```
import pandas as pd
import numpy as np
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
import string
from string import punctuation
import os
from os import listdir
from collections import Counter
from tensorflow.keras.preprocessing.text import Tokenizer
from nltk.tokenize import word_tokenize
from wordcloud import WordCloud
from textblob import TextBlob
from plotnine import *
from pandas.api.types import CategoricalDtype
```

```
def file_to_string(filename):
 '''Opens the input text file and
 returns a string of all its text.'''
 file = open(filename, 'r')
 text = file.read()
 file.close()
 text = text.replace('\n', ' ')
 text = text.replace('  ', ' ')
 return text
```

```
# cd ..
```

```
/Users/kamilapalys/Desktop/school/data450/capstone
```

```
pwd
```

```
'/Users/kamilapalys/Desktop/school/data450/capstone'
```

```
# example of using the function

filepath = 'data/text/cnn_trump.txt'
test_txt = file_to_string(filepath)
```

```
test_txt
```

```
"Donald Trump faces more than 30 counts related to business fraud in an indictment from a Ma
the first time in American history that a current or former president has faced criminal cha
which has never seen one of its ex-leaders confronted with criminal charges, let alone while
into uncharted waters. Trump released a statement in response to the indictment claiming it
united and strong - will first defeat Alvin Bragg, and then we will defeat Joe Biden, and we
or more - away. "Is this a shock today? Hell yes," the person said, speaking on a condition
as well as his 2024 GOP rivals - have condemned the Manhattan district attorney's office ove
```

```
# initialize a dictionary to be able to find the original word from the stemmed word
stemmed_dict = {}

# how to later access the key by the value
#value = {i for i in dic if dic[i]=="n"}
#print("key by value:",value)
```

```
def clean_text(text, stem):
    '''Takes in a string of text cleans it by converting
    to lowercase, removing punctuation, and removing stopwords.
    Also takes in a binary value to indicate if stemming should
    be performed. Returns the new string.'''
    if stem not in [0, 1]:
        raise ValueError("Stem must be a binary value (0 or 1)")
    ps = PorterStemmer()
    # create list of stopwords
    stopwords_list = stopwords.words('english')
    # make the text lowercase
    text = text.lower()
    text = text.replace('-', ' ')
    # convert to ascii characters
    text = text.encode("ascii", "ignore").decode()
    for chr in text:
        # only keep characters in the string that are not punctuation symbols
        if (chr in string.punctuation or chr in string.digits):
            text = text.replace(chr, ' ')
    text = text.replace('  ', ' ')
    # stem the tokens within the text
    tokens = text.split()
    new_tokens = []
```

```
    for token in tokens[:-2]:
        # only include new token in the cleaned list if not a stopword
        if token not in stopwords_list:
            if stem == 1:
                stemmed_word = ps.stem(token)
                new_tokens.append(stemmed_word)
                # to be able to map each token to the resulting stemmed word
                if token not in stemmed_dict:
                    stemmed_dict[token] = stemmed_word
            else:
                new_tokens.append(token)
    new_tokens.append(tokens[-2])
    new_tokens.append(tokens[-1])
    cleaned_text = " ".join(new_tokens)
    cleaned_text = cleaned_text.replace('  ', ' ')
    return cleaned_text
```

```
# looping through all text files to apply preprocessing functions
article_docs = []
dir = os.listdir('data/text/')
dir.sort()
for filename in dir:
    filepath = os.path.join('data/text/', filename)
    if filename.split(".")[-1] == "txt":
        article_string = file_to_string(filepath)
        new_string = clean_text(article_string, 1)
        article_docs.append(new_string)

# convert the list of article strings into a binary-value dataframe
t = Tokenizer()
t.fit_on_texts(article_docs)
print(t)
encoded_docs = t.texts_to_matrix(article_docs, mode='binary')
words = [x for x in t.word_index.keys()]
binary_df = pd.DataFrame(data = encoded_docs[:, 1:], columns=words)
# List of conditions
source_conditions = [
      binary_df['abcarticle'] == 1
    , binary_df['bbcarticle'] == 1
    , binary_df['cnnarticle'] == 1
    , binary_df['foxarticle'] == 1
```

```
    , binary_df['nbcarticle'] == 1
    , binary_df['nyparticle'] == 1
    , binary_df['nytarticle'] == 1
    , binary_df['wparticle'] == 1
    , binary_df['wsjarticle'] == 1
]

# List of values to return
source_choices  = [
      "ABC News"
    , "BBC"
    , "CNN"
    , "Fox News"
    , "NBC News"
    , "New York Post"
    , "The New York Times"
    , "The Washington Post"
    , "The Wall Street Journal"
]

# List of conditions
topic_conditions = [
      binary_df['affirmativearticle'] == 1
    , binary_df['balloonarticle'] == 1
    , binary_df['bidenarticle'] == 1
    , binary_df['hamasarticle'] == 1
    , binary_df['pentagonarticle'] == 1
    , binary_df['santosarticle'] == 1
    , binary_df['tanksarticle'] == 1
    , binary_df['trumparticle'] == 1
]
# List of values to return
topic_choices  = [
      "Supreme Court Ruling on Affirmative Action"
    , "Chinese Surveillance Balloon"
    , "Biden's Low Approval Rates in Polls"
    , "The Deadliest Attack by Hamas"
    , "Pentagon Documents Leak"
    , "George Santos' Expulsion from Congress"
    , "U.S. and Germany Send Tanks to Ukraine"
    , "Trump's Indictment"
```

```
]
# create a new source column
binary_df["article_source"] = np.select(source_conditions, source_choices, "ERROR")

# create a new topic column
binary_df["article_topic"] = np.select(topic_conditions, topic_choices, "ERROR")

binary_df.head()
```

```
<keras.src.preprocessing.text.Tokenizer object at 0x177e06250>
```

