Import required libraries

In [1]:
```
!pip install tabulate
!pip install wordcloud
!pip install gensim
!pip install --user --upgrade scikit-learn == 1.1.2

%pylab inline

import pickle as pk
from scipy import sparse as sp
import warnings
from wordcloud import WordCloud
import itertools
import collections
from tabulate import tabulate
import pandas as pd
import nltk
from nltk.tokenize import TweetTokenizer
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
from nltk.corpus import wordnet
import matplotlib.pyplot as plt
import seaborn as sns
import re
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation as LDA
import numpy as np
from nltk.stem.wordnet import WordNetLemmatizer
from nltk.tokenize import RegexpTokenizer
from gensim.models import Phrases
from gensim.models import LdaModel
import pyLDAvis
import pyLDAvis.gensim_models as gensimvis
import warnings
import sklearn
from gensim.corpora import Dictionary
from time import time
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.decomposition import NMF, MiniBatchNMF, LatentDirichletAllocation
from sklearn.datasets import fetch_20newsgroups
```

```
from sklearn.feature_extraction.text import CountVectorizer


pyLDAvis.enable_notebook()

warnings.filterwarnings("ignore", category=DeprecationWarning)
warnings.simplefilter("ignore", DeprecationWarning)

print('The scikit-learn version is {}.'.format(sklearn.__version__))
```

```
Requirement already satisfied: tabulate in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages (0.
8.9)
Requirement already satisfied: wordcloud in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages (1.
8.2.2)
Requirement already satisfied: matplotlib in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages (f
rom wordcloud) (3.5.1)
Requirement already satisfied: pillow in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages (from
wordcloud) (9.0.1)
Requirement already satisfied: numpy>=1.6.1 in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages
(from wordcloud) (1.21.5)
Requirement already satisfied: cycler>=0.10 in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages
(from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-pack
ages (from matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: python-dateutil>=2.7 in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-p
ackages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: packaging>=20.0 in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-packag
es (from matplotlib->wordcloud) (21.3)
Requirement already satisfied: pyparsing>=2.2.1 in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-packa
ges (from matplotlib->wordcloud) (3.0.4)
Requirement already satisfied: kiwisolver>=1.0.1 in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-pack
ages (from matplotlib->wordcloud) (1.3.2)
Requirement already satisfied: six>=1.5 in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages (fro
m python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
Requirement already satisfied: gensim in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages (4.1.
2)
Requirement already satisfied: numpy>=1.17.0 in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages
(from gensim) (1.21.5)
Requirement already satisfied: scipy>=0.18.1 in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages
(from gensim) (1.7.3)
Requirement already satisfied: smart-open>=1.8.1 in /Users/haileythanki/opt/anaconda3/lib/python3.9/site-pack
ages (from gensim) (5.2.1)
zsh:1: = not found
%pylab is deprecated, use %matplotlib inline and import the required libraries.
Populating the interactive namespace from numpy and matplotlib
The scikit-learn version is 1.1.2.
```

Import data

In [2]:
```
df_raw = pd.read_csv("training_data.csv", low_memory = False)
df_raw
```

Out[2]:

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country
0	Remember the #WuhanCoronaVirus? The pandemic w...	NaN	FALSE	TRUE	WuhanCoronaVirus KillerCuomo	us
1	My sources @WhiteHouse say 2 tactics will be u...	NaN	FALSE	TRUE	Trump	us
2	I'll venture a wild guess: If you were running...	NaN	FALSE	TRUE	COVID19	us
3	#Pakistan (#GreenStimulus = #Nature protection...	NaN	FALSE	TRUE	Pakistan GreenStimulus Nature Green	us
4	🇺🇸 Pandémie de #coronavirus: 30 pasteurs améri...	NaN	FALSE	TRUE	coronavirus COVID__19 COVID—19	us
...	...	...	...	...	...	...
239995	Aa Likes, Retweets yentra 🙏\n🔥🔥🔥 \n#Master	NaN	TRUE	TRUE	Master	new_zealand
239996	Very interesting\nAny thoughts? \n\n#TheFive #T...	NaN	FALSE	TRUE	TheFive Trump2020 KAG2020 mondaythoughts COVID...	new_zealand
239997	As we deal with #COVID19 don't forget that #Ch...	NaN	TRUE	TRUE	COVID19 Christians persecution Nigeria	new_zealand
239998	While we hit 150,000 in #COVID19 deaths, the P...	NaN	FALSE	TRUE	COVID19	new_zealand
239999	This too shall pass #Covid_19 . May remain sta...	NaN	FALSE	TRUE	Covid_19 HopeAlive	new_zealand

240000 rows × 6 columns

Descriptive Analysis on Raw Data

Analyzing character and word counts for tweets

In [3]:
```
# Calculating character and word count for each tweet
```

```
df_raw['text_char_count'] = df_raw['text'].astype(str).apply(len)
df_raw['text_word_count'] = df_raw['text'].apply(lambda x: len(str(x).split()))
df_raw
```

Out[3]:

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_wo
0	Remember the #WuhanCoronaVirus? The pandemic w...	NaN	FALSE	TRUE	WuhanCoronaVirus KillerCuomo	us	267	
1	My sources @WhiteHouse say 2 tactics will be u...	NaN	FALSE	TRUE	Trump	us	281	
2	I'll venture a wild guess: If you were running...	NaN	FALSE	TRUE	COVID19	us	292	
3	#Pakistan (#GreenStimulus = #Nature protection...	NaN	FALSE	TRUE	Pakistan GreenStimulus Nature Green	us	236	
4	🇺🇸 Pandémie de #coronavirus: 30 pasteurs améri...	NaN	FALSE	TRUE	coronavirus COVID__19 COVID —19	us	279	
...	...	...	...	...	...	...	...	...
239995	Aa Likes, Retweets yentra 🙏\n🔥🔥🔥 \n#Master	NaN	TRUE	TRUE	Master	new_zealand	39	
239996	Very interesting\nAny thoughts? \n\n#TheFive #T...	NaN	FALSE	TRUE	TheFive Trump2020 KAG2020 mondaythoughts COVID...	new_zealand	142	
239997	As we deal with #COVID19 don't forget that #Ch...	NaN	TRUE	TRUE	COVID19 Christians persecution Nigeria	new_zealand	307	
239998	While we hit 150,000 in #COVID19 deaths, the P...	NaN	FALSE	TRUE	COVID19	new_zealand	115	
239999	This too shall pass #Covid_19 . May remain sta...	NaN	FALSE	TRUE	Covid_19 HopeAlive	new_zealand	280	

240000 rows × 8 columns

In [4]:
```
# Calculating the mean, median, maximum and minimum of word count and character count of tweets

table = [['Attribute', 'Aggregation Type', 'Value'],
         ['Character Count', 'mean', round((df_raw.text_char_count.mean()),2)],
         ['', 'median', round((df_raw.text_char_count.median()),2)],
         ['', 'max', round((df_raw.text_char_count.max()),2)],
         ['', 'min', round((df_raw.text_char_count.min()),2)],
         ['Word Count', 'mean', round((df_raw.text_word_count.mean()),2)],
         ['', 'median', round((df_raw.text_word_count.median()),2)],
         ['', 'max', round((df_raw.text_word_count.max()),2)],
         ['', 'min', round((df_raw.text_word_count.min()),2)]]
print(tabulate(table, headers='firstrow'))
```

```
Attribute        Aggregation Type      Value
---------------  ------------------  -------
Character Count  mean                204.98
                 median              221
                 max                 425
                 min                   1
Word Count       mean                 28.83
                 median               30
                 max                  82
                 min                   1
```

In [5]:
```
# Calculating character and word count for each tweet's hashtags

df_raw['hashtags_char_count'] = df_raw['hashtags'].astype(str).apply(len)
df_raw['hashtags_word_count'] = df_raw['hashtags'].apply(lambda x: len(str(x).split()))
df_raw
```

Out[5]:

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_wo
0	Remember the #WuhanCoronaVirus? The pandemic w...	NaN	FALSE	TRUE	WuhanCoronaVirus KillerCuomo	us	267	
1	My sources @WhiteHouse say 2 tactics will be u...	NaN	FALSE	TRUE	Trump	us	281	
2	I'll venture a wild guess: If you were running...	NaN	FALSE	TRUE	COVID19	us	292	
3	#Pakistan (#GreenStimulus = #Nature protection...	NaN	FALSE	TRUE	Pakistan GreenStimulus Nature Green	us	236	
4	🇺🇸 Pandémie de #coronavirus: 30 pasteurs améri...	NaN	FALSE	TRUE	coronavirus COVID__19 COVID —19	us	279	
...	...	...	...	...	...	...	...	...
239995	Aa Likes, Retweets yentra 🙏\n🔥🔥🔥 \n#Master	NaN	TRUE	TRUE	Master	new_zealand	39	
239996	Very interesting\nAny thoughts? \n\n#TheFive #T...	NaN	FALSE	TRUE	TheFive Trump2020 KAG2020 mondaythoughts COVID...	new_zealand	142	
239997	As we deal with #COVID19 don't forget that #Ch...	NaN	TRUE	TRUE	COVID19 Christians persecution Nigeria	new_zealand	307	
239998	While we hit 150,000 in #COVID19 deaths, the P...	NaN	FALSE	TRUE	COVID19	new_zealand	115	
239999	This too shall pass #Covid_19 . May remain sta...	NaN	FALSE	TRUE	Covid_19 HopeAlive	new_zealand	280	

240000 rows × 10 columns

In [6]:
```
# Calculating the mean, median, maximum and minimum of word count and character count of hashtags

table = [['Attribute', 'Aggregation Type', 'Value'],
         ['Character Count', 'mean', round((df_raw.hashtags_char_count.mean()),2)],
         ['', 'median', round((df_raw.hashtags_char_count.median()),2)],
         ['', 'max', round((df_raw.hashtags_char_count.max()),2)],
         ['', 'min', round((df_raw.hashtags_char_count.min()),2)],
         ['Word Count', 'mean', round((df_raw.hashtags_word_count.mean()),2)],
         ['', 'median', round((df_raw.hashtags_word_count.median()),2)],
         ['', 'max', round((df_raw.hashtags_word_count.max()),2)],
         ['', 'min', round((df_raw.hashtags_word_count.min()),2)]]
print(tabulate(table, headers='firstrow'))
```

```
Attribute        Aggregation Type      Value
---------------  ------------------  -------
Character Count  mean                  16.5
                 median                11
                 max                  124
                 min                    1
Word Count       mean                   1.81
                 median                 1
                 max                   20
                 min                    1
```

