0.02
287	0.0	0.0	0.0	0.0	0.0	0.0	0.0387	0.00
855	0.0	0.0	0.0	0.0	0.0	0.0	0.0000	0.00
2084	0.0	0.0	0.0	0.0	0.0	0.0	0.0000	0.00
1537	0.0	0.0	0.0	0.0	0.0	0.0	0.0000	0.00
991	0.0	0.0	0.0	0.0	0.0	0.0	0.0000	0.00
2245	0.0	0.0	0.0	0.0	0.0	0.0	0.0000	0.00

Modeling

Algorithm setup

Gradient Boosting Classifier - Using BayesSearchCV

In [58]:
```
from sklearn.model_selection import RepeatedStratifiedKFold

# Start timer script
start_time = dt.datetime.today()

# Citation: Hochberg, 2018; Shanmukh, 2021
m2v1_gbc_pip = Pipeline([('gbc',
                          GradientBoostingClassifier(random_state=1699))])

loss_hparam = Categorical(['log_loss', 'exponential'])
lrate_hparam = Real(1e-3, 1e3, prior='log-uniform')
nest_hparam = Integer(1e2, 1e3, prior='log-uniform')
msamp_hparam = Real(.01, .95, prior='log-uniform')
mdepth_hparam = Integer(1, 20, prior='log-uniform')
mfeat_hparam = Categorical(['sqrt', 'log2', None])
#wstart_hparam = Categorical([True, False])
#calph_hparam = Real(0.0, 100.0, prior='log-uniform')
                #'gbc__warm_start': wstart_hparam
                #'gbc__ccp_alpha': calph_hparam

m2v1_gbc_grd = {'gbc__loss': loss_hparam,
                'gbc__learning_rate': lrate_hparam,
                'gbc__n_estimators': nest_hparam,
                'gbc__min_samples_split': msamp_hparam,
                'gbc__max_depth': mdepth_hparam,
                'gbc__max_features': mfeat_hparam
               }

'''Change GBC default scoring from accuracy to F1 score citation:
https://chat.openai.com/share/254f382b-4a8e-48e8-acd5-2918f0bbc59d
'''
f1_scorer = make_scorer(f1_score,
                        pos_label='right')

'''Customize cross-validation citation:
https://machinelearningmastery.com
/scikit-optimize-for-hyperparameter-tuning-in-machine-learning/
'''
cv = RepeatedStratifiedKFold(n_splits=5,
                             n_repeats=2,
                             random_state=1699)

m2v1_gbc = BayesSearchCV(m2v1_gbc_pip,
                         m2v1_gbc_grd,
                         n_iter=15,
                         scoring=f1_scorer,
                         cv=cv,
                         n_jobs=3,
                         refit=True,
                         verbose=4,
                         random_state=1699)
```

```
m2v1_gbc.fit(nlm_train_x01_mtx, nlm_train_y01)

# End timer script
end_time = dt.datetime.today()
time_elapse = end_time - start_time
print(f'Start Time = {start_time}')
print(f'End Time = {end_time}')
print(f'Elapsed Time = {time_elapse}')
```

```
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Fitting 10 folds for each of 1 candidates, totalling 10 fits
Start Time = 2023-06-18 16:04:48.076748
End Time = 2023-06-18 17:07:33.173517
Elapsed Time = 1:02:45.096769
```

In [59]:
```
print(f'\nBest Estimator:\n{m2v1_gbc.best_estimator_}')

print('\nCross-validaton results:')
display(pd.DataFrame(m2v1_gbc.cv_results_))

train_m2v1_gbc_y01_pred = m2v1_gbc.predict_proba(nlm_train_x01_mtx)
print(f'\nFirst 10 train set predictions:\n{train_m2v1_gbc_y01_pred[:10]}')

test_m2v1_gbc_y01_pred = m2v1_gbc.predict_proba(nlm_test_x01_mtx)
print(f'\nFirst 10 test set predictions:\n{test_m2v1_gbc_y01_pred[:10]}')

print(f'\nBest Score for "{m2v1_gbc.scorer_}" is {m2v1_gbc.best_score_}')
```

```
Best Estimator:
Pipeline(steps=[('gbc',
                 GradientBoostingClassifier(learning_rate=1.443983517854453,
                                            loss='exponential', max_depth=20,
                                            max_features='log2',
                                            min_samples_split=0.5120315024377693,
                                            n_estimators=1000,
                                            random_state=1699))])

Cross-validaton results:
```