	said	trump	biden	offici	mr	u	israel	presid	tank	hous	...	messr	overlap	vs	convert
0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
1	1.0	0.0	1.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	...	0.0	0.0	0.0	0.0
2	1.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	...	0.0	0.0	0.0	0.0
3	1.0	0.0	1.0	1.0	0.0	1.0	1.0	1.0	0.0	1.0	...	0.0	0.0	0.0	0.0
4	1.0	0.0	0.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0	...	0.0	0.0	0.0	0.0

```
# check what word a given stemmed word represents
value = {i for i in stemmed_dict if stemmed_dict[i]=="marku"}
print("key by value:",value)
```

```
key by value: {'markus'}
```

```
encoded_docs_freq = t.texts_to_matrix(article_docs, mode='count')
freq_df = pd.DataFrame(data = encoded_docs_freq[:, 1:], columns=words)
# List of conditions
source_conditions = [
      freq_df['abcarticle'] == 1
    , freq_df['bbcarticle'] == 1
    , freq_df['cnnarticle'] == 1
    , freq_df['foxarticle'] == 1
    , freq_df['nbcarticle'] == 1
    , freq_df['nyparticle'] == 1
    , freq_df['nytarticle'] == 1
    , freq_df['wparticle'] == 1
    , freq_df['wsjarticle'] == 1
]
```

```
# List of values to return
source_choices  = [
      "ABC News"
    , "BBC"
    , "CNN"
    , "Fox News"
    , "NBC News"
    , "New York Post"
    , "The New York Times"
    , "The Washington Post"
    , "The Wall Street Journal"
]

# List of conditions
topic_conditions = [
      freq_df['affirmativearticle'] == 1
    , freq_df['balloonarticle'] == 1
    , freq_df['bidenarticle'] == 1
    , freq_df['hamasarticle'] == 1
    , freq_df['pentagonarticle'] == 1
    , freq_df['santosarticle'] == 1
    , freq_df['tanksarticle'] == 1
    , freq_df['trumparticle'] == 1
]
# List of values to return
topic_choices  = [
      "Supreme Court Ruling on Affirmative Action"
    , "Chinese Surveillance Balloon"
    , "Biden's Low Approval Rates in Polls"
    , "The Deadliest Attack by Hamas"
    , "Pentagon Documents Leak"
    , "George Santos' Expulsion from Congress"
    , "U.S. and Germany Send Tanks to Ukraine"
    , "Trump's Indictment"
]
# create a new source column
freq_df["article_source"] = np.select(source_conditions, source_choices, "ERROR")

# create a new topic column
freq_df["article_topic"] = np.select(topic_conditions, topic_choices, "ERROR")
```

```
freq_df.head()
```

	said	trump	biden	offici	mr	u	israel	presid	tank	hous	...	messr	overlap	vs	conver
0	5.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
1	30.0	0.0	5.0	28.0	0.0	15.0	0.0	4.0	0.0	3.0	...	0.0	0.0	0.0	0.0
2	9.0	17.0	27.0	0.0	0.0	0.0	0.0	8.0	0.0	4.0	...	0.0	0.0	0.0	0.0
3	17.0	0.0	3.0	6.0	0.0	10.0	28.0	2.0	0.0	1.0	...	0.0	0.0	0.0	0.0
4	9.0	0.0	0.0	5.0	0.0	20.0	2.0	2.0	3.0	0.0	...	0.0	0.0	0.0	0.0

```
# create dataframe with tf-idf values

encoded_docs_tfidf = t.texts_to_matrix(article_docs, mode='tfidf')
tfidf_df = pd.DataFrame(data = encoded_docs_tfidf[:, 1:], columns=words)
# List of conditions
source_conditions = [
      tfidf_df['abcarticle'] != 0
    , tfidf_df['bbcarticle'] != 0
    , tfidf_df['cnnarticle'] != 0
    , tfidf_df['foxarticle'] != 0
    , tfidf_df['nbcarticle'] != 0
    , tfidf_df['nyparticle'] != 0
    , tfidf_df['nytarticle'] != 0
    , tfidf_df['wparticle'] != 0
    , tfidf_df['wsjarticle'] != 0
]

# List of values to return
source_choices  = [
      "ABC News"
    , "BBC"
    , "CNN"
    , "Fox News"
    , "NBC News"
    , "New York Post"
    , "The New York Times"
    , "The Washington Post"
    , "The Wall Street Journal"
]

# List of conditions
```

```
topic_conditions = [
      tfidf_df['affirmativearticle'] != 0
    , tfidf_df['balloonarticle'] != 0
    , tfidf_df['bidenarticle'] != 0
    , tfidf_df['hamasarticle'] != 0
    , tfidf_df['pentagonarticle'] != 0
    , tfidf_df['santosarticle'] != 0
    , tfidf_df['tanksarticle'] != 0
    , tfidf_df['trumparticle'] != 0
]
# List of values to return
topic_choices  = [
      "Supreme Court Ruling on Affirmative Action"
    , "Chinese Surveillance Balloon"
    , "Biden's Low Approval Rates in Polls"
    , "The Deadliest Attack by Hamas"
    , "Pentagon Documents Leak"
    , "George Santos' Expulsion from Congress"
    , "U.S. and Germany Send Tanks to Ukraine"
    , "Trump's Indictment"
]
# create a new source column
tfidf_df["article_source"] = np.select(source_conditions, source_choices, "ERROR")

# create a new topic column
tfidf_df["article_topic"] = np.select(topic_conditions, topic_choices, "ERROR")

tfidf_df.head()
```

