```
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
```

```
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]=="B"}
#print("key by value:",value
```

```
def clean_text(text):
    '''Takes in a string of text and cleans it by converting
    to lowercase, removing punctuation, removing stopwords,
    and stemming. Returns the new string.'''
    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(" ")
    stemmed = []
    for token in tokens[:-2]:
        # only include new token in the cleaned list if not a stopword
        if token not in stopwords_list:
            stemmed_word = ps.stem(token)
            stemmed.append(stemmed_word)
            if token not in stemmed_dict:
```

```
                stemmed_dict[token] = stemmed_word
    stemmed.append(tokens[-2])
    stemmed.append(tokens[-1])
    cleaned_text = " ".join(stemmed)
    cleaned_text = cleaned_text.replace('  ', ' ')
    return cleaned_text
```

```
# looping through all text files to apply preprocessing functions
file_list = []
article_docs = []
dir = os.listdir('data/text/')
dir.sort()
for filename in dir:
    filepath = os.path.join('data/text/', filename)
    file_list.append(f"{filepath}")
    if filename.split(".")[-1] == "txt":
        article_string = file_to_string(filepath)
        new_string = clean_text(article_string)
        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 0x1609bcf10>
```

	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	..

```
article_docs
```

```
['suprem court thursday set new limit affirm action program case involv whether public privat
 'massiv spi balloon believ china seen montana track fli across continent unit state presid
 'two day dismal new poll number presid joe biden show public view unfavor rival donald trump
 'hundr peopl israel report dead thousand injur hama milit fire rocket gaza launch ground inc
 'post social media appear sever highli classifi u intellig document might begin could turn s
 'hous repres friday vote expel republican rep georg santo histor move happen year santo scan
 'major increas u support ukrain presid joe biden sign send abram tank war torn countri conce
 'manhattan grand juri indict former presid donald trump make first current former presid fac
 'us suprem court rule race longer consid factor univers admiss landmark rule upend decad old
 'us track suspect chines surveil balloon spot fli sensit site recent day defenc offici said
 'us presid joe biden run elect next year nation opinion poll weak job approv rate suggest vo
 'least  peopl report kill  wound israel palestinian milit group hama launch biggest attack y
 'document includ detail account train provid ukrain foreign power make dozen classifi us def
```

'us hous repres expel congressman georg santo follow damn ethic report dozen crimin charg h
'us send power battl tank ukrain join germani send vehicl support fight russia invas decis
'former us presid donald trump charg hush money payment made porn star presidenti elect de
'suprem court say colleg univers longer take race consider specif basi grant admiss landmar
'us track suspect chines high altitud surveil balloon continent unit state defens offici sa
'one third regist voter approv presid joe biden handl isra palestinian conflict new poll ne
'gaza jerusalem cnn israel prime minist benjamin netanyahu declar countri war saturday pale
'man arrest fbi connect massiv us classifi document leak charg boston friday unauthor reten
'hous vote friday expel gop rep georg santo histor vote make new york congressman sixth law
'leader unit state germani announc wednesday send conting tank ukrain revers longstand trep
'donald trump face count relat busi fraud indict manhattan grand juri accord two sourc fami
'u suprem court hand major rule affirm action thursday reject use race factor colleg admiss
'u govern monitor suspect chines surveil balloon move northern state past sever day pentago
'biden battl rough poll number fox news white hous correspond peter dooci latest presid ele
'hama widespread coordin attack may suggest outsid help trey yingst foreign correspond trey
'bret baier intel suspect year old govern worker special report bret baier anchor question
'hous repres vote expel scandal plagu rep georg santo r n friday make first hous lawmak exp
'german chancellor olaf scholz formal announc wednesday week stall frustrat negoti berlin a
'former presid donald trump indict part manhattan district attorney offic year long investi
'washington suprem court thursday struck affirm action program univers north carolina harva
'u militari monitor suspect chines surveil balloon hover northern u past day militari defen
'differ poll agre presid joe biden polit stand lower barack obama point elect even lower do
'ashkelon israel israel plung chao saturday palestinian milit group hama launch deadli land
'dozen leak defens depart classifi document post onlin reveal detail u spi russia war machi
'washington hous vote overwhelmingli expel indict rep georg santo friday pull curtain tempe
'ukrain set receiv battl tank germani western countri fierc debat expos fissur among alli a
'grand juri new york citi vote thursday indict donald trump first time former u presid face
'suprem court struck affirm action program harvard univers univers north carolina thursday
'us track chines spi balloon float northern part countri day pentagon offici announc thursd
'presid biden readi ring new year see number year old command chief end lower approv rate
'jerusalem ap hama milit fire thousand rocket sent dozen fighter isra town near gaza strip
'massachusett air nation guardsman jack teixeira leader discord group dozen sensit us intel
'bye georg lie long island rep georg santo r ny becam sixth member ever expel us hous repre
'week excus foot drag german final said ye transfer limit number leopard tank ukrain agre w
'donald trump indict manhattan grand juri thursday hush money payment made ahead elect mar
'chief justic john g robert jr finish read major opinion suprem court chamber thursday hush
'helena mont larri mayer newspap photograph point camera sky wednesday began snap pictur ap
'democrat battleground state grow increasingli anxiou presid biden low approv rate worri vo
'israel news site compil list dead miss funer take place around countri weekend attack conf
'washington would year old nation guardsman posit access top secret document begin dramat a
'georg santo new york republican congressman whose tapestri lie scheme made figur nation ri
'precis militari drill first germani unit state announc wednesday agre provid battl tank he
'manhattan grand juri indict donald j trump thursday role pay hush money porn star accord p

```
 'suprem court thursday held race consciou admiss program harvard univers north carolina viol
 'chines surveil balloon collect intellig continent unit state right u offici disclos thursda
 'night presid biden depart washington celebr thanksgiv nantucket mass gather closest aid mee
 'sderot israel israel formal declar war palestinian milit group hama sunday reel surpris att
 'saturday u offici foreign alli scrambl understand dozen classifi intellig document end inte
 'hous vote friday expel rep georg santo r n congress action chamber previous taken five time
 'biden administr announc wednesday send premier battl tank ukrain follow agreement germani c
 'new york manhattan grand juri vote indict former presid donald trump make first person u hi
 'thursday  decis forc rework admiss criteria throughout american higher educ decad pursuit c
 'washington u track offici describ chines reconnaiss balloon continent state week would aggr
 'mr biden low stand head head gener elect poll reflect voter dour apprais perform presid sur
 'tel aviv isra prime minist benjamin netanyahu said countri war hama milit group forc pour a
 'crimin case unfold u govern scrambl protect secret unauthor disclosur appear provid detail
 'lawmak vote   remov two third hous supermajor requir constitut almost democrat mani republi
 'u germani outlin plan wednesday send dozen modern battl tank ukrain mark signific new infus
 'grand juri return indict mr trump vote thursday kick process former presid expect come new
```

```
# 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
```

```
# adjust the following code to create a dataframe with each row being all articles of one

# convert the list of article strings into a binary-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)
```