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_gbc__learning_rate
0	12.6265	1.2722	0.0266	7.1826e-03	0.0162
1	11.8357	1.1532	0.0282	6.1911e-03	0.035
2	3.2453	0.3068	0.0250	7.6687e-03	55.3871
3	317.1236	21.4360	0.0313	7.3904e-06	4.7311
4	6.3836	0.5247	0.0399	1.0156e-02	25.6546
5	13.6031	1.3130	0.0554	1.0068e-02	23.2484
6	12.4504	1.2054	0.0500	9.3930e-03	173.3432
7	237.9337	23.0674	0.0266	7.1457e-03	36.8407
8	7.4626	1.0567	0.0299	4.9103e-03	274.3051
9	4.3345	0.4789	0.0372	7.5743e-03	0.5714
10	5.3237	0.5666	0.0415	7.1466e-03	0.3414
11	36.5412	5.3809	0.0711	1.6271e-02	1.6801
12	32.2294	3.3634	0.0548	1.2631e-02	1.444
13	253.3072	32.7042	0.0281	6.2242e-03	5.7333
14	30.3414	3.6930	0.0501	1.1672e-02	0.001

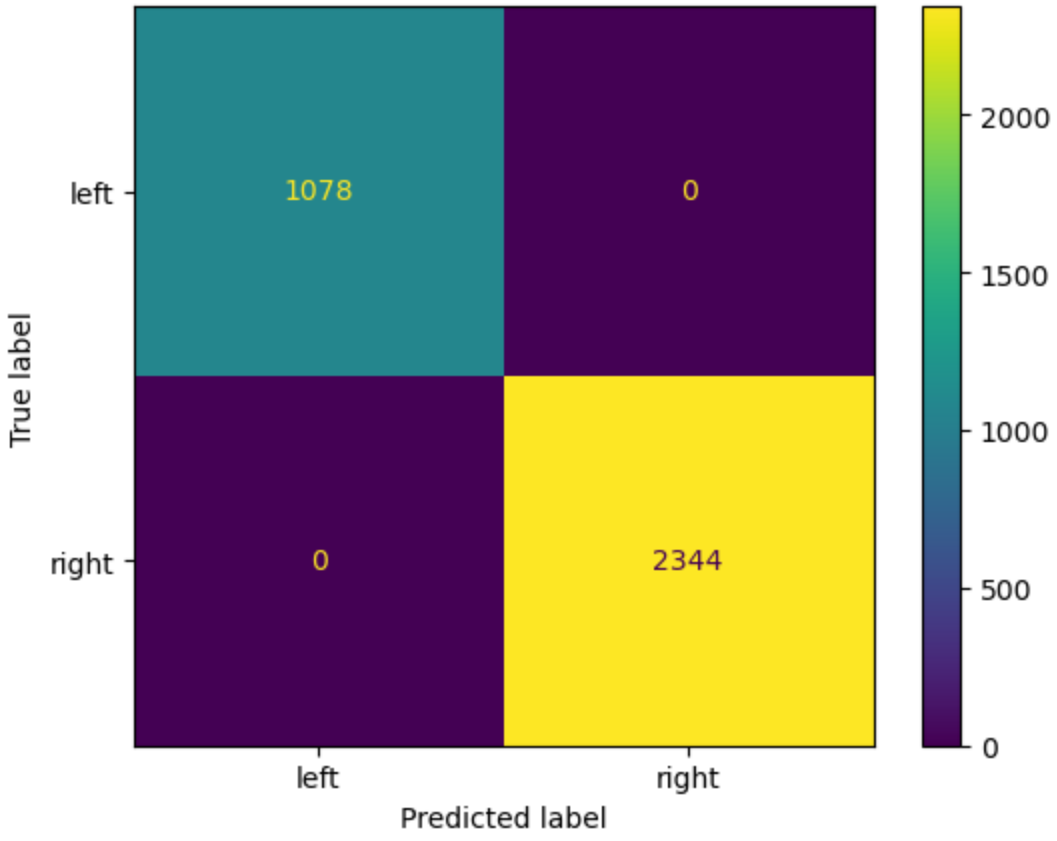
15 rows × 24 columns

```
First 10 train set predictions:
[[0. 1.]
 [0. 1.]
 [1. 0.]
 [0. 1.]
 [0. 1.]
 [0. 1.]
 [0. 1.]
 [0. 1.]
 [0. 1.]
 [1. 0.]]

First 10 test set predictions:
[[0.     1.    ]
 [0.     1.    ]
 [1.     0.    ]
 [1.     0.    ]
 [1.     0.    ]
 [0.     1.    ]
 [0.9975 0.0025]
 [1.     0.    ]
 [1.     0.    ]
 [0.6303 0.3697]]
```