Analyzing frequency of words in tweets

In [7]:
```
# Splitting each tweet into individual words

words_in_tweet = [tweet.lower().split() for tweet in df_raw.hashtags]

# List of all words across tweets

all_words = list(itertools.chain(*words_in_tweet))

# Create a word frequency counter

word_freq = collections.Counter(all_words)
top_10_words_freq = word_freq.most_common(10)
top_10_words_freq
```

Out[7]:
```
[('covid19', 115377),
 ('coronavirus', 33708),
 ('covid', 14892),
 ('covid_19', 7617),
 ('covid—19', 5628),
 ('staysafe', 4699),
 ('stayhome', 3670),
 ('pandemic', 3487),
 ('covidiots', 2584),
 ('lockdown', 2294)]
```

In [8]:
```
top_10_words = []

for i in range(0,10):
    word = top_10_words_freq[i][0]
    top_10_words.append(word)

top_10_words
```

Out[8]:
```
['covid19',
 'coronavirus',
 'covid',
 'covid_19',
 'covid—19',
 'staysafe',
 'stayhome',
 'pandemic',
 'covidiots',
 'lockdown']
```

In [9]:
```
countries = df_raw.country.unique()
countries
```

Out[9]:
```
array(['us', 'uk', 'canada', 'australia', 'ireland', 'new_zealand'],
      dtype=object)
```

In [10]:
```
df_top_10_words_by_country = pd.DataFrame(columns=['country', 'word','freq'])

for country in countries:
    df_raw_country = df_raw[df_raw.country == country]
    words_in_tweet = [tweet.lower().split() for tweet in df_raw_country.hashtags]
    all_words = list(itertools.chain(*words_in_tweet))
    for word in top_10_words:
        freq = all_words.count(word)
        df_top_10_words_curr_country = pd.DataFrame([[country, word, freq]], columns = ['country', 'word','fre
        df_top_10_words_by_country = pd.concat([df_top_10_words_by_country, df_top_10_words_curr_country])
```

```
df_top_10_words_by_country.reset_index().drop(columns='index')
```

Out[10]:

	country	word	freq
0	us	covid19	19293
1	us	coronavirus	6622
2	us	covid	2719
3	us	covid_19	983
4	us	covid—19	897
5	us	staysafe	329
6	us	stayhome	583
7	us	pandemic	683
8	us	covidiots	461
9	us	lockdown	140
10	uk	covid19	17240
11	uk	coronavirus	6922
12	uk	covid	2407
13	uk	covid_19	1284
14	uk	covid—19	957
15	uk	staysafe	977
16	uk	stayhome	675
17	uk	pandemic	590
18	uk	covidiots	401
19	uk	lockdown	592
20	canada	covid19	19512
21	canada	coronavirus	5517
22	canada	covid	2269
23	canada	covid_19	1229
24	canada	covid—19	904
25	canada	staysafe	686

	country	word	freq
26	canada	stayhome	537
27	canada	pandemic	627
28	canada	covidiots	448
29	canada	lockdown	453
30	australia	covid19	18833
31	australia	coronavirus	5249
32	australia	covid	2735
33	australia	covid_19	1278
34	australia	covid—19	825
35	australia	staysafe	528
36	australia	stayhome	680
37	australia	pandemic	618
38	australia	covidiots	532
39	australia	lockdown	241
40	ireland	covid19	20004
41	ireland	coronavirus	4642
42	ireland	covid	2000
43	ireland	covid_19	1528
44	ireland	covid—19	962
45	ireland	staysafe	1650
46	ireland	stayhome	566
47	ireland	pandemic	334
48	ireland	covidiots	357
49	ireland	lockdown	548
50	new_zealand	covid19	20495
51	new_zealand	coronavirus	4756

	country	word	freq
52	new_zealand	covid	2762
53	new_zealand	covid_19	1315
54	new_zealand	covid—19	1083
55	new_zealand	staysafe	529
56	new_zealand	stayhome	629
57	new_zealand	pandemic	635
58	new_zealand	covidiots	385
59	new_zealand	lockdown	320

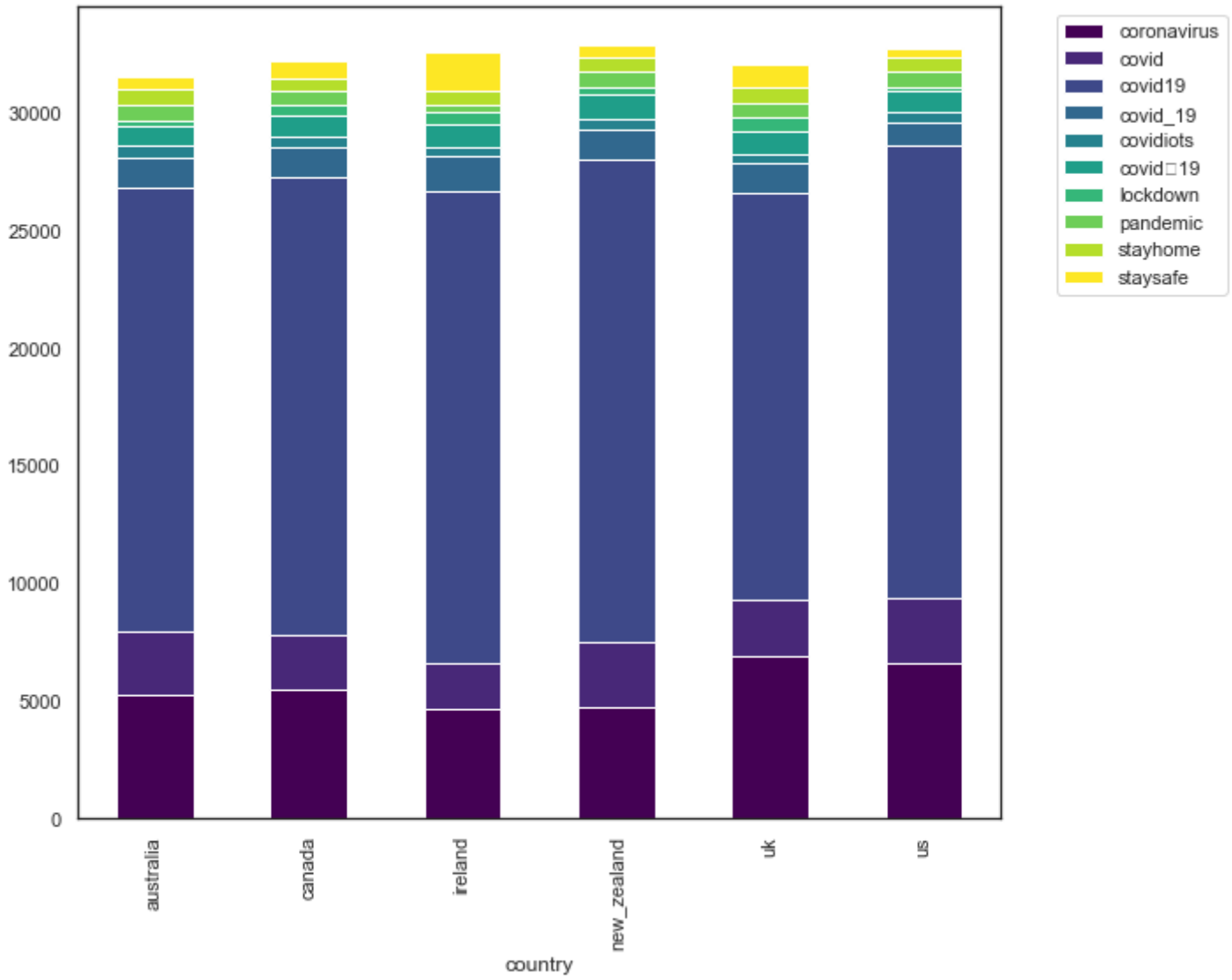
In [11]:
```
df_top_10_words_by_country = df_top_10_words_by_country.groupby(['country', 'word'])['freq'].sum().unstack().
df_top_10_words_by_country
```

Out[11]:

word	coronavirus	covid	covid19	covid_19	covidiots	covid—19	lockdown	pandemic	stayhome	staysafe
country										
australia	5249	2735	18833	1278	532	825	241	618	680	528
canada	5517	2269	19512	1229	448	904	453	627	537	686
ireland	4642	2000	20004	1528	357	962	548	334	566	1650
new_zealand	4756	2762	20495	1315	385	1083	320	635	629	529
uk	6922	2407	17240	1284	401	957	592	590	675	977
us	6622	2719	19293	983	461	897	140	683	583	329

In [12]:
```
plt.rcParams["figure.figsize"] = (10,8)

#set seaborn plotting aesthetics
sns.set(style='white')

#create stacked bar chart
df_top_10_words_by_country.plot(kind='bar', stacked=True, cmap="viridis")
plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')
plt.tight_layout()

plt.show()
```

```
/var/folders/_y/ch74wgzn7s1dxtq4ysb993sr0000gn/T/ipykernel_61326/164951854.py:9: UserWarning: Glyph 12540 (\N
{KATAKANA-HIRAGANA PROLONGED SOUND MARK}) missing from current font.
  plt.tight_layout()
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/IPython/core/pylabtools.py:151: UserWarning: Gl
yph 12540 (\N{KATAKANA-HIRAGANA PROLONGED SOUND MARK}) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```



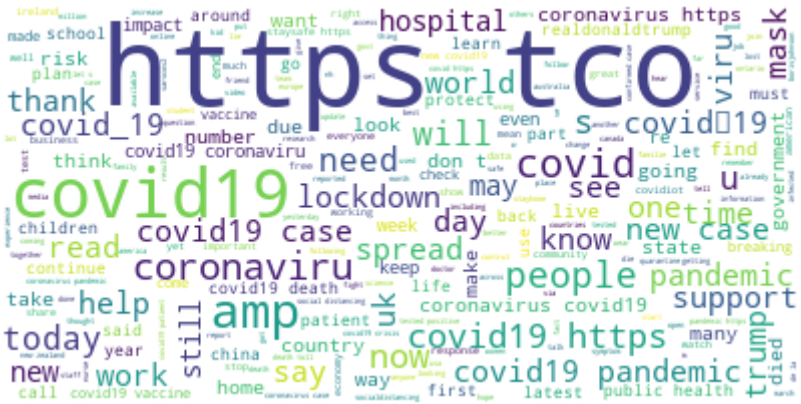
Latent Dirichlet Allocation (LDA)

In [13]:
```
df_raw_lda = df_raw.copy(deep = True)