	said	trump	biden	offici	mr	u	israel	presid	tank	hous	..
0	1.827036	0.000000	0.000000	0.000000	0.0	2.313275	0.000000	0.000000	0.000000	0.000000	..
1	3.081563	0.000000	2.267700	4.139471	0.0	3.594586	0.000000	1.881491	0.000000	1.871958	..
2	2.238584	5.086179	3.733245	0.000000	0.0	0.000000	0.000000	2.428008	0.000000	2.128570	..
3	2.683881	0.000000	1.823774	2.667558	0.0	3.201528	6.446654	1.334974	0.000000	0.891998	..
4	2.238584	0.000000	0.000000	2.493348	0.0	3.873465	2.519533	1.334974	3.689062	0.000000	..

```
# create a list of strings where each string is all articles from one source
source_docs = []

j = 0
```

```
for i in range(9):
    source = " ".join(article_docs[j].split()[:-2]) + " " + " ".join(article_docs[j+1].spl
        + " ".join(article_docs[j+2].split()[:-2]) + " " + " ".join(article_docs[j+3].spli
        + " ".join(article_docs[j+4].split()[:-2]) + " " + " ".join(article_docs[j+5].spli
        + " ".join(article_docs[j+6].split()[:-2]) + " " + " ".join(article_docs[j+7].spli
    source_docs.append(source)
    j += 8

source_docs
```

```
['suprem court thursday set new limit affirm action program case involv whether public privat
 'us suprem court rule race longer consid factor univers admiss landmark rule upend decad old
 'suprem court say colleg univers longer take race consider specif basi grant admiss landmark
 'u suprem court hand major rule affirm action thursday reject use race factor colleg admiss
 'washington suprem court thursday struck affirm action program univers north carolina harvar
 'suprem court struck affirm action program harvard univers univers north carolina thursday r
 'chief justic john g robert jr finish read major opinion suprem court chamber thursday hush
 'suprem court thursday held race consciou admiss program harvard univers north carolina viol
 'thursday decis forc rework admiss criteria throughout american higher educ decad pursuit di
```

```
# create a dataframe of token tf-idf's with each row representing all articles of one sour

# convert the list of article strings into a tf-idf-value dataframe
t = Tokenizer()
t.fit_on_texts(source_docs)
print(t)
encoded_source_docs = t.texts_to_matrix(source_docs, mode='tfidf')
words = [x for x in t.word_index.keys()]
tfidf_source_df = pd.DataFrame(data = encoded_source_docs[:, 1:], columns=words)
# List of conditions
source_conditions = [
      tfidf_source_df['abcarticle'] != 0
    , tfidf_source_df['bbcarticle'] != 0
    , tfidf_source_df['cnnarticle'] != 0
    , tfidf_source_df['foxarticle'] != 0
    , tfidf_source_df['nbcarticle'] != 0
    , tfidf_source_df['nyparticle'] != 0
    , tfidf_source_df['nytarticle'] != 0
    , tfidf_source_df['wparticle'] != 0
    , tfidf_source_df['wsjarticle'] != 0
```

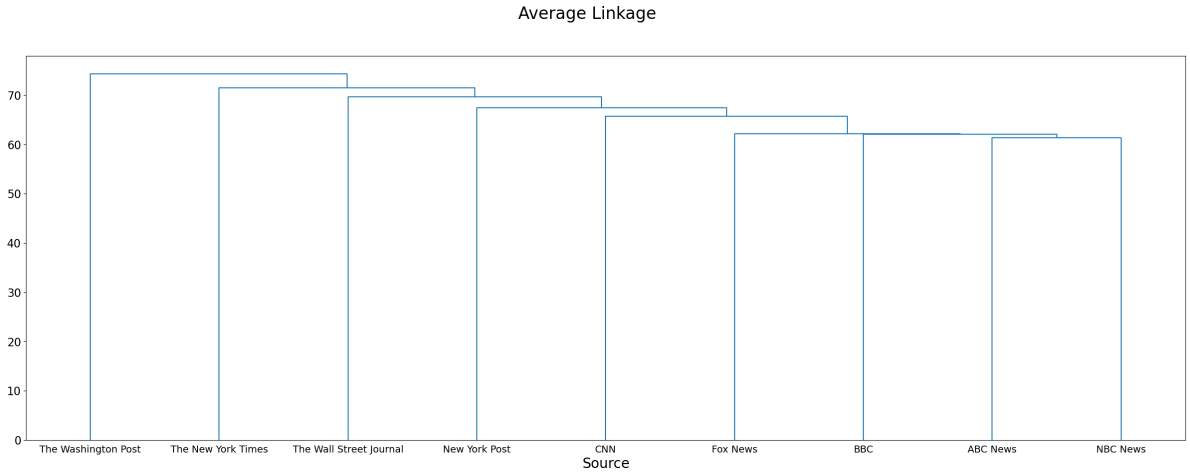
```
]

# List of values to return
source_choices  = [
      "ABC News"
    , "BBC"
    , "CNN"
    , "Fox News"
    , "NBC News"
    , "New York Post"
    , "The New York Times"
    , "The Washington Post"
    , "The Wall Street Journal"
]

# create a new source column
tfidf_source_df["article_source"] = np.select(source_conditions, source_choices, "ERROR")
tfidf_source_df.set_index('article_source', inplace=True)
tfidf_source_df.drop(['abcarticle', 'bbcarticle', 'cnnarticle', 'foxarticle',
                      'nbcarticle', 'nyparticle', 'nytarticle', 'wparticle',
                      'wsjarticle'], axis=1, inplace=True)
tfidf_source_df
```

```
<keras.src.preprocessing.text.Tokenizer object at 0x17634ee50>
```