Train set check

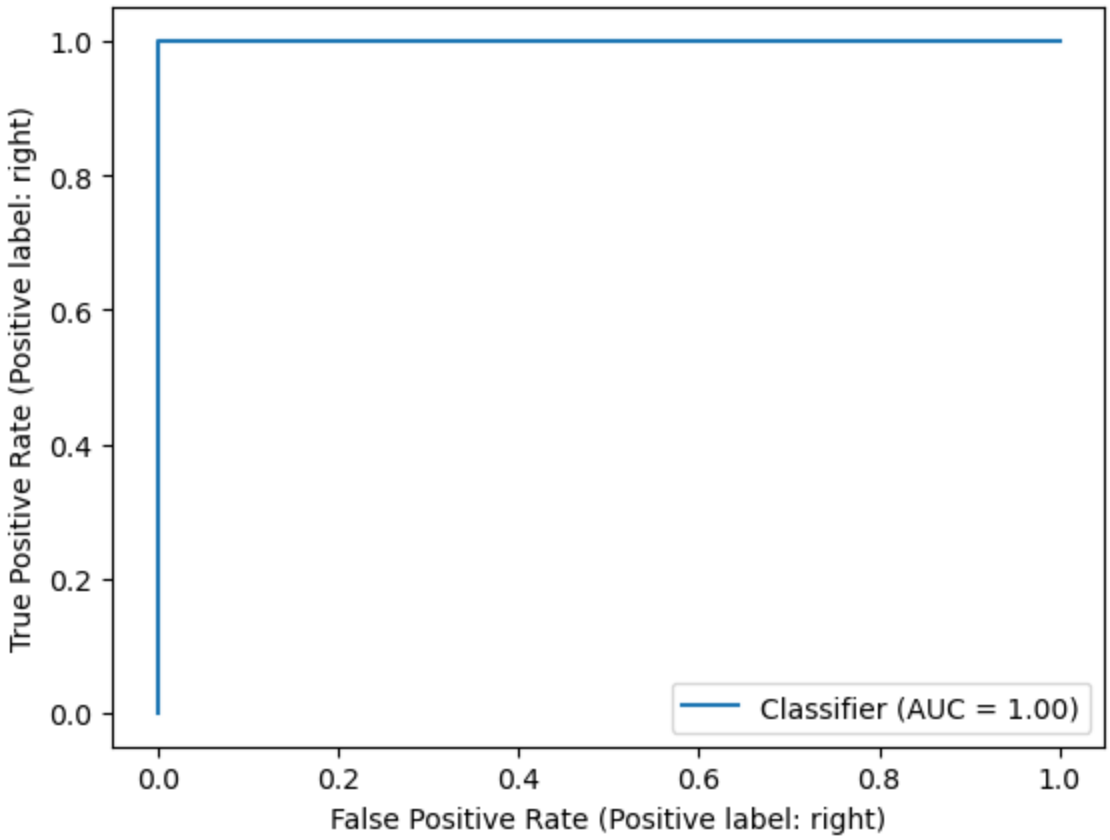
In [60]:
```
nlm_train_y01_pred = m2v1_gbc.predict(nlm_train_x01_mtx)
nlm_train_y01_pred_cm = confusion_matrix(nlm_train_y01, nlm_train_y01_pred)

print(classification_report(nlm_train_y01, nlm_train_y01_pred))
print(nlm_train_y01_pred_cm)

'''Citation:
https://scikit-learn.org/stable/modules/generated
/sklearn.metrics.ConfusionMatrixDisplay.html
#sklearn.metrics.ConfusionMatrixDisplay.plot
'''
nlm_train_cm_dsp = ConfusionMatrixDisplay(confusion_matrix=nlm_train_y01_pred_cm,
                                          display_labels=m2v1_gbc.classes_)
nlm_train_cm_dsp.plot()
plt.show()
```

```
              precision    recall  f1-score   support

        left       1.00      1.00      1.00      1078
       right       1.00      1.00      1.00      2344

    accuracy                           1.00      3422
   macro avg       1.00      1.00      1.00      3422
weighted avg       1.00      1.00      1.00      3422

[[1078    0]
 [   0 2344]]
```