# Remove punctuation
df_raw_lda['text_processed'] = df_raw['text'].map(lambda x:re.sub('[,\.!?]', '', x))

# Convert the titles to lowercase
df_raw_lda['text_processed'] = df_raw_lda['text_processed'].map(lambda x: x.lower())

# Print out the first rows of processed tweets
df_raw_lda['text_processed'].head()
```

Out[13]:
```
0    remember the #wuhancoronavirus the pandemic wh...
1    my sources @whitehouse say 2 tactics will be u...
2    i'll venture a wild guess: if you were running...
3    #pakistan (#greenstimulus = #nature protection...
4    🇺🇸 pandémie de #coronavirus: 30 pasteurs améri...
Name: text_processed, dtype: object
```

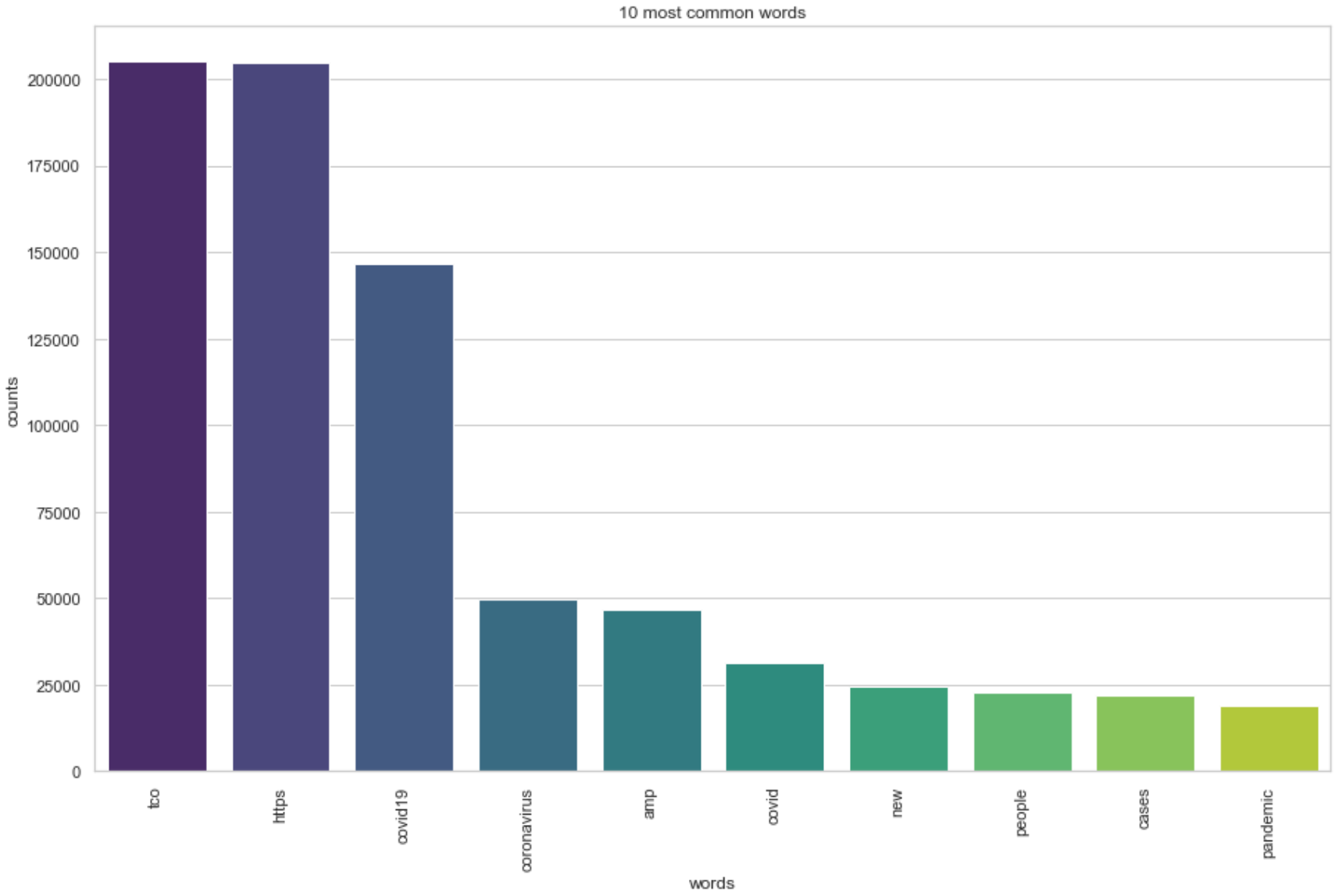
In [14]:
```
plt.rcParams["figure.figsize"] = (10,8)

# Join the different processed tweets together
long_string = ','.join(list(df_raw_lda['text_processed'].values))

# Create a WordCloud object
wordcloud = WordCloud(background_color="white", max_words=5000, contour_color='steelblue')

# Generate a word cloud
wordcloud.generate(long_string)

# Visualize the word cloud
wordcloud.to_image()
```

Out[14]:



In [15]:
```
sns.set_style('whitegrid')
%matplotlib inline

# Helper function
def plot_10_most_common_words(count_data, count_vectorizer):
    import matplotlib.pyplot as plt
    words = count_vectorizer.get_feature_names()
    total_counts = np.zeros(len(words))
    for t in count_data:
        total_counts+=t.toarray()[0]

    count_dict = (zip(words, total_counts))
    count_dict = sorted(count_dict, key=lambda x:x[1], reverse=True)[0:10]
    words = [w[0] for w in count_dict]
    counts = [w[1] for w in count_dict]
    x_pos = np.arange(len(words))

    plt.figure(2, figsize=(15, 15/1.6180))
    plt.subplot(title='10 most common words')
    sns.set_context("notebook", font_scale=1.25, rc=
    {"lines.linewidth": 2.5})
    sns.barplot(x_pos, counts, palette='viridis')
    plt.xticks(x_pos, words, rotation=90)
    plt.xlabel('words')
    plt.ylabel('counts')
    plt.show()

# Initialise the count vectorizer with the English stop words
count_vectorizer = CountVectorizer(stop_words='english')

# Fit and transform the processed titles
count_data = count_vectorizer.fit_transform(df_raw_lda['text_processed'])

# Visualise the 10 most common words
plot_10_most_common_words(count_data, count_vectorizer)
```

```
/Users/haileythanki/.local/lib/python3.9/site-packages/sklearn/utils/deprecation.py:87: FutureWarning: Functi
on get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will be removed in 1.2. Please
use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass
the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be
`data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [16]:
```
p_df = pd.read_csv("training_data.csv", low_memory = False)
docs = array(p_df['text'])
```

In [17]:
```
def docs_preprocessor(docs):
    tokenizer = RegexpTokenizer(r'\w+')
    for idx in range(len(docs)):
        docs[idx] = docs[idx].lower()  # Convert to lowercase.
        docs[idx] = tokenizer.tokenize(docs[idx])  # Split into words.
```

```
    # Remove numbers, but not words that contain numbers.
    docs = [[token for token in doc if not token.isdigit()] for doc in docs]

    # Remove words that are only one character.
    docs = [[token for token in doc if len(token) > 3] for doc in docs]

    # Lemmatize all words in documents.
    lemmatizer = WordNetLemmatizer()
    docs = [[lemmatizer.lemmatize(token) for token in doc] for doc in docs]

    return docs

docs = docs_preprocessor(docs)
```

In [18]:
```
# Add bigrams and trigrams to docs (only ones that appear 10 times or more)

bigram = Phrases(docs, min_count=10)
trigram = Phrases(bigram[docs])

for idx in range(len(docs)):
    for token in bigram[docs[idx]]:
        if '_' in token:
            # Token is a bigram, add to document.
            docs[idx].append(token)
    for token in trigram[docs[idx]]:
        if '_' in token:
            # Token is a bigram, add to document.
            docs[idx].append(token)
```

In [19]:
```
# Create a dictionary representation of the tweets

dictionary = Dictionary(docs)
print('Number of unique words in initital documents:', len(dictionary))

# Filter out words that occur less than 1000 tweets, or more than 50% of the tweets

dictionary.filter_extremes(no_below=1000, no_above=0.5)
print('Number of unique words after removing rare and common words:', len(dictionary))
```

```
Number of unique words in initital documents: 415014
Number of unique words after removing rare and common words: 608
```

In [20]:
```
corpus = [dictionary.doc2bow(doc) for doc in docs]
print('Number of unique tokens: %d' % len(dictionary))
```

```
print('Number of tweets: %d' % len(corpus))
```

```
Number of unique tokens: 608
Number of tweets: 240000
```

In [21]:
```
# Set training parameters

num_topics = 10
chunksize = 500 # size of the tweet looked at every pass
passes = 20 # number of passes through tweets
iterations = 400
eval_every = 1

# Make an index to word dictionary

temp = dictionary[0]
id2word = dictionary.id2token

%time model = LdaModel(corpus = corpus, id2word = id2word, chunksize = chunksize, alpha = 'auto', \
                       eta = 'auto', iterations = iterations, num_topics = num_topics, passes  =passes, \
                       eval_every = eval_every)
```

```
CPU times: user 11min 1s, sys: 1.85 s, total: 11min 3s
Wall time: 11min 5s
```

In [22]:
```
# feed the LDA model into the pyLDAvis instance

lda_viz = gensimvis.prepare(model, corpus, dictionary)
```

```
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/pyLDAvis/_prepare.py:246: FutureWarning: In a f
uture version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-onl
y.
  default_term_info = default_term_info.sort_values(
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
```

In [23]:

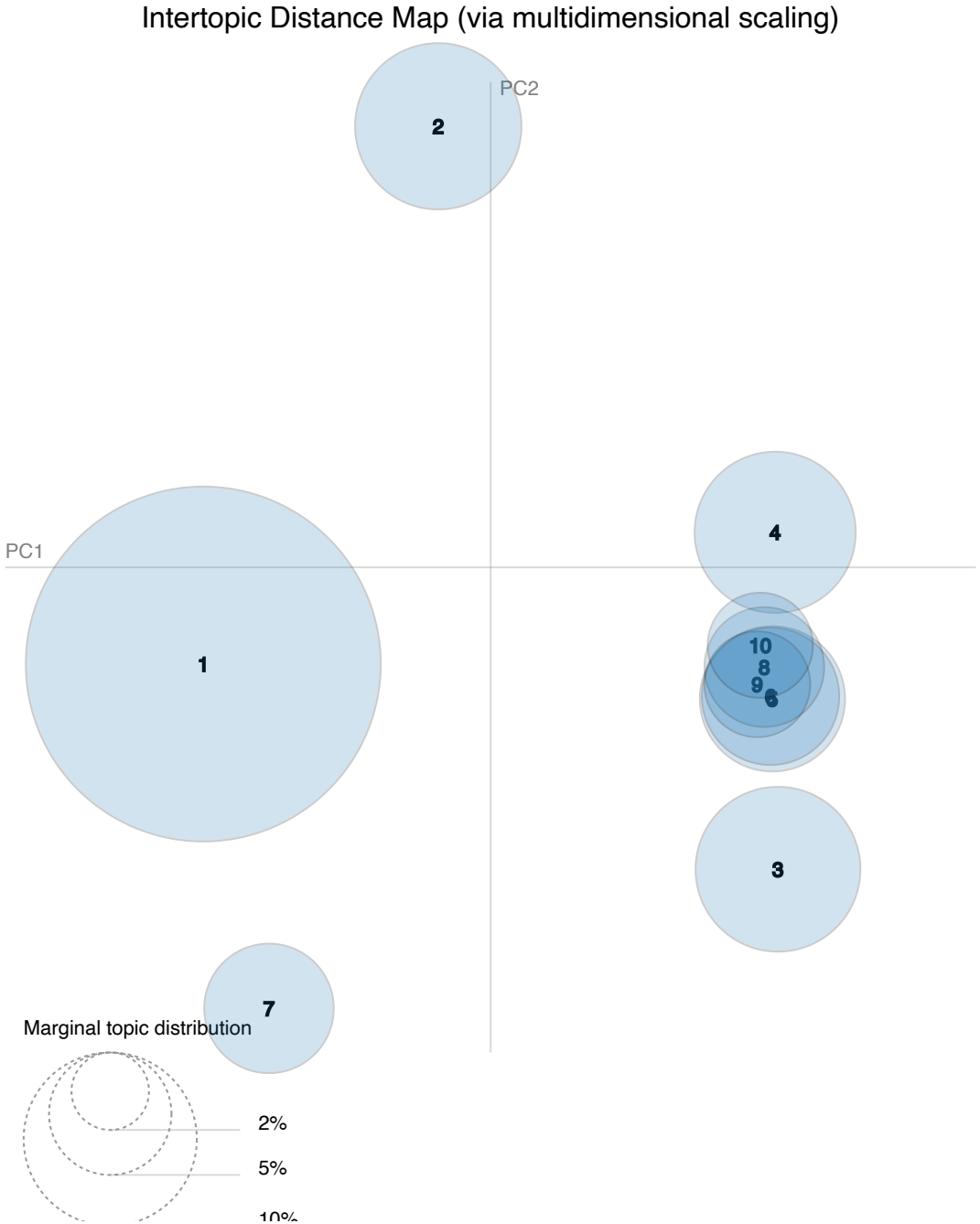
```
lda_viz
```

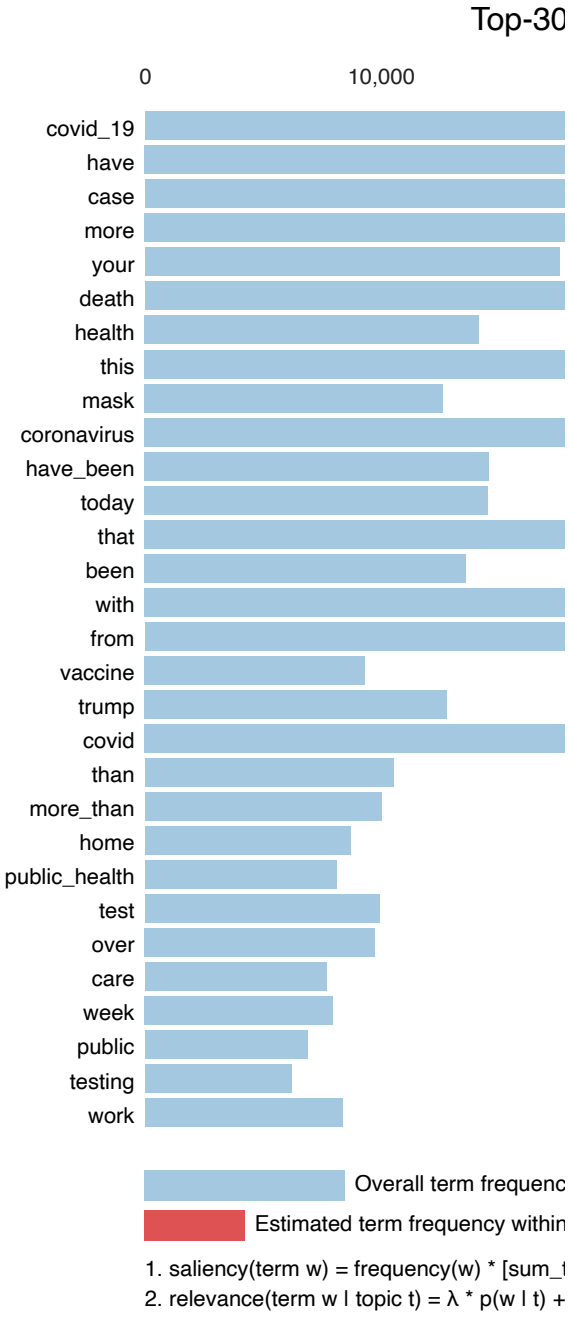
Out[23]:

Selected Topic: 0 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric:(2)

λ = 1





Non-negative Matrix Factorization

In [24]:
```
# Author: Olivier Grisel <olivier.grisel@ensta.org>
#         Lars Buitinck
#         Chyi-Kwei Yau <chyikwei.yau@gmail.com>
# License: BSD 3 clause

from time import time
import matplotlib.pyplot as plt

from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.decomposition import NMF, MiniBatchNMF, LatentDirichletAllocation
from sklearn.datasets import fetch_20newsgroups

n_samples = 2000
n_features = 1000
n_components = 10
n_top_words = 20
batch_size = 128
init = "nndsvda"


def plot_top_words(model, feature_names, n_top_words, title):
    fig, axes = plt.subplots(2, 5, figsize=(30, 15), sharex=True)
    axes = axes.flatten()
    for topic_idx, topic in enumerate(model.components_):
        top_features_ind = topic.argsort()[: -n_top_words - 1 : -1]
        top_features = [feature_names[i] for i in top_features_ind]
        weights = topic[top_features_ind]

        ax = axes[topic_idx]
        ax.barh(top_features, weights, height=0.7)
        ax.set_title(f"Topic {topic_idx +1}", fontdict={"fontsize": 30})
        ax.invert_yaxis()
        ax.tick_params(axis="both", which="major", labelsize=20)
```