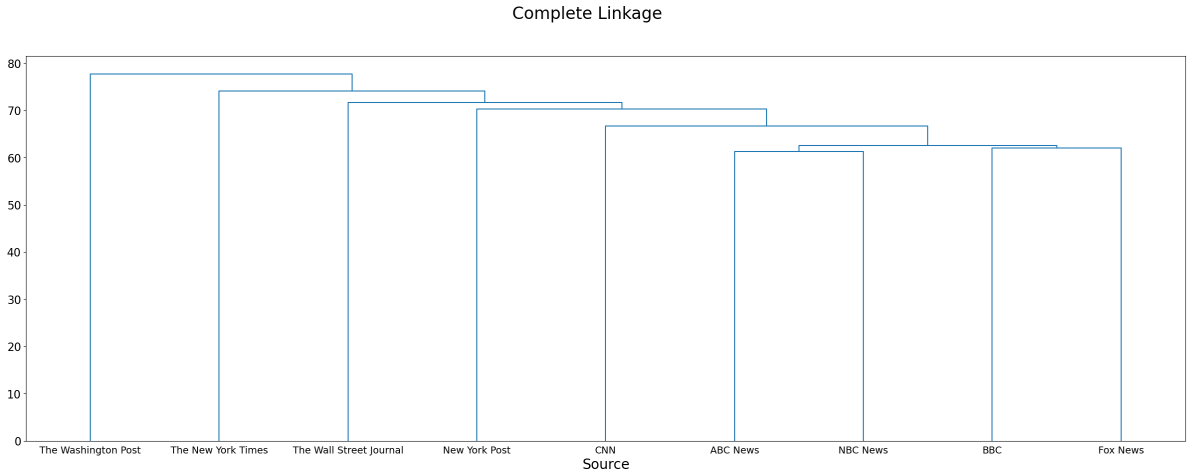
article_source	said	trump	biden	offici	mr	u	israel	presid
ABC News	3.591249	2.959536	3.099282	3.139834	1.173600	3.586612	2.824926	2.8861
BBC	3.280434	2.780642	2.335743	2.460363	3.283902	0.000000	2.460363	2.6543
CNN	3.698474	3.025425	2.976653	3.331000	0.693147	1.173600	3.225542	2.9933
Fox News	3.055995	3.350161	2.824926	2.421451	0.693147	3.027179	2.866350	2.9052
NBC News	3.670441	3.311249	2.866350	2.654383	1.654053	3.445144	3.099282	2.5317
New York Post	3.340652	3.126599	2.757300	2.335743	1.654053	2.924302	3.085175	3.0095
The New York Times	3.584733	3.269825	2.993325	2.654383	4.166255	1.935100	1.347002	2.8249
The Washington Post	3.785551	3.269825	3.269825	3.225542	0.000000	3.376459	2.976653	3.0992
The Wall Street Journal	3.849334	3.213976	2.866350	3.126599	3.919027	3.667067	2.905262	2.8031

```
from scipy.cluster.hierarchy import dendrogram, linkage
import matplotlib.pyplot as plt
```

```
Z = linkage(tfidf_source_df, 'average')
fig = plt.figure(figsize=(30, 10))
fig.suptitle("Average Linkage", fontsize=24)
plt.xlabel('Source', fontsize=20)
plt.yticks(fontsize = 16)
dn = dendrogram(Z, labels=tfidf_source_df.index)
plt.xticks(fontsize = 14)
plt.show()
```




```
from scipy.cluster.hierarchy import dendrogram, linkage
import matplotlib.pyplot as plt

Z = linkage(tfidf_source_df, 'complete')
fig = plt.figure(figsize=(30, 10))
fig.suptitle("Complete Linkage", fontsize=24)
plt.xlabel('Source', fontsize=20)
plt.yticks(fontsize = 16)
dn = dendrogram(Z, labels=tfidf_source_df.index)
plt.xticks(fontsize = 14)
plt.show()
```




```
def create_wordcloud(text, title):
    '''Given a string of all text and a string
    for the title, creates a wordcloud.'''
    plt.figure()
    wc = WordCloud().generate(text)
    plt.title(title)
    plt.axis("off")
    plt.imshow(wc)
```

```
for i in range(len(source_docs)):
    create_wordcloud(source_docs[i], tfidf_source_df.index[i])
```

ABC News


said
ukrain
trump
balloon
biden
u
democrat
appear
israel
document
republican
presid
tank
includ
abc news
offici said
militari
attack
statement
countri
american
russia
china
photo
report
peopl
inform
continu
support
appar
vote
use
new
take
work
isra
govern
hous
major
campaign
civilian
ukrainian
race
announc
say
time
set
one
defens
accord
year
two
justic
case
unit state
situat
state
elect
access
brief
santo
still
forc
potenti
protect
thursday
wednesday
depart
first
secur
view
number

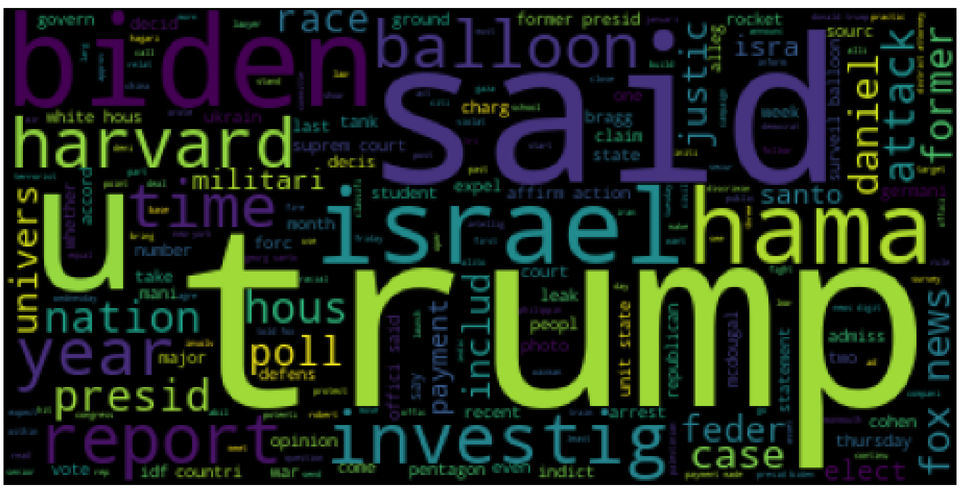
BBC


said
us
ukrain
elect
gaza
one
time
go
document
israel
includ
militari
isra
offici
decis
report
call
trump
mr trump
justic
tank
attack
year
made
case
republican
democrat
countri
say
campaign
balloon
hama
palestinian
first
vote
launch
ukrainian
court
alleg
presid