ROC-AUC Curve

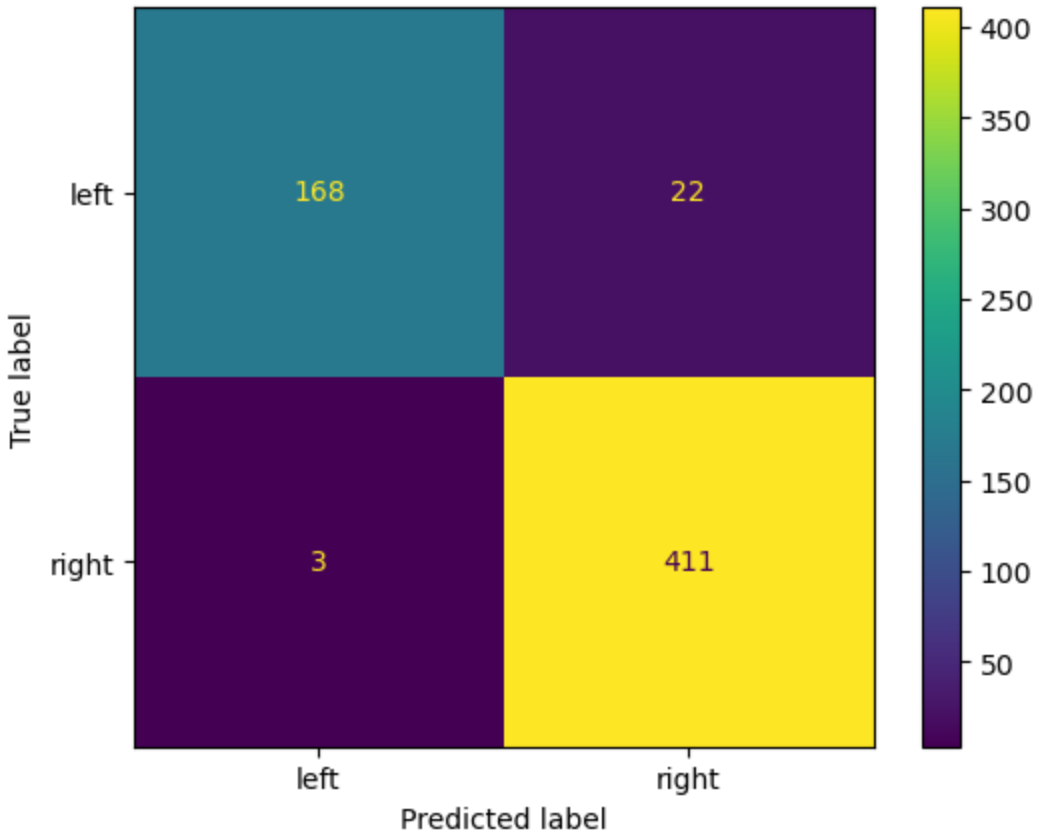
In [61]:
```
nlm_train_y01_pred_decf = m2v1_gbc.decision_function(nlm_train_x01_mtx)
RocCurveDisplay.from_predictions(nlm_train_y01, nlm_train_y01_pred_decf,
                                 pos_label='right')
plt.show()
```



Test set results

In [62]:
```
nlm_test_y01_pred = m2v1_gbc.predict(nlm_test_x01_mtx)
nlm_test_y01_pred_cm = confusion_matrix(nlm_test_y01, nlm_test_y01_pred)

print(classification_report(nlm_test_y01, nlm_test_y01_pred))
print(nlm_test_y01_pred_cm)

'''Citation:
https://scikit-learn.org/stable/modules/generated
/sklearn.metrics.ConfusionMatrixDisplay.html
#sklearn.metrics.ConfusionMatrixDisplay.plot
'''
nlm_test_cm_dsp = ConfusionMatrixDisplay(confusion_matrix=nlm_test_y01_pred_cm,
                                         display_labels=m2v1_gbc.classes_)
nlm_test_cm_dsp.plot()
plt.show()
```

```
              precision    recall  f1-score   support

        left       0.98      0.88      0.93       190
       right       0.95      0.99      0.97       414

    accuracy                           0.96       604
   macro avg       0.97      0.94      0.95       604
weighted avg       0.96      0.96      0.96       604

[[168  22]
 [  3 411]]
```