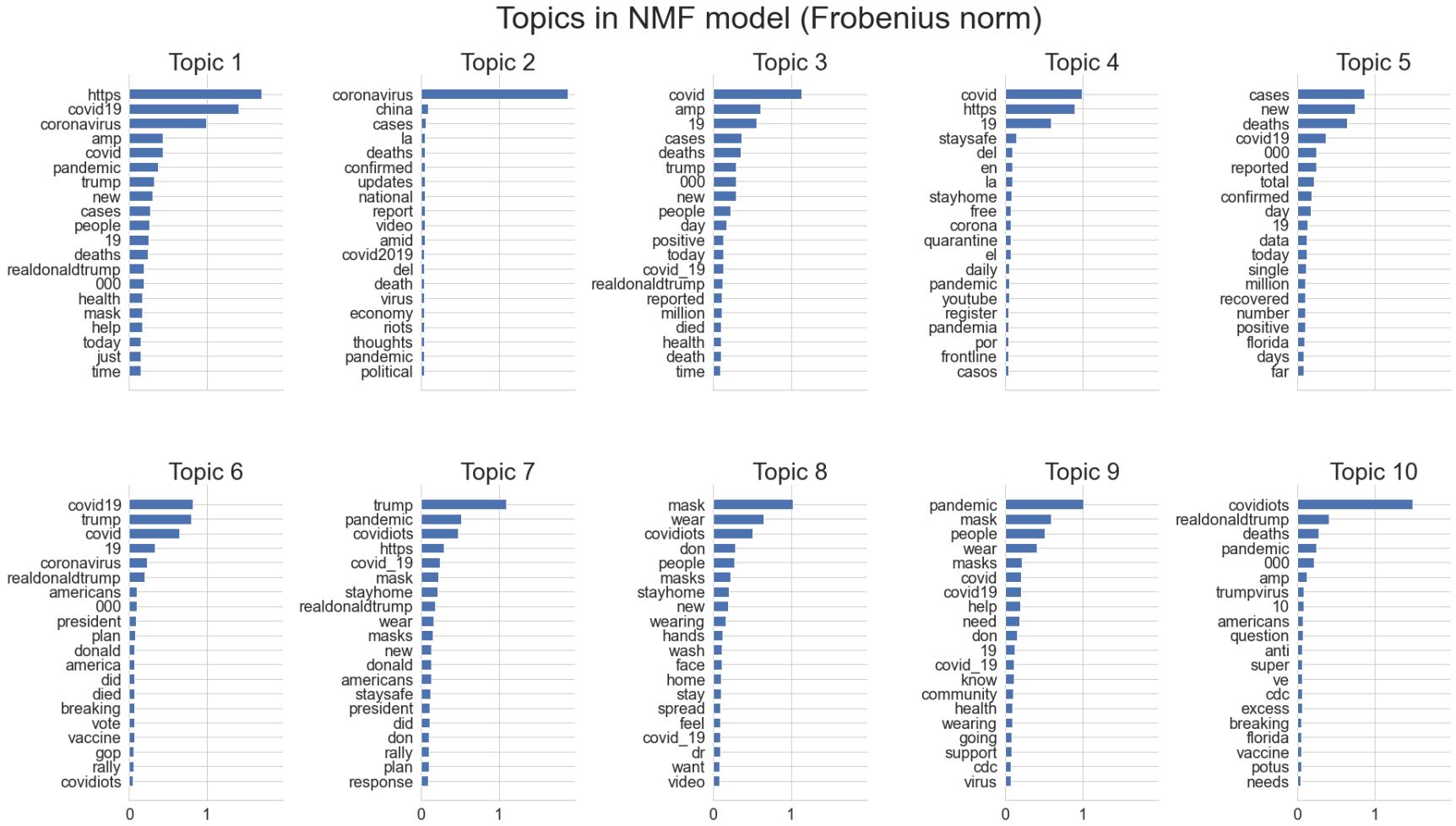
```
        for i in "top right left".split():
            ax.spines[i].set_visible(False)
        fig.suptitle(title, fontsize=40)

    plt.subplots_adjust(top=0.90, bottom=0.05, wspace=0.90, hspace=0.3)
    plt.show()


# Load the 20 newsgroups dataset and vectorize it. We use a few heuristics
# to filter out useless terms early on: the posts are stripped of headers,
# footers and quoted replies, and common English words, words occurring in
# only one document or in at least 95% of the documents are removed.

print("Loading dataset...")
t0 = time()
data = df_raw['text']

data_samples = data[:n_samples]
print("done in %0.3fs." % (time() - t0))

# Use tf-idf features for NMF.
print("Extracting tf-idf features for NMF...")
tfidf_vectorizer = TfidfVectorizer(max_features=n_features, stop_words="english"
)
t0 = time()
tfidf = tfidf_vectorizer.fit_transform(data_samples)
print("done in %0.3fs." % (time() - t0))

# Use tf (raw term count) features for LDA.
print("Extracting tf features for LDA...")
tf_vectorizer = CountVectorizer(max_features=n_features, stop_words="english"
)
t0 = time()
tf = tf_vectorizer.fit_transform(data_samples)
print("done in %0.3fs." % (time() - t0))
print()

# Fit the NMF model
print(
    "Fitting the NMF model (Frobenius norm) with tf-idf features, "
    "n_samples=%d and n_features=%d..." % (n_samples, n_features)
)
t0 = time()
nmf = NMF(
    n_components=n_components,
```

```
    random_state=1,
    init=init,
    beta_loss="frobenius",
    alpha_W=0.005,
    alpha_H=0.005,
    l1_ratio=1,
).fit(tfidf)
print("done in %0.3fs." % (time() - t0))


tfidf_feature_names = tfidf_vectorizer.get_feature_names_out()
plot_top_words(
    nmf, tfidf_feature_names, n_top_words, "Topics in NMF model (Frobenius norm)"
)

# Fit the NMF model
print(
    "\n" * 2,
    "Fitting the NMF model (generalized Kullback-Leibler "
    "divergence) with tf-idf features, n_samples=%d and n_features=%d..."
    % (n_samples, n_features),
)
t0 = time()
nmf = NMF(
    n_components=n_components,
    random_state=1,
    init=init,
    beta_loss="kullback-leibler",
    solver="mu",
    max_iter=1000,
    alpha_W=0.005,
    alpha_H=0.005,
    l1_ratio=0.5,
).fit(tfidf)
print("done in %0.3fs." % (time() - t0))

tfidf_feature_names = tfidf_vectorizer.get_feature_names_out()
plot_top_words(
    nmf,
    tfidf_feature_names,
    n_top_words,
    "Topics in NMF model (generalized Kullback-Leibler divergence)",
)

# Fit the MiniBatchNMF model
```

```
print(
    "\n" * 2,
    "Fitting the MiniBatchNMF model (Frobenius norm) with tf-idf "
    "features, n_samples=%d and n_features=%d, batch_size=%d..."
    % (n_samples, n_features, batch_size),
)
t0 = time()
mbnmf = MiniBatchNMF(
    n_components=n_components,
    random_state=1,
    batch_size=batch_size,
    init=init,
    beta_loss="frobenius",
    alpha_W=0.005,
    alpha_H=0.005,
    l1_ratio=0.5,
).fit(tfidf)
print("done in %0.3fs." % (time() - t0))


tfidf_feature_names = tfidf_vectorizer.get_feature_names_out()
plot_top_words(
    mbnmf,
    tfidf_feature_names,
    n_top_words,
    "Topics in MiniBatchNMF model (Frobenius norm)",
)

# Fit the MiniBatchNMF model
print(
    "\n" * 2,
    "Fitting the MiniBatchNMF model (generalized Kullback-Leibler "
    "divergence) with tf-idf features, n_samples=%d and n_features=%d, "
    "batch_size=%d..." % (n_samples, n_features, batch_size),
)
t0 = time()
mbnmf = MiniBatchNMF(
    n_components=n_components,
    random_state=1,
    batch_size=batch_size,
    init=init,
    beta_loss="kullback-leibler",
    alpha_W=0.00005,
    alpha_H=0.00005,
    l1_ratio=0.5,
```

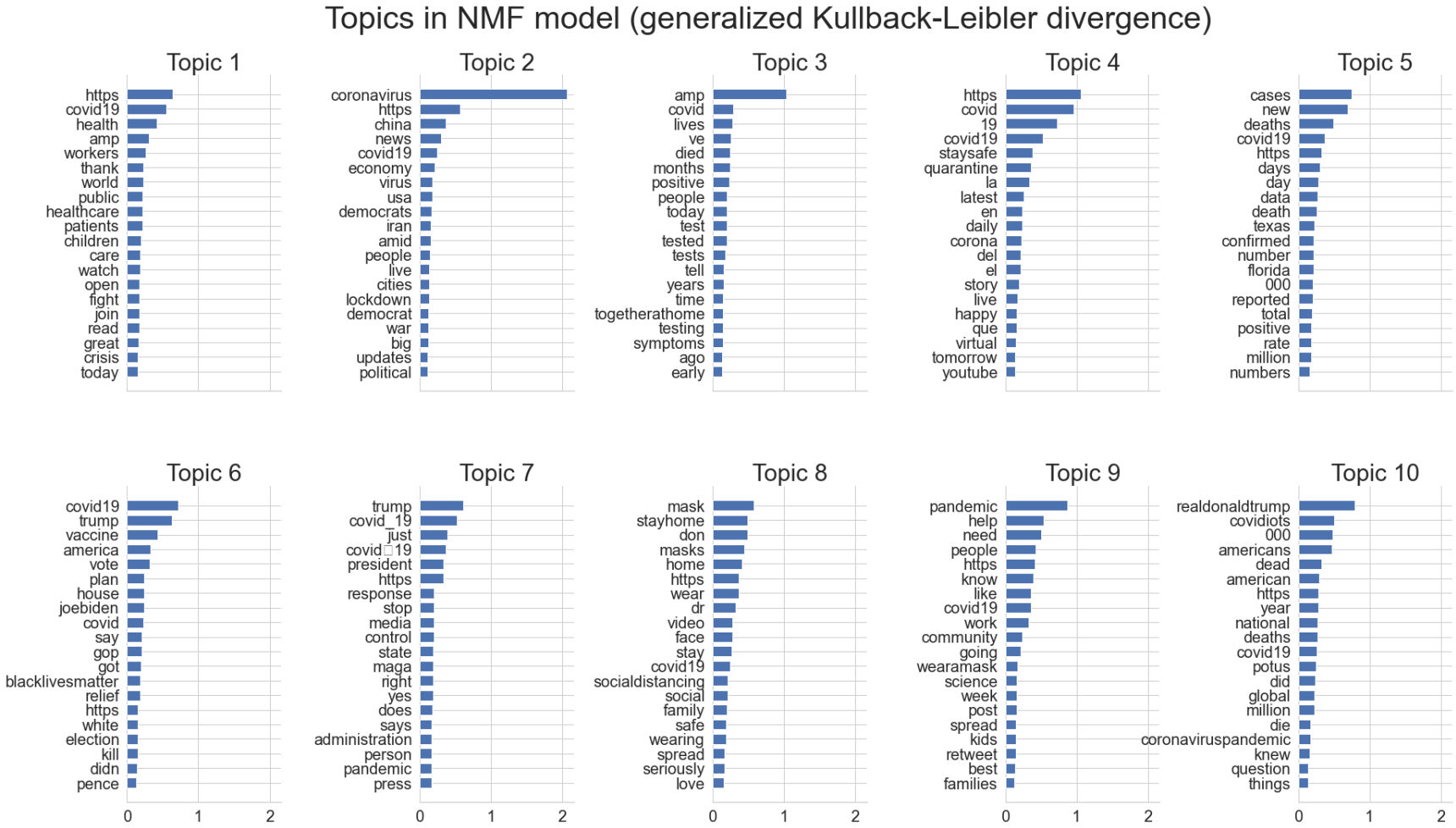
```
).fit(tfidf)
print("done in %0.3fs." % (time() - t0))

tfidf_feature_names = tfidf_vectorizer.get_feature_names_out()
plot_top_words(
    mbnmf,
    tfidf_feature_names,
    n_top_words,
    "Topics in MiniBatchNMF model (generalized Kullback-Leibler divergence)",
)

print(
    "\n" * 2,
    "Fitting LDA models with tf features, n_samples=%d and n_features=%d..."
    % (n_samples, n_features),
)
lda = LatentDirichletAllocation(
    n_components=n_components,
    max_iter=500,
    learning_method="online",
    learning_offset=0,
    random_state=0,
)
t0 = time()
lda.fit(tf)
print("done in %0.3fs." % (time() - t0))

tf_feature_names = tf_vectorizer.get_feature_names_out()
plot_top_words(lda, tf_feature_names, n_top_words, "Topics in LDA model")
```

```
Loading dataset...
done in 0.000s.
Extracting tf-idf features for NMF...
done in 0.054s.
Extracting tf features for LDA...
done in 0.047s.

Fitting the NMF model (Frobenius norm) with tf-idf features, n_samples=2000 and n_features=1000...
/Users/haileythanki/.local/lib/python3.9/site-packages/sklearn/decomposition/_nmf.py:1692: ConvergenceWarnin
g: Maximum number of iterations 200 reached. Increase it to improve convergence.
  warnings.warn(
done in 0.126s.
```