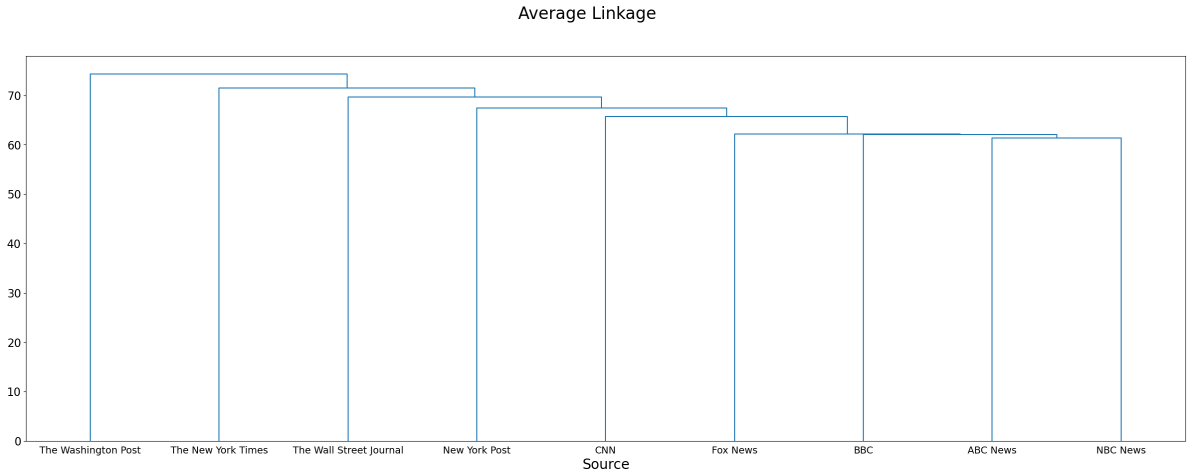
```
tfidf_source_df
```

```
<keras.src.preprocessing.text.Tokenizer object at 0x1710e0ca0>
```

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

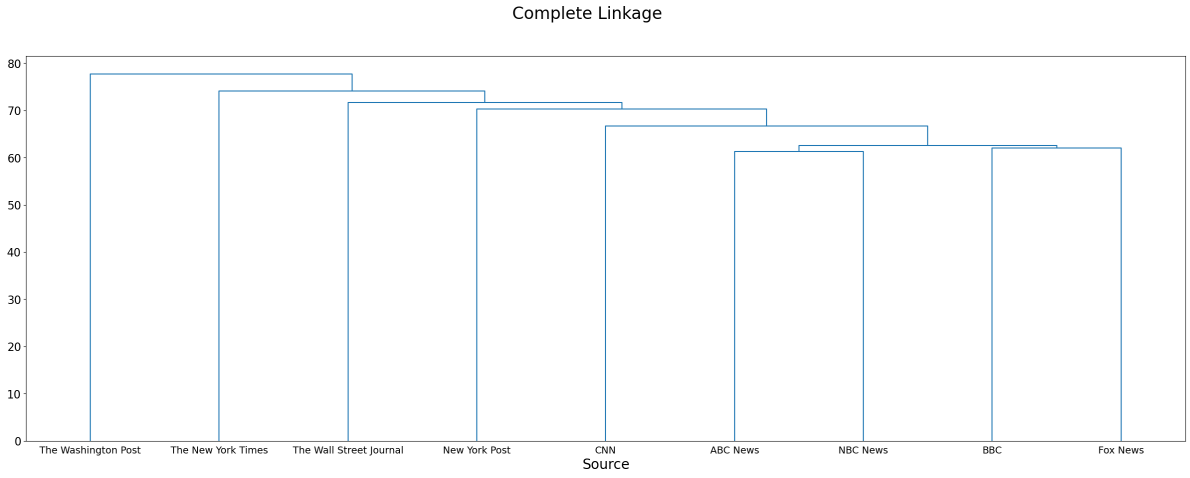
```
from scipy.cluster.hierarchy import dendrogram, linkage
import matplotlib.pyplot as plt

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()
```