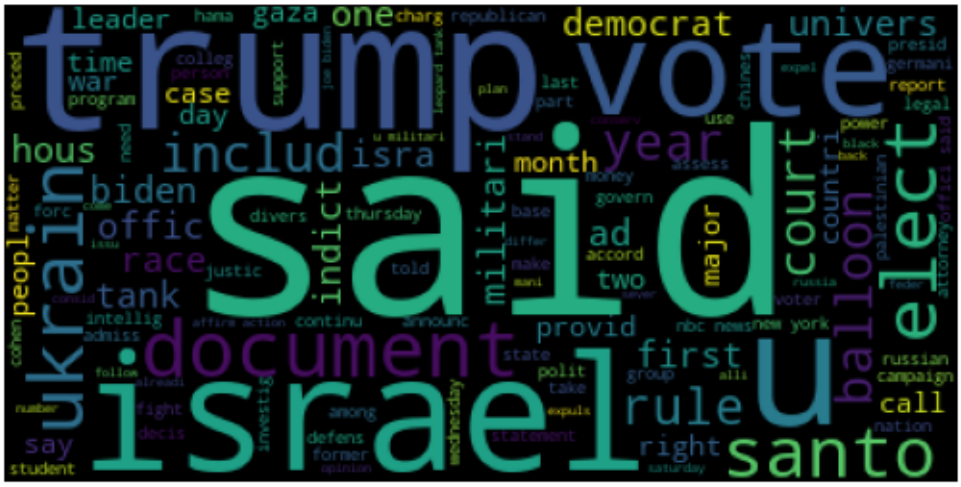
CNN


said
u
israel
ukrain
document
trump
includ
report
militari
intellig
court
gaza
isra
biden
offici said
hama
decis
chines
hous
countri
american
system
day
vote
peopl
inform
two
balloon
leak
accord
told cnn
democrat
call
respons
take
right
assess
major
investig
support
use
first
harvard
defens
govern
say
cnn
provid
indict
justic
senior
offic
teixeira
month
effort
make
come
santo
one
person
alli

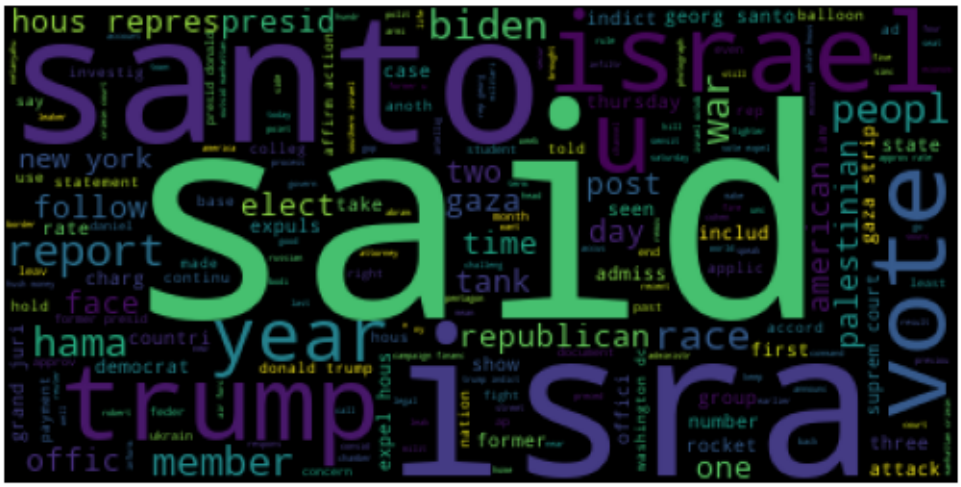
Fox News


trump
said
u
biden
israel
hama
balloon
harvard
investig
report
time
year
nation
hous
presid
poll
univers
militari
race
justic
daniel
attack
former
fox news
includ
payment
feder
case
santo
isra
alleg
week
bragg
claim
state
charg
supreme court
affirm action
student
expel
tank
last
decis
forc
month
number
take
mani
major
defens
opinion
court
leak
peopl
photo
unit state
republican
mcdougal
statement
arrest
cohen
thursday
recent
elect
vote
idf countri
pentagon
indict
white hous
ground
former presid
rocket
sourc
govern
admiss
two