Pickle best model

In [63]:
```
# Path to save the pickled model
file_path = "m2v1_gbc.pkl"

# Pickle the model
with open(file_path, "wb") as file:
    pickle.dump(m2v1_gbc, file)

print("Model pickled and saved successfully.")
```

```
Model pickled and saved successfully.
```

Business problem application

In [64]:
```
center_df01 = pd.read_csv(file_in_path02)

print(center_df01.shape)
display(center_df01.head())
```

```
(181, 12)
```

	Unnamed: 0.1	Unnamed: 0	Source	Author	Title	
0	0	0	The Hill	Zach Schonfeld	Ketanji Brown Jackson issues solo dissent in r...	https://thehill.com/regulation/cou
1	1	1	The Hill	Brett Samuels	How Biden pulled it off...	https://thehill.com/homenews/admir
2	2	2	The Hill	the hill	How Christie could be wildcard in 2024 race...	https://thehill.com/homenews/campaig
3	3	3	The Hill	Alexander Bolton	Schumer announces agreement to pass debt ceili...	https://thehill.com/homenews/senate/
4	4	4	The Hill	Zack Budryk	Kaine introduces amendment to strip Manchin-ba...	https://thehill.com/policy/energy-er

In [65]:
```
# Apply transformers to pandas dataframe, w/ new col containing tokens
center_df01['processed_text'] = center_df01['article_text']\
.progress_apply(prepare, pipeline=transformers01)

center_df01['processed_text_split'] = center_df01['processed_text']\
.progress_apply(str.split)

center_df01['num_tokens'] = center_df01['processed_text_split']\
.map(len)

display(center_df01.head())
```

```
# Review unique tokens across entire dataset
for c in range(0,1):
    try:
        print(center_df01['processed_text'][c], '\n')
    except:
        print(f'Skip {c}')
```

```
100%|██████████| 181/181 [00:00<00:00, 206.25it/s]
100%|██████████| 181/181 [00:00<?, ?it/s]
```

	Unnamed: 0.1	Unnamed: 0	Source	Author	Title	
0	0	0	The Hill	Zach Schonfeld	Ketanji Brown Jackson issues solo dissent in r...	https://thehill.com/regulation/cou
1	1	1	The Hill	Brett Samuels	How Biden pulled it off...	https://thehill.com/homenews/admin
2	2	2	The Hill	the hill	How Christie could be wildcard in 2024 race...	https://thehill.com/homenews/campaig
3	3	3	The Hill	Alexander Bolton	Schumer announces agreement to pass debt ceili...	https://thehill.com/homenews/senate/
4	4	4	The Hill	Zack Budryk	Kaine introduces amendment to strip Manchin-ba...	https://thehill.com/policy/energy-en

```
liberal justice ketanji brown jackson issued first solo dissent supreme court merits
case thursday disagreeing colleagues labor dispute ruling makes easier companies sue
worker strikes 8 1 decision high court overturned lower ruling found federal union l
aws preempted concrete company glacier northwest bringing lawsuit international brot
herhood teamsters represents companys truck drivers jackson wrote courts no business
delving particular labor dispute time the majority also misapplies boards cases mann
er threatens impede boards uniform development labor law erode right strike jackson
dissented case arose union directed drivers go strike morning company mixing concret
e loading onto trucks making deliveries concrete mixed day ruined glacier sued union
damages state court 1959 supreme court precedent san diego building trades council v
garmon national labor relations act nlra federal law governs strikes collective barg
aining preempts state law two arguably conflict union got lawsuit tossed washington
supreme court garmon glacier northwest appealed nations highest court majority opini
on authored conservative justice amy coney barrett joined four colleagues court rule
d nlra preempt lawsuit strike take reasonable precautions protect companys property
foreseeable imminent danger the unions actions resulted destruction concrete glacier
prepared day also posed risk foreseeable aggravated imminent harm glaciers trucks un
ion took affirmative steps endanger glaciers property rather reasonable precautions
mitigate risk nlra arguably protect conduct barrett wrote three additional conservat
ive justices justices samuel alito clarence thomas neil gorsuch wrote separately rev
erse unions win grounds jackson hand stood alone dissenting marking first solo disse
nt merits case since joining bench last year jackson has however dissented solo outs
ide courts normal docket jpmorgan says former virgin islands first lady aided epstei
n trump indictment lays bare security risks storage mar lago noted complaint nationa
l labor relations boards general counsel filed state supreme courts ruling alleged g
lacier northwest engaged unfair labor practices relation strike majority found this
issue properly us lower courts addressed significance complaint leaving state courts
consider case proceeds the filing general counsels administrative complaint necessar
ily suffices establish unions strike conduct arguably protected within meaning garmo
n thus general counsels complaint marked end court involvement matter jackson wrote
```