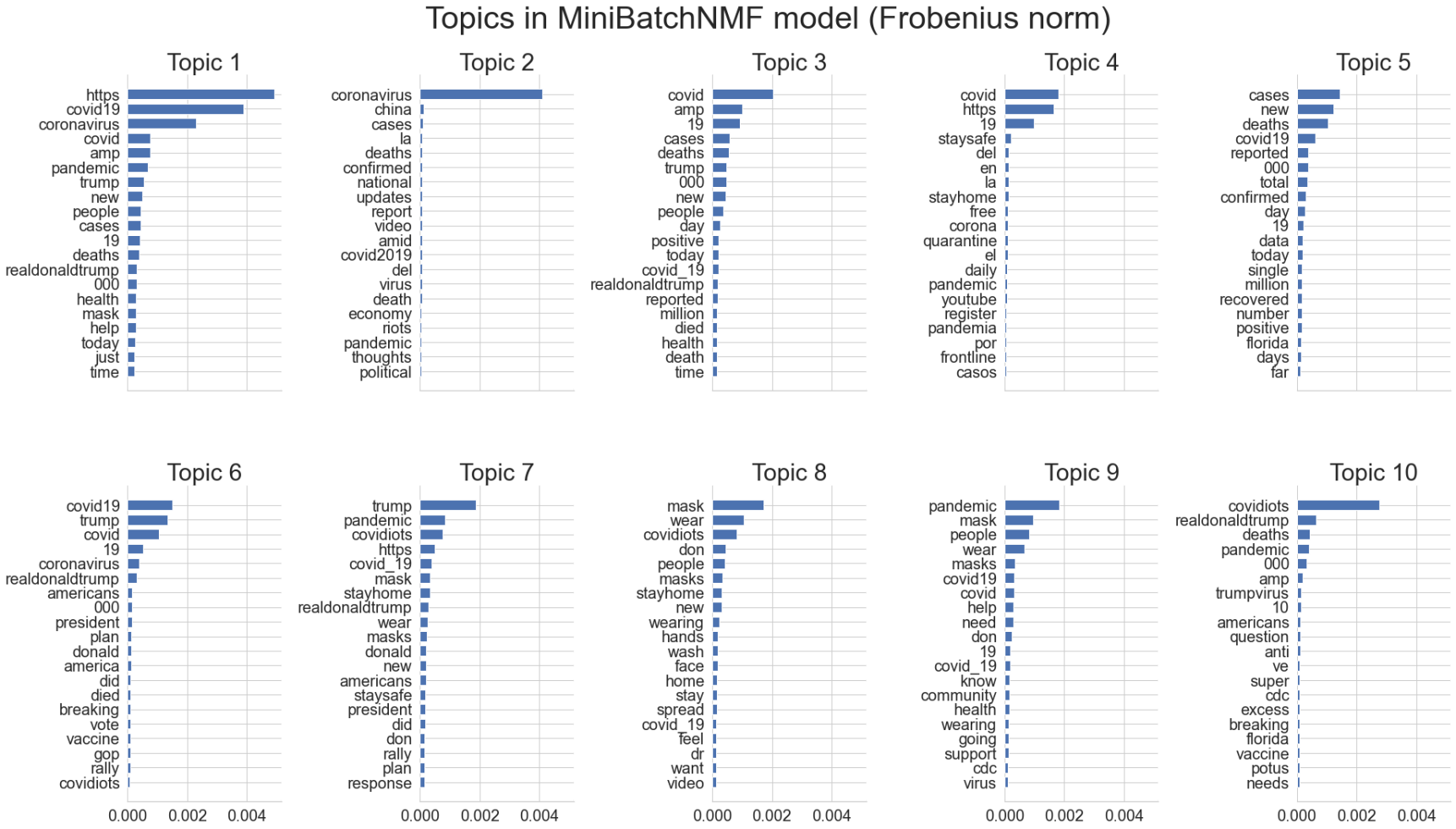



```
 Fitting the NMF model (generalized Kullback-Leibler divergence) with tf-idf features, n_samples=2000 and n_f
eatures=1000...
done in 0.482s.
```

```
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/IPython/core/pylabtools.py:151: UserWarning: Gl
yph 12540 (\N{KATAKANA-HIRAGANA PROLONGED SOUND MARK}) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```

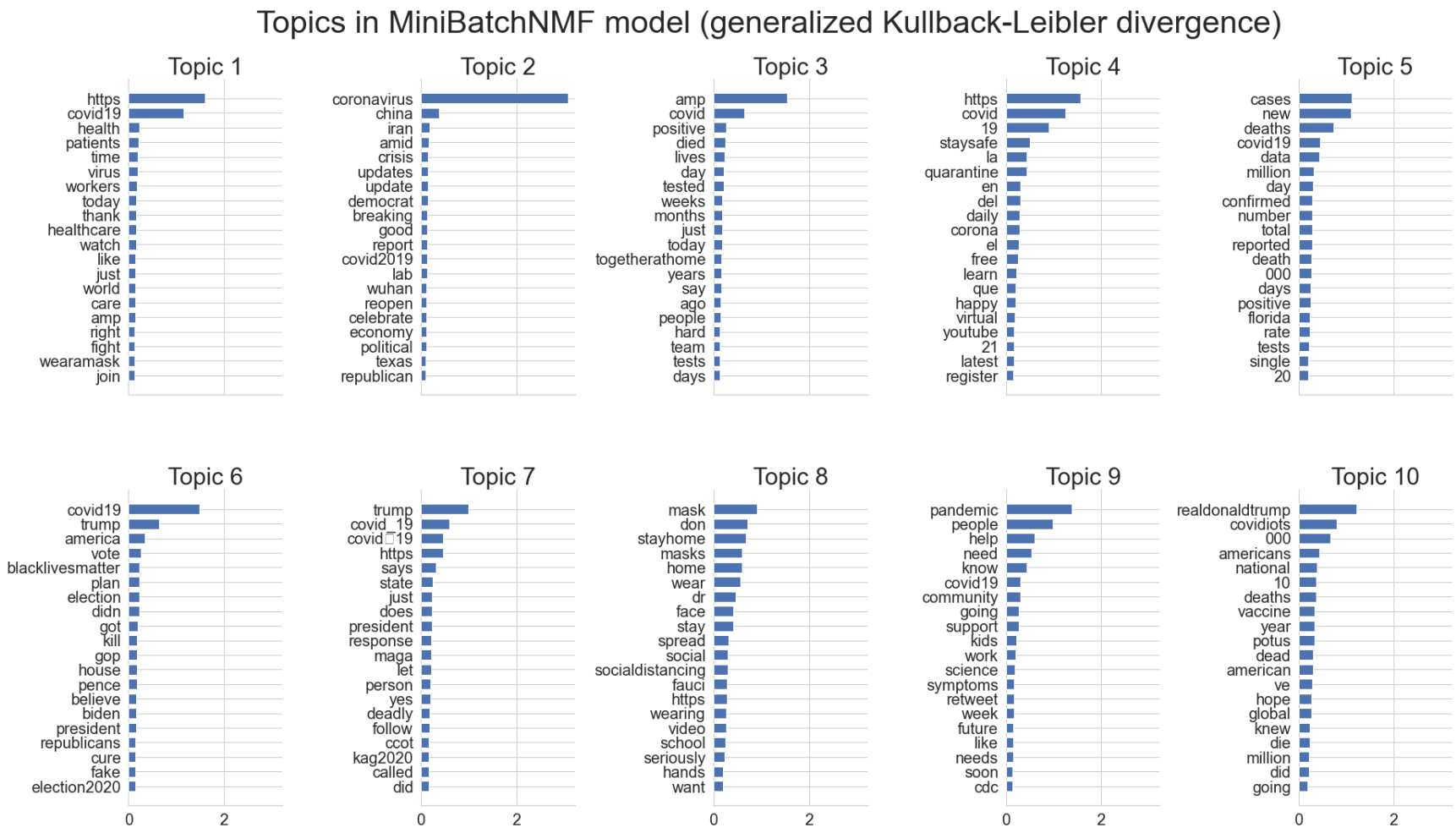