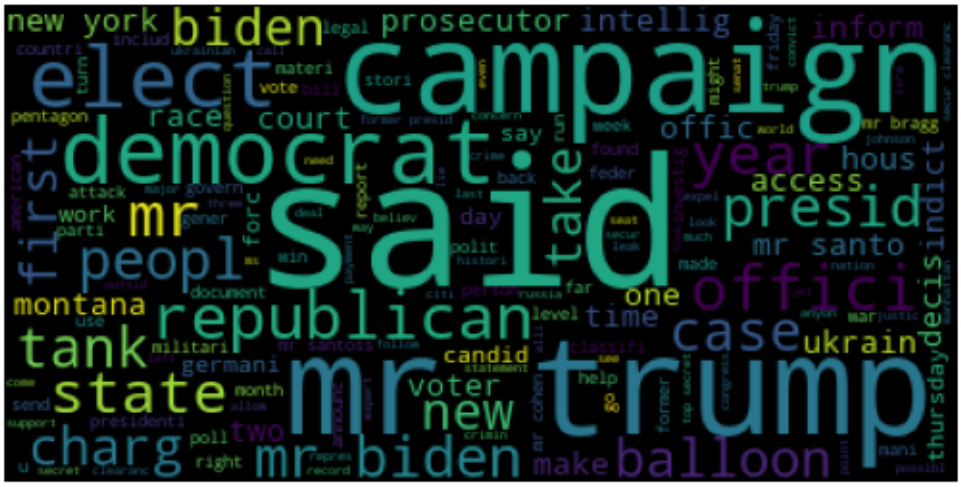
NBC News



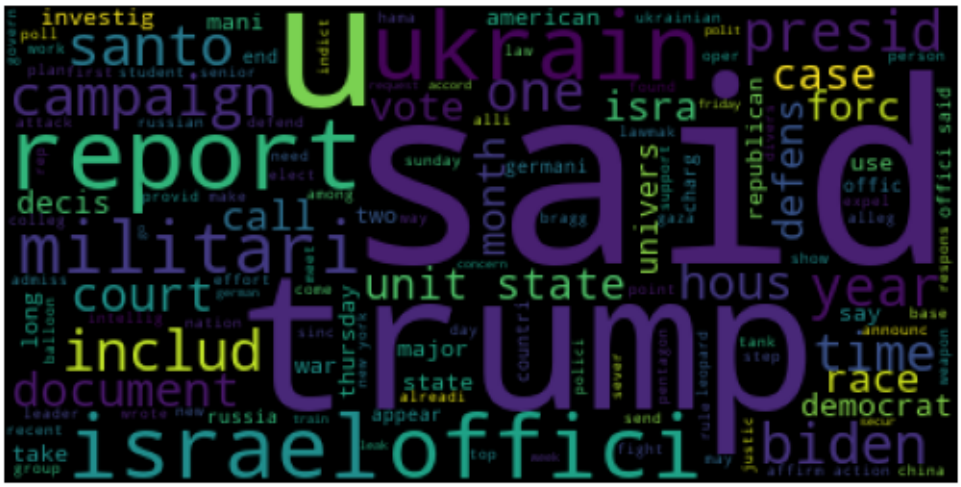
New York Post

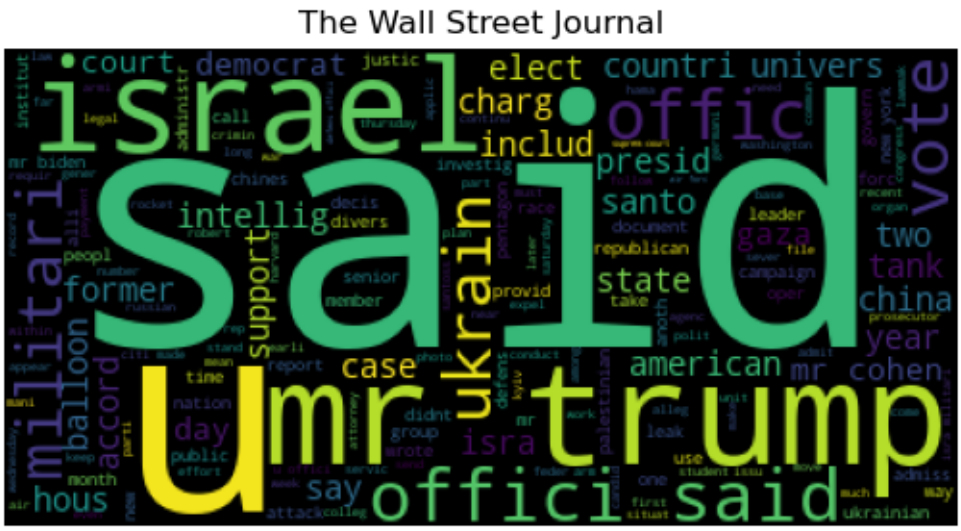


The New York Times



The Washington Post






```
# create a list of unstemmed article document texts
# looping through all text files to apply preprocessing functions
article_docs_unstemmed = []
dir = os.listdir('data/text/')
dir.sort()
for filename in dir:
    filepath = os.path.join('data/text/', filename)
    if filename.split(".")[-1] == "txt":
        article_string = file_to_string(filepath)
        new_string = clean_text(article_string, 0)
        article_docs_unstemmed.append(new_string)

article_docs_unstemmed
```

```
['supreme court thursday set new limits affirmative action programs cases involving whether p
 'massive spy balloon believed china seen montana tracked flies across continental united sta
 'two days dismal new polling numbers president joe biden showed public views unfavorably riv
 'hundreds people israel reported dead thousands injured hamas militants fired rockets gaza l
 'posting social media appears several highly classified u intelligence documents might begin
 'house representatives friday voted expel republican rep george santos historic move happene
 'major increase u support ukraine president joe biden signed sending abrams tanks war torn c
 'manhattan grand jury indicted former president donald trump making first current former pre
 'us supreme court ruled race longer considered factor university admissions landmark ruling
 'us tracking suspected chinese surveillance balloon spotted flying sensitive sites recent da
 'us president joe biden running election next year national opinion polls weak job approval
```