In [66]:
```
nlm_apply_x01_mtx = nlm_tfidf.transform(center_df01['processed_text'])

print(nlm_apply_x01_mtx.shape)
display(nlm_apply_x01_mtx)
```

```
(181, 49302)
<181x49302 sparse matrix of type '<class 'numpy.float64'>'
        with 64944 stored elements in Compressed Sparse Row format>
```

In [67]:
```
display_samp_dwm(sm=nlm_apply_x01_mtx,
                 vec=nlm_tfidf,
                 n=(17,11),
                 rs_tup=(5,1699))
```

	banking committee	drug treatment	jake	million dollars	government service	among conservatives	humanitarian reasons significant	presi
27	0.0	0.0	0.0503	0.0	0.0	0.0	0.0	
152	0.0	0.0	0.0000	0.0	0.0	0.0	0.0	
154	0.0	0.0	0.0000	0.0	0.0	0.0	0.0	
131	0.0	0.0	0.0000	0.0	0.0	0.0	0.0	
75	0.0	0.0	0.0000	0.0	0.0	0.0	0.0	
77	0.0	0.0	0.0000	0.0	0.0	0.0	0.0	
170	0.0	0.0	0.0000	0.0	0.0	0.0	0.0	
121	0.0	0.0	0.0000	0.0	0.0	0.0	0.0	
43	0.0	0.0	0.0000	0.0	0.0	0.0	0.0	
48	0.0	0.0	0.0000	0.0	0.0	0.0	0.0	
53	0.0	0.0	0.0000	0.0	0.0	0.0	0.0	

In [68]:
```
nlm_apply_mtx_pred_prob = m2v1_gbc.predict_proba(nlm_apply_x01_mtx)

print(nlm_apply_mtx_pred_prob.shape)
print(nlm_apply_mtx_pred_prob[:10])

nlm_apply_mtx_pred = m2v1_gbc.predict(nlm_apply_x01_mtx)

print(nlm_apply_mtx_pred.shape)
print(nlm_apply_mtx_pred)
```

```
(181, 2)
[[1.     0.    ]
 [1.     0.    ]
 [1.     0.    ]
 [1.     0.    ]
 [1.     0.    ]
 [1.     0.    ]
 [0.0001 0.9999]
 [1.     0.    ]
 [1.     0.    ]
 [0.0174 0.9826]]
(181,)
['left' 'left' 'left' 'left' 'left' 'left' 'right' 'left' 'left' 'right'
 'left' 'left' 'left' 'right' 'left' 'right' 'left' 'left' 'right' 'left'
 'right' 'left' 'left' 'left' 'left' 'left' 'left' 'left' 'right' 'right'
 'left' 'left' 'right' 'left' 'left' 'left' 'left' 'left' 'left' 'left'
 'left' 'left' 'left' 'left' 'left' 'left' 'left' 'left' 'left' 'left'
 'left' 'right' 'right' 'left' 'left' 'left' 'left' 'left' 'left' 'right'
 'left' 'left' 'left' 'left' 'left' 'left' 'left' 'left' 'right' 'left'
 'left' 'left' 'left' 'left' 'left' 'left' 'left' 'left' 'left' 'left'
 'left' 'left' 'left' 'left' 'right' 'left' 'left' 'left' 'left' 'left'
 'left' 'left' 'left' 'right' 'left' 'left' 'left' 'left' 'left' 'left'
 'left' 'left' 'left' 'left' 'left' 'left' 'left' 'left' 'left' 'left'
 'left' 'left' 'right' 'left' 'left' 'left' 'right' 'left' 'left' 'left'
 'right' 'right' 'left' 'right' 'left' 'left' 'right' 'left' 'right'
 'right' 'right' 'left' 'right' 'left' 'right' 'left' 'right' 'left'
 'left' 'left' 'left' 'left' 'left' 'right' 'right' 'right' 'right' 'left'
 'right' 'left' 'right' 'left' 'left' 'right' 'left' 'left' 'left' 'left'
 'left' 'right' 'right' 'left' 'left' 'right' 'right' 'left' 'left' 'left'
 'right' 'left' 'right' 'right' 'left' 'left' 'left' 'left' 'left' 'right'
 'right' 'left' 'left']
```