```
 Fitting the MiniBatchNMF model (Frobenius norm) with tf-idf features, n_samples=2000 and n_features=1000, ba
tch_size=128...
done in 0.097s.
```

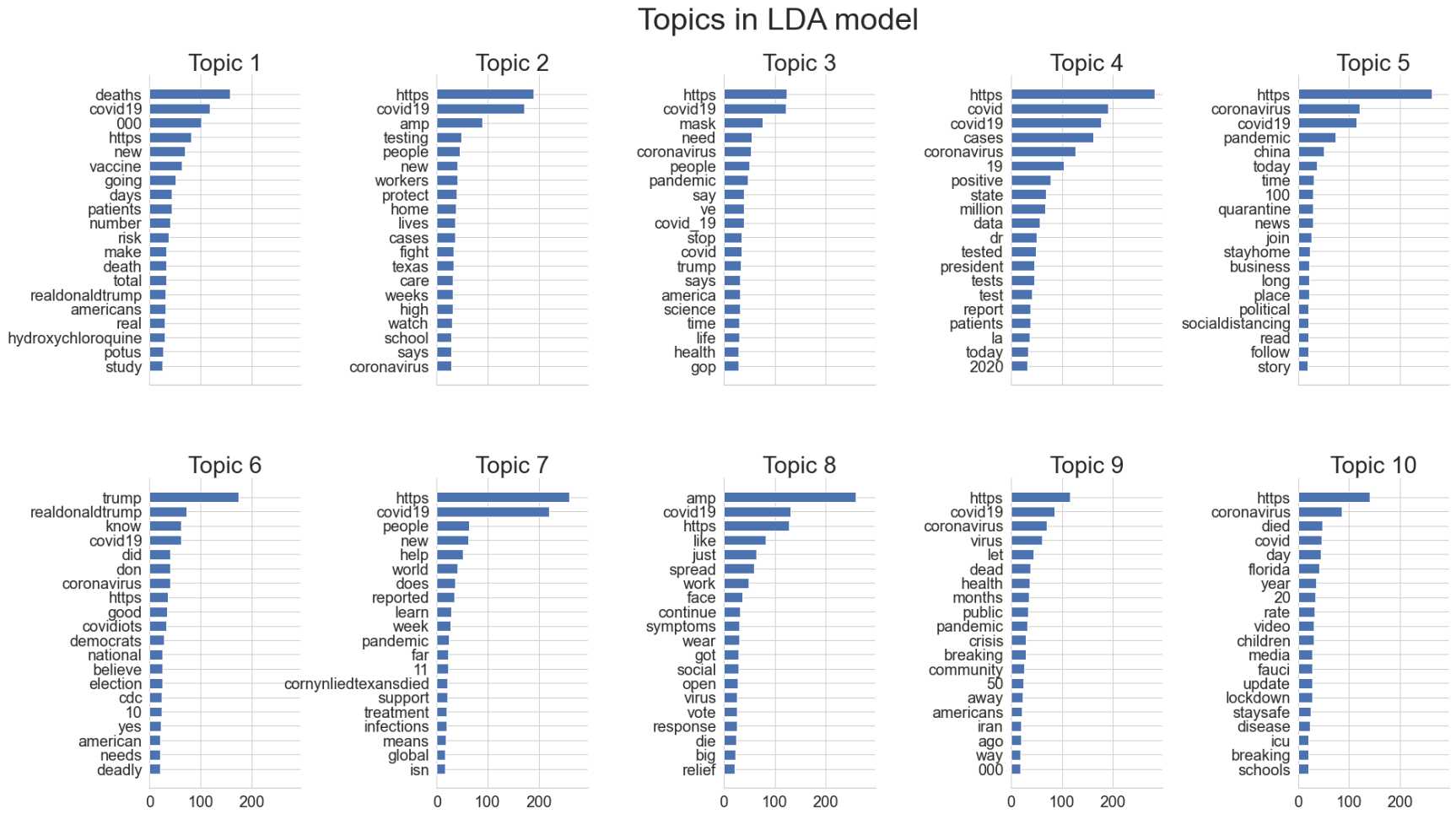



```
 Fitting the MiniBatchNMF model (generalized Kullback-Leibler divergence) with tf-idf features, n_samples=200
0 and n_features=1000, batch_size=128...
done in 0.135s.
```

```
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/IPython/core/pylabtools.py:151: UserWarning: Gl
yph 12540 (\N{KATAKANA-HIRAGANA PROLONGED SOUND MARK}) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```




```
 Fitting LDA models with tf features, n_samples=2000 and n_features=1000...
done in 71.706s.
```



Data pre-processing

In [25]:
```
from gensim.parsing.preprocessing import remove_stopwords
from gensim.parsing.preprocessing import STOPWORDS
import string

# Tokenizes tweets

def tweet_cleaner(tweet):

    # transform all words in tweets to lower case
    tweet = tweet.lower()
```

```
    # remove links
    tweet = re.sub(r"http\S+", "", tweet)

    # remove punctuation
    tweet = tweet.translate(str.maketrans('', '', string.punctuation))

    # remove emojis
    emoji_list = re.compile("["
        u"\U0001F600-\U0001F64F"  # emoticons
        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
        u"\U0001F680-\U0001F6FF"  # transport & map symbols
        u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
        u"\U00002500-\U00002BEF"  # chinese char
        u"\U00002702-\U000027B0"
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
        u"\U0001f926-\U0001f937"
        u"\U00010000-\U0010ffff"
        u"\u2640-\u2642"
        u"\u2600-\u2B55"
        u"\u200d"
        u"\u23cf"
        u"\u23e9"
        u"\u231a"
        u"\ufe0f"  # dingbats
        u"\u3030"
                      "]+", re.UNICODE)
    tweet = re.sub(emoji_list, '', tweet)

    # tokenize tweets
    tweet_tokenizer = TweetTokenizer()
    text_tokenized = tweet_tokenizer.tokenize(tweet)

    # remove stopwords
    all_stopwords_gensim = STOPWORDS.union(set(['covid', 'covid-19', 'covid19', 'pandemic', 'coronavirus', 'co
    text_filtered_1 = [word for word in text_tokenized if not word in all_stopwords_gensim]

    # remove words shorter than 3 characters
    text_filtered_2 = [text for text in text_filtered_1 if len(text) >= 3]

    return(" ".join(text_filtered_2))
```

In [26]:
```
# Tags the words in the tweets
```

```
def nltk_tag_to_wordnet_tag(nltk_tag):
    if nltk_tag.startswith('J'):
        return(wordnet.ADJ)
    elif nltk_tag.startswith('V'):
        return(wordnet.VERB)
    elif nltk_tag.startswith('N'):
        return(wordnet.NOUN)
    elif nltk_tag.startswith('R'):
        return(wordnet.ADV)
    else:
        return(None)
```

In [27]:
```
lemmatizer = WordNetLemmatizer()

# Lemmatizes the words in tweets and returns the cleaned and lemmatized tweet

def lemmatize_tweet(tweet):
    #tokenize the tweet and find the POS tag for each token
    tweet_cleaned = tweet_cleaner(tweet) #tweet_cleaner() will be the function you will write
    nltk_tagged = nltk.pos_tag(nltk.word_tokenize(tweet_cleaned))
    #tuple of (token, wordnet_tag)
    wordnet_tagged = map(lambda x: (x[0], nltk_tag_to_wordnet_tag(x[1])), nltk_tagged)
    lemmatized_tweet = []
    for word, tag in wordnet_tagged:
        if tag is None:
            #if there is no available tag, append the token as is
            lemmatized_tweet.append(word)
        else:
            #else use the tag to lemmatize the token
            lemmatized_tweet.append(lemmatizer.lemmatize(word, tag))
    return(" ".join(lemmatized_tweet))
```

In [28]:
```
# add lemmatized tweet data as another column to the original dataframe

text_cleaned_arr = []
for tweet in df_raw.text:
    cleaned_tweet = lemmatize_tweet(tweet)
    text_cleaned_arr.append(cleaned_tweet)

df_raw['text_clean'] = text_cleaned_arr
df_raw
```

Out[28]:

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_wo
0	Remember the #WuhanCoronaVirus? The pandemic w...	NaN	FALSE	TRUE	WuhanCoronaVirus KillerCuomo	us	267	
1	My sources @WhiteHouse say 2 tactics will be u...	NaN	FALSE	TRUE	Trump	us	281	
2	I'll venture a wild guess: If you were running...	NaN	FALSE	TRUE	COVID19	us	292	
3	#Pakistan (#GreenStimulus = #Nature protection...	NaN	FALSE	TRUE	Pakistan GreenStimulus Nature Green	us	236	
4	🇺🇸 Pandémie de #coronavirus: 30 pasteurs améri...	NaN	FALSE	TRUE	coronavirus COVID__19 COVID —19	us	279	
...	...	...	...	...	...	...	...	
239995	Aa Likes, Retweets yentra 🙏\n🔥🔥🔥 \n#Master	NaN	TRUE	TRUE	Master	new_zealand	39	
239996	Very interesting\nAny thoughts? \n\n#TheFive #T...	NaN	FALSE	TRUE	TheFive Trump2020 KAG2020 mondaythoughts COVID...	new_zealand	142	
239997	As we deal with #COVID19 don't forget that #Ch...	NaN	TRUE	TRUE	COVID19 Christians persecution Nigeria	new_zealand	307	
239998	While we hit 150,000 in #COVID19 deaths, the P...	NaN	FALSE	TRUE	COVID19	new_zealand	115	
239999	This too shall pass #Covid_19 . May remain sta...	NaN	FALSE	TRUE	Covid_19 HopeAlive	new_zealand	280	

240000 rows × 11 columns

In [29]:
```
# Save the data frame with lemmatized tweets

df_raw.to_csv("df_with_lemmatized_tweets.csv", index=False)
```

In [30]:
```
# Import the data frame with lemmatized tweets

df_clean = pd.read_csv("df_with_lemmatized_tweets.csv", low_memory = False)
df_clean['text_clean'] = df_clean['text_clean'] .astype('str')
df_clean
```

Out[30]:

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_wo
0	Remember the #WuhanCoronaVirus? The pandemic w...	NaN	FALSE	TRUE	WuhanCoronaVirus KillerCuomo	us	267	
1	My sources @WhiteHouse say 2 tactics will be u...	NaN	FALSE	TRUE	Trump	us	281	
2	I'll venture a wild guess: If you were running...	NaN	FALSE	TRUE	COVID19	us	292	
3	#Pakistan (#GreenStimulus = #Nature protection...	NaN	FALSE	TRUE	Pakistan GreenStimulus Nature Green	us	236	
4	🇺🇸 Pandémie de #coronavirus: 30 pasteurs améri...	NaN	FALSE	TRUE	coronavirus COVID__19 COVID —19	us	279	
...	...	...	...	...	...	...	...	
239995	Aa Likes, Retweets yentra 🙏\n🔥🔥🔥 \n#Master	NaN	TRUE	TRUE	Master	new_zealand	39	
239996	Very interesting\nAny thoughts? \n\n#TheFive #T...	NaN	FALSE	TRUE	TheFive Trump2020 KAG2020 mondaythoughts COVID...	new_zealand	142	
239997	As we deal with #COVID19 don't forget that #Ch...	NaN	TRUE	TRUE	COVID19 Christians persecution Nigeria	new_zealand	307	
239998	While we hit 150,000 in #COVID19 deaths, the P...	NaN	FALSE	TRUE	COVID19	new_zealand	115	
239999	This too shall pass #Covid_19 . May remain sta...	NaN	FALSE	TRUE	Covid_19 HopeAlive	new_zealand	280	

240000 rows × 11 columns

Descriptive Analysis on Clean Data

Analyzing character and word counts for tweets

In [31]:
```
# Calculating character and word count for each tweet

df_clean['clean_text_char_count'] = df_clean['text_clean'].astype(str).apply(len)
df_clean['clean_text_word_count'] = df_clean['text_clean'].apply(lambda x: len(str(x).split()))
df_clean
```