'least people reported killed wounded israel palestinian militant group hamas launched bigg
'documents include detailed accounts training provided ukraine foreign powers make dozens c
'us house representatives expelled congressman george santos following damning ethics repor
'us send powerful battle tanks ukraine joining germany sending vehicles support fight russi
'former us president donald trump charged hush money payments made porn star presidential e
'supreme court says colleges universities longer take race consideration specific basis gra
'us tracking suspected chinese high altitude surveillance balloon continental united states
'one third registered voters approve president joe bidens handling israeli palestinian conf
'gaza jerusalem cnn israels prime minister benjamin netanyahu declared country war saturday
'man arrested fbi connection massive us classified documents leak charged boston friday una
'house voted friday expel gop rep george santos historic vote makes new york congressman si
'leaders united states germany announced wednesday send contingents tanks ukraine reversing
'donald trump faces counts related business fraud indictment manhattan grand jury according
'u supreme court handed major ruling affirmative action thursday rejecting use race factor
'u government monitoring suspected chinese surveillance balloon moving northern states past
'biden battles rough poll numbers fox news white house correspondent peter doocy latest pre
'hamas widespread coordinated attack may suggest outside help trey yingst foreign correspon
'bret baier intel suspect year old government worker special report bret baier anchor quest
'house representatives voted expel scandal plagued rep george santos r n friday making firs
'german chancellor olaf scholz formally announced wednesday weeks stalling frustrating nego
'former president donald trump indicted part manhattan district attorney office years long
'washington supreme court thursday struck affirmative action programs university north caro
'u military monitoring suspected chinese surveillance balloon hovering northern u past days
'different polls agree president joe bidens political standing lower barack obamas point el
'ashkelon israel israel plunged chaos saturday palestinian militant group hamas launched de
'dozens leaked defense department classified documents posted online reveal details u spyin
'washington house voted overwhelmingly expel indicted rep george santos friday pulling curt
'ukraine set receive battle tanks germany western countries fierce debate exposed fissures
'grand jury new york city voted thursday indict donald trump first time former u president
'supreme court struck affirmative action programs harvard university university north caro
'us tracking chinese spy balloon floating northern part country days pentagon officials ann
'president biden ready ring new year seeing numbers year old commander chief ends lower app
'jerusalem ap hamas militants fired thousands rockets sent dozens fighters israeli towns ne
'massachusetts air national guardsman jack teixeira leader discord group dozens sensitive u
'bye george lying long island rep george santos r ny became sixth member ever expelled us h
'weeks excuses foot dragging germans finally said yes transferring limited number leopard t
'donald trump indicted manhattan grand jury thursday hush money payments made ahead electio
'chief justice john g roberts jr finished reading majority opinion supreme court chamber th
'helena mont larry mayer newspaper photographer pointed camera sky wednesday began snapping
'democrats battleground states growing increasingly anxious president bidens low approval r
'israels news sites compiling lists dead missing funerals taking place around country weeke
'washington would year old national guardsman position access top secret documents begin dr
'george santos new york republican congressman whose tapestry lies schemes made figure nati

```
'precision military drill first germany united states announced wednesday agreed provide bat
'manhattan grand jury indicted donald j trump thursday role paying hush money porn star acc
'supreme court thursday held race conscious admissions programs harvard university north ca
'chinese surveillance balloon collecting intelligence continental united states right u offi
'night president biden departed washington celebrate thanksgiving nantucket mass gathered c
'sderot israel israel formally declared war palestinian militant group hamas sunday reeled s
'saturday u officials foreign allies scrambled understand dozens classified intelligence doc
'house voted friday expel rep george santos r n congress action chamber previously taken fiv
'biden administration announced wednesday send premier battle tanks ukraine following agreem
'new york manhattan grand jury voted indict former president donald trump making first perso
'thursdays decision force reworking admissions criteria throughout american higher educatio
'washington u tracked officials described chinese reconnaissance balloon continental states
'mr bidens low standing head head general election polls reflects voters dour appraisal peri
'tel aviv israeli prime minister benjamin netanyahu said country war hamas militant groups i
'criminal case unfolding u government scrambles protect secrets unauthorized disclosures app
'lawmakers voted remove two thirds house supermajority required constitution almost democrat
'u germany outlined plans wednesday send dozens modern battle tanks ukraine marking signific
'grand jury returned indictment mr trump vote thursday kicking process former president expe
```

```
# create a list of strings where each string is all articles from one source (unstemmed)
source_docs_unstemmed = []

j = 0

for i in range(9):
    source = " ".join(article_docs_unstemmed[j].split()[:-2]) + " " + " ".join(article_doc
        + " ".join(article_docs_unstemmed[j+2].split()[:-2]) + " " + " ".join(article_docs
        + " ".join(article_docs_unstemmed[j+4].split()[:-2]) + " " + " ".join(article_docs
        + " ".join(article_docs_unstemmed[j+6].split()[:-2]) + " " + " ".join(article_docs
    source_docs_unstemmed.append(source)
    j += 8

source_docs_unstemmed
```

```
['supreme court thursday set new limits affirmative action programs cases involving whether p
 'us supreme court ruled race longer considered factor university admissions landmark ruling
 'supreme court says colleges universities longer take race consideration specific basis gran
 'u supreme court handed major ruling affirmative action thursday rejecting use race factor c
 'washington supreme court thursday struck affirmative action programs university north carol
 'supreme court struck affirmative action programs harvard university university north caroli
 'chief justice john g roberts jr finished reading majority opinion supreme court chamber thu
```

```
'supreme court thursday held race conscious admissions programs harvard university north ca
'thursdays decision force reworking admissions criteria throughout american higher educatio
```

```
# calculate polarity and subjectivity scores, create dataframe

scores_df = pd.DataFrame({'source':tfidf_df['article_source'], 'topic':tfidf_df['article_t

polarity_scores = []
subjectivity_scores = []

for article in article_docs_unstemmed:
    polarity_scores.append(round(TextBlob(article).sentiment.polarity, 2))
    subjectivity_scores.append(round(TextBlob(article).sentiment.subjectivity, 2))

scores_df['polarity_score'] = polarity_scores
scores_df['subjectivity_score'] = subjectivity_scores

average_topic_polarity_scores = []
average_source_polarity_scores = []

for topic in scores_df['topic'].value_counts().index:
    mean_score = round(scores_df[scores_df['topic'] == topic]['polarity_score'].mean(), 2)
    average_topic_polarity_scores.append(mean_score)

for source in scores_df['source'].value_counts().index:
    mean_score = round(scores_df[scores_df['source'] == source]['polarity_score'].mean(),
    for i in range(8):
        average_source_polarity_scores.append(mean_score)

scores_df['average_polarity_for_topic'] = average_topic_polarity_scores * 9
scores_df['average_polarity_for_source'] = average_source_polarity_scores

average_topic_subjectivity_scores = []
average_source_subjectivity_scores = []

for topic in scores_df['topic'].value_counts().index:
    mean_score = round(scores_df[scores_df['topic'] == topic]['subjectivity_score'].mean()
    average_topic_subjectivity_scores.append(mean_score)

for source in scores_df['source'].value_counts().index:
    mean_score = round(scores_df[scores_df['source'] == source]['subjectivity_score'].mean
```