In [69]:
```
# Compute the maximum values along the second dimension
max_values = np.amax(nlm_apply_mtx_pred_prob, axis=1)
max_values_df01 = pd.DataFrame(max_values,
                               columns=['decision_prob'])
max_values_df01['pred'] = nlm_apply_mtx_pred
print(max_values_df01.shape)
display(max_values_df01.head())
```

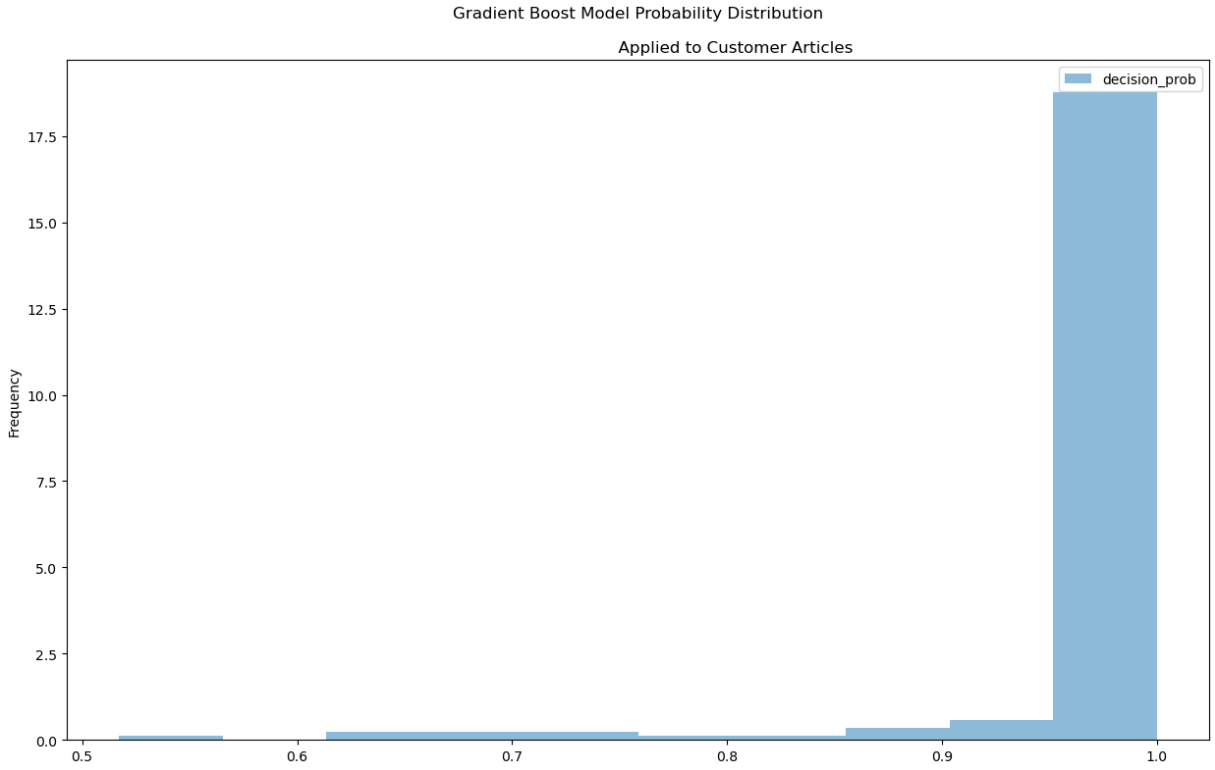
```
(181, 2)
```

	decision_prob	pred
0	1.0	left
1	1.0	left
2	1.0	left
3	1.0	left
4	1.0	left

In [70]:
```
max_values_df01['decision_prob'].plot(kind="hist",
                                      density=True,
                                      alpha=0.5,
```