Out[31]:

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_wo
0	Remember the #WuhanCoronaVirus? The pandemic w...	NaN	FALSE	TRUE	WuhanCoronaVirus KillerCuomo	us	267	
1	My sources @WhiteHouse say 2 tactics will be u...	NaN	FALSE	TRUE	Trump	us	281	
2	I'll venture a wild guess: If you were running...	NaN	FALSE	TRUE	COVID19	us	292	
3	#Pakistan (#GreenStimulus = #Nature protection...	NaN	FALSE	TRUE	Pakistan GreenStimulus Nature Green	us	236	
4	🇺🇸 Pandémie de #coronavirus: 30 pasteurs améri...	NaN	FALSE	TRUE	coronavirus COVID__19 COVID —19	us	279	
...	...	...	...	...	...	...	...	
239995	Aa Likes, Retweets yentra 🙏\n🔥🔥🔥 \n#Master	NaN	TRUE	TRUE	Master	new_zealand	39	
239996	Very interesting\nAny thoughts? \n\n#TheFive #T...	NaN	FALSE	TRUE	TheFive Trump2020 KAG2020 mondaythoughts COVID...	new_zealand	142	
239997	As we deal with #COVID19 don't forget that #Ch...	NaN	TRUE	TRUE	COVID19 Christians persecution Nigeria	new_zealand	307	
239998	While we hit 150,000 in #COVID19 deaths, the P...	NaN	FALSE	TRUE	COVID19	new_zealand	115	
239999	This too shall pass #Covid_19 . May remain sta...	NaN	FALSE	TRUE	Covid_19 HopeAlive	new_zealand	280	

240000 rows × 13 columns

In [32]:
```
# Calculating the mean, median, maximum and minimum of word count and character count of tweets

table = [['Attribute', 'Aggregation Type', 'Value'],
         ['Character Count', 'mean', round((df_clean.clean_text_char_count.mean()),2)],
         ['', 'median', round((df_clean.clean_text_char_count.median()),2)],
         ['', 'max', round((df_clean.clean_text_char_count.max()),2)],
         ['', 'min', round((df_clean.clean_text_char_count.min()),2)],
         ['Word Count', 'mean', round((df_clean.clean_text_word_count.mean()),2)],
         ['', 'median', round((df_clean.clean_text_word_count.median()),2)],
         ['', 'max', round((df_clean.clean_text_word_count.max()),2)],
         ['', 'min', round((df_clean.clean_text_word_count.min()),2)]]
print(tabulate(table, headers='firstrow'))
```

```
Attribute        Aggregation Type      Value
---------------  ------------------  -------
Character Count  mean                112.13
                 median              116
                 max                 316
                 min                   2
Word Count       mean                 14.86
                 median               15
                 max                  44
                 min                   1
```

Analyzing frequency of words in cleaned tweets

In [33]:
```
# Splitting each tweet into individual words

words_in_tweet = [str(tweet).lower().split() for tweet in df_clean.text_clean]

# List of all words across tweets

all_words = list(itertools.chain(*words_in_tweet))

# Create a word frequency counter

word_freq = collections.Counter(all_words)
top_10_words_freq = word_freq.most_common(10)
top_10_words_freq
```

Out[33]:
```
[('amp', 45165),
 ('case', 25477),
 ('new', 24486),
 ('people', 22965),
 ('test', 19252),
 ('death', 18444),
 ('day', 14883),
 ('health', 13505),
 ('trump', 13151),
 ('need', 12933)]
```

In [34]:
```
top_10_words = []

for i in range(0,10):
    word = top_10_words_freq[i][0]
    top_10_words.append(word)

top_10_words
```

Out[34]:
```
['amp',
 'case',
 'new',
 'people',
 'test',
 'death',
 'day',
 'health',
 'trump',
 'need']
```

In [35]:
```
countries = df_raw.country.unique()
countries
```

Out[35]:
```
array(['us', 'uk', 'canada', 'australia', 'ireland', 'new_zealand'],
      dtype=object)
```

In [36]:
```
df_top_10_words_by_country = pd.DataFrame(columns=['country', 'word','freq'])

for country in countries:
    df_clean_country = df_clean[df_clean.country == country]
    words_in_tweet = [str(tweet).lower().split() for tweet in df_clean_country.text_clean]
    all_words = list(itertools.chain(*words_in_tweet))
    for word in top_10_words:
        freq = all_words.count(word)
        df_top_10_words_curr_country = pd.DataFrame([[country, word, freq]], columns = ['country', 'word','fre
        df_top_10_words_by_country = pd.concat([df_top_10_words_by_country, df_top_10_words_curr_country])
```

```
df_top_10_words_by_country.reset_index().drop(columns='index')
```

Out[36]:

	country	word	freq
0	us	amp	6934
1	us	case	4018
2	us	new	3596
3	us	people	3877
4	us	test	3526
5	us	death	3518
6	us	day	2523
7	us	health	2023
8	us	trump	4765
9	us	need	2101
10	uk	amp	7244
11	uk	case	3008
12	uk	new	3636
13	uk	people	3906
14	uk	test	3399
15	uk	death	3129
16	uk	day	2281
17	uk	health	2062
18	uk	trump	1601
19	uk	need	2173
20	canada	amp	6825
21	canada	case	4523
22	canada	new	4297
23	canada	people	3621
24	canada	test	3128
25	canada	death	3074

	country	word	freq
26	canada	day	2166
27	canada	health	2371
28	canada	trump	1241
29	canada	need	2180
30	australia	amp	7932
31	australia	case	5034
32	australia	new	3850
33	australia	people	3638
34	australia	test	2867
35	australia	death	2583
36	australia	day	2303
37	australia	health	2311
38	australia	trump	2066
39	australia	need	2020
40	ireland	amp	8293
41	ireland	case	3287
42	ireland	new	3365
43	ireland	people	4302
44	ireland	test	2671
45	ireland	death	2896
46	ireland	day	2901
47	ireland	health	2467
48	ireland	trump	1086
49	ireland	need	2404
50	new_zealand	amp	7937
51	new_zealand	case	5607

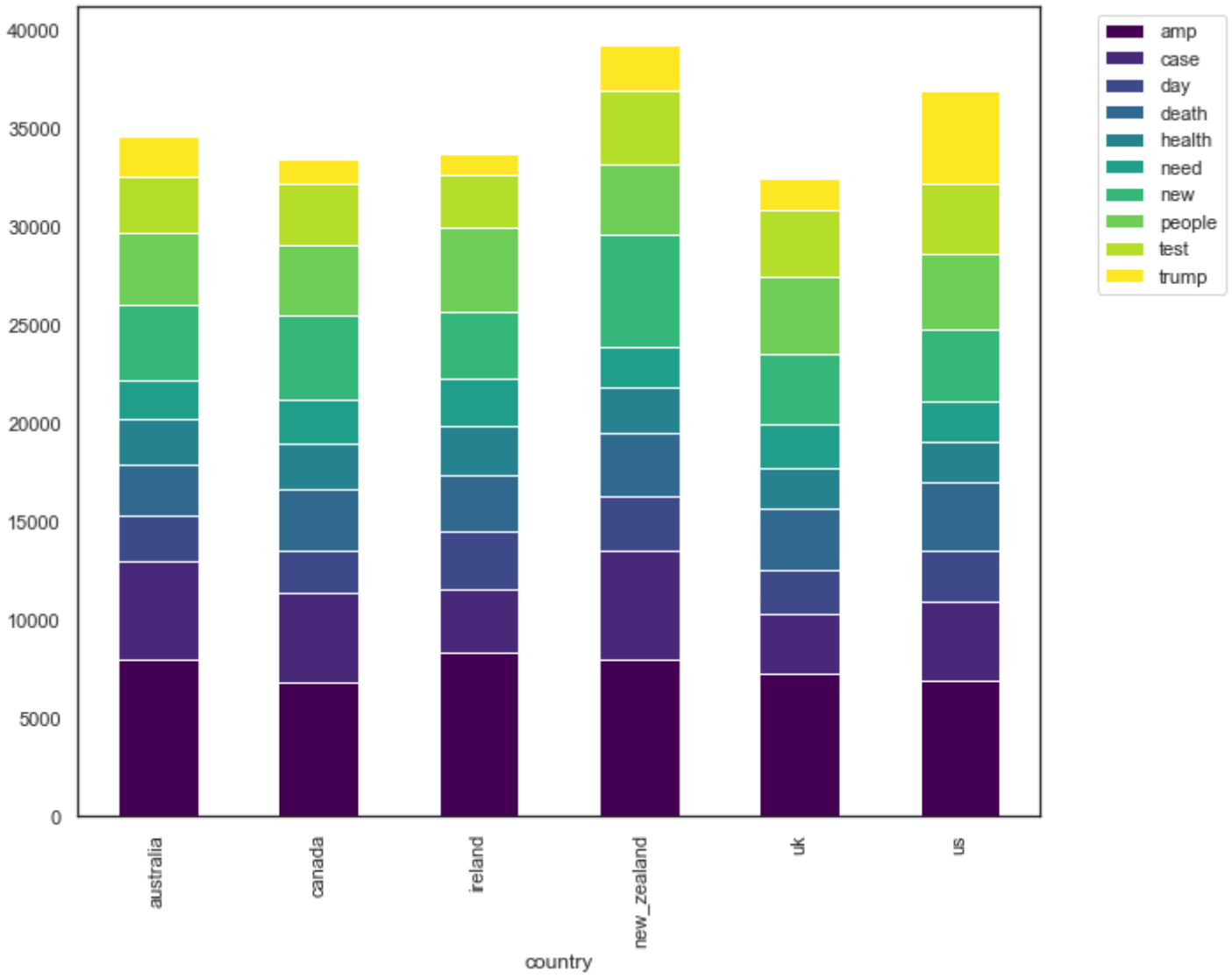
	country	word	freq
52	new_zealand	new	5742
53	new_zealand	people	3621
54	new_zealand	test	3661
55	new_zealand	death	3244
56	new_zealand	day	2709
57	new_zealand	health	2271
58	new_zealand	