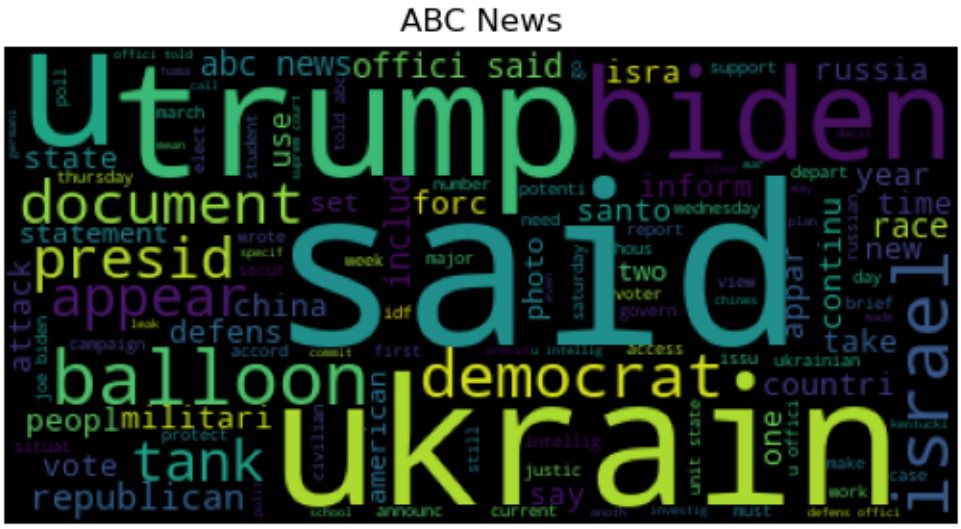

```
from scipy.cluster.hierarchy import dendrogram, linkage
import matplotlib.pyplot as plt

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()
```



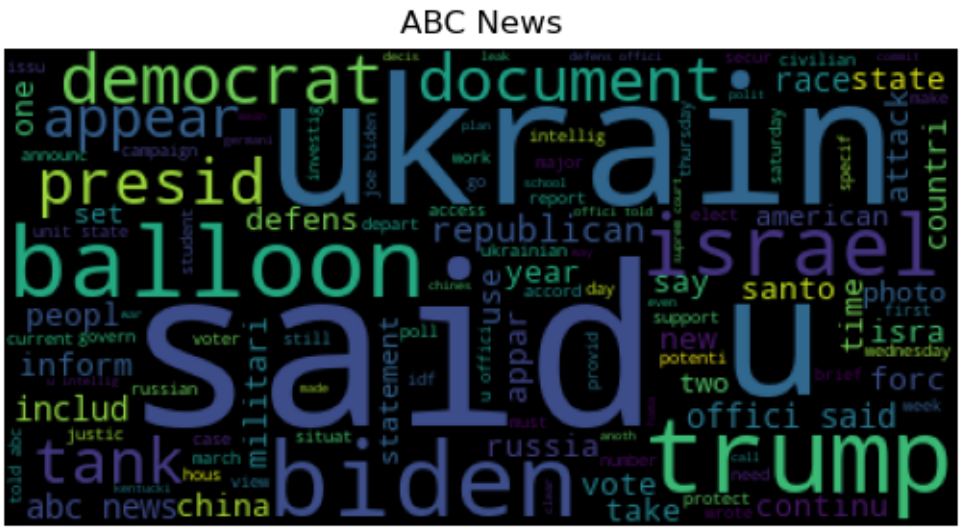
```
def create_wordcloud(text, title):
    '''Given a string of all text and a string
    for the title, creates a wordcloud'''
    wc = WordCloud().generate(text)
    plt.title(title)
    plt.axis("off")
    plt.imshow(wc)
```

```
create_wordcloud(source_docs[0], "ABC News")
```




```
wc = WordCloud().generate(source_docs[0])
plt.title("ABC News")
plt.axis("off")
plt.imshow(wc)
```

```
<matplotlib.image.AxesImage at 0x1734a59a0>
```




```
wc = WordCloud().generate(source_docs[1])
plt.title("ABC News")
plt.axis("off")
plt.imshow(wc)
```