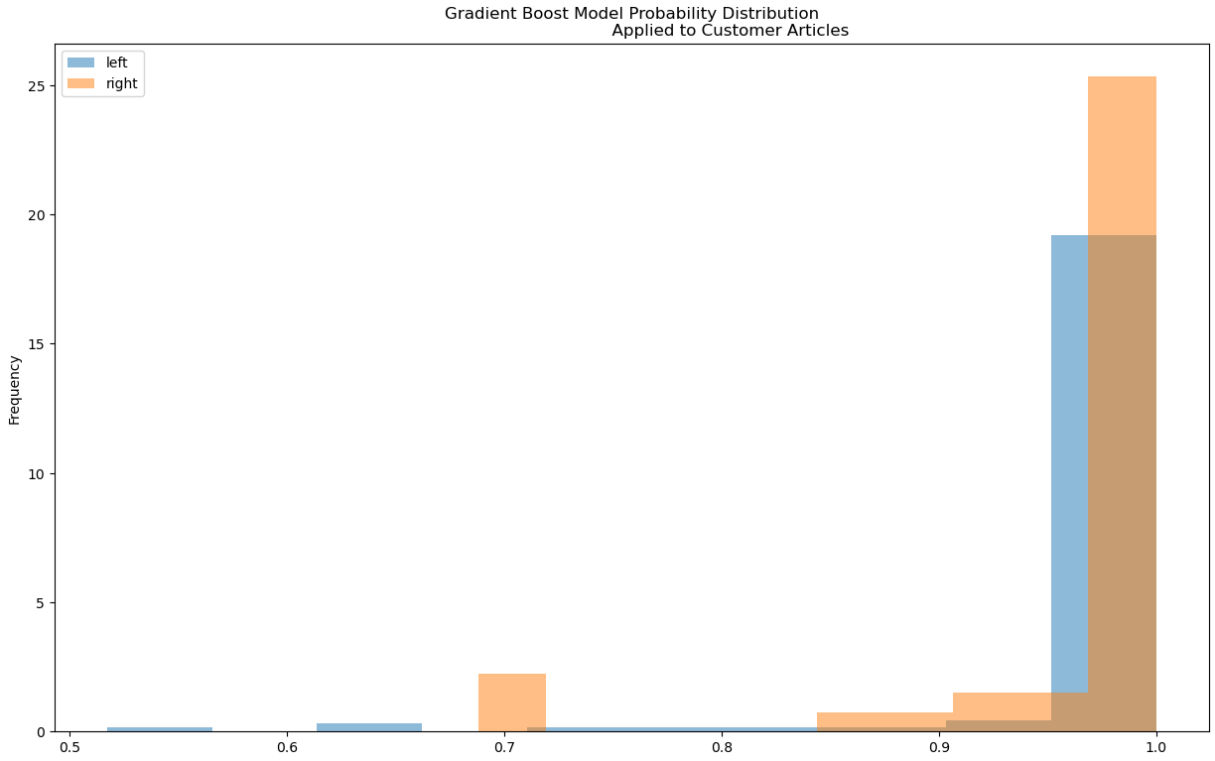
```
                                        legend=True,
                                        figsize=(15,9),
title='''Gradient Boost Model Probability Distribution\n
                                        Applied to Customer Articles''')
```

Out[70]:
```
<Axes: title={'center': 'Gradient Boost Model Probability Distribution\n\n
Applied to Customer Articles'}, ylabel='Frequency'>
```



In [71]:
```
max_values_df01.groupby('pred')['decision_prob'].plot(kind="hist",
                                        density=True,
                                        alpha=0.5,
                                        legend=True,
                                        figsize=(15,9),
                                        title='''Gradient Boost Model Probability Dis
                                        Applied to Customer Articles''')
```

Out[71]:
```
pred
left     Axes(0.125,0.11;0.775x0.77)
right    Axes(0.125,0.11;0.775x0.77)
Name: decision_prob, dtype: object
```


Gradient Boost Model Probability Distribution
Applied to Customer Articles
left
right
Frequency
0
5
10
15
20
25
0.5
0.6
0.7
0.8
0.9
1.0