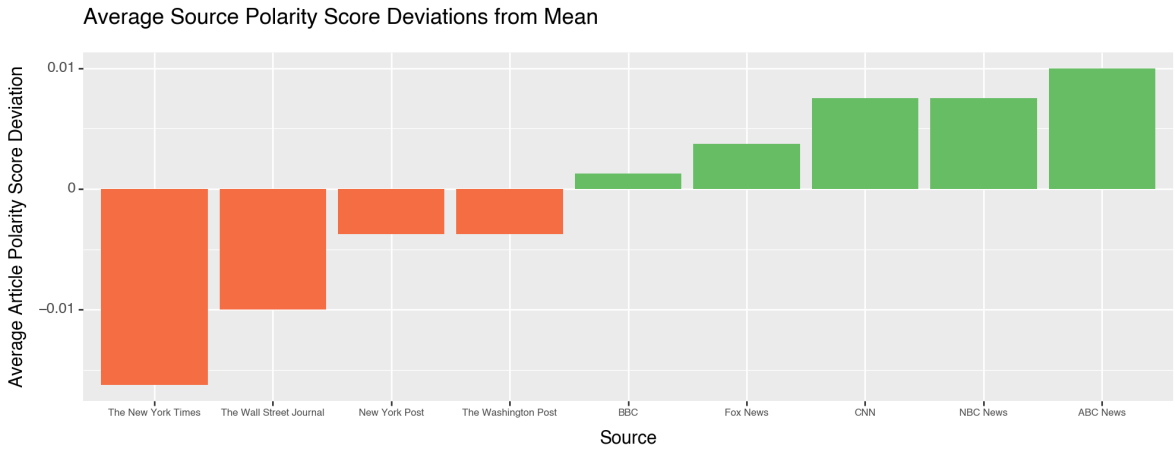
```
    for i in range(8):
        average_source_subjectivity_scores.append(mean_score)

scores_df['average_subjectivity_for_topic'] = average_topic_subjectivity_scores * 9
scores_df['average_subjectivity_for_source'] = average_source_subjectivity_scores

scores_df['polarity_diff_from_topic_mean'] = scores_df['polarity_score'] - scores_df['aver
scores_df['subjectivity_diff_from_topic_mean'] = scores_df['subjectivity_score'] - scores_
scores_df['polarity_diff_from_source_mean'] = scores_df['polarity_score'] - scores_df['ave
scores_df['subjectivity_diff_from_source_mean'] = scores_df['subjectivity_score'] - scores

sources_polarity_dev = []

for source in scores_df['source'].value_counts().index:
    total_dev = scores_df[scores_df['source'] == source]['polarity_diff_from_topic_mean'].
    for i in range(8):
        sources_polarity_dev.append(total_dev/8)

sources_subjectivity_dev = []

for source in scores_df['source'].value_counts().index:
    total_dev = scores_df[scores_df['source'] == source]['subjectivity_diff_from_topic_mea
    for i in range(8):
        sources_subjectivity_dev.append(total_dev/8)

scores_df['average_source_polarity_deviation_from_topic_mean'] = sources_polarity_dev
scores_df['average_source_subjectivity_deviation_from_topic_mean'] = sources_subjectivity_

scores_df.iloc[64:, :]
```

	source	topic	polarity_score	subjectivity_s
64	The Wall Street Journal	Supreme Court Ruling on Affirmative Action	0.07	0.39
65	The Wall Street Journal	Chinese Surveillance Balloon	0.02	0.28
66	The Wall Street Journal	Biden's Low Approval Rates in Polls	0.07	0.42
67	The Wall Street Journal	The Deadliest Attack by Hamas	0.04	0.32
68	The Wall Street Journal	Pentagon Documents Leak	0.05	0.39
69	The Wall Street Journal	George Santos' Expulsion from Congress	-0.04	0.35
70	The Wall Street Journal	U.S. and Germany Send Tanks to Ukraine	0.11	0.32
71	The Wall Street Journal	Trump's Indictment	0.03	0.42

```
# create a df for each source and its polarity score total deviations from the topic means

polarity_devs = pd.DataFrame({'source':scores_df['source'].value_counts().index,
                              'polarity_dev':scores_df[scores_df['topic'] == 'Chinese Surv
                                ['average_source_polarity_deviation_from_topic_mean']})
polarity_devs['sign'] = np.where(polarity_devs['polarity_dev'] > 0, 'pos', 'neg')
polarity_devs.sort_values(by='polarity_dev', ascending=True, inplace=True)
polarity_devs.reset_index(drop=True, inplace=True)

source_order = CategoricalDtype(
    ["The New York Times", "The Wall Street Journal", "New York Post",
     "The Washington Post", "BBC", "Fox News", "CNN", "NBC News", "ABC News"],
    ordered=False
)
polarity_devs['source'] = polarity_devs['source'].astype(source_order)

polarity_devs
```

	source	polarity_dev	sign
0	The New York Times	-0.01625	neg
1	The Wall Street Journal	-0.01000	neg
2	New York Post	-0.00375	neg
3	The Washington Post	-0.00375	neg
4	BBC	0.00125	pos
5	Fox News	0.00375	pos
6	CNN	0.00750	pos
7	NBC News	0.00750	pos
8	ABC News	0.01000	pos

```
colors = {"pos":'#66bd63', "neg":'#f46d43'}

(
ggplot(polarity_devs, aes(x="source", y="polarity_dev", fill="sign"))
+ geom_col(stat="identity")
+ labs(x='Source', y='Average Article Polarity Score Deviation', title='Average Source Pol
+ theme(figure_size = (10, 4), axis_text_x=element_text(size=6, face='bold'), legend_posit
+ scale_fill_manual(values = colors)
)
```



<Figure Size: (1000 x 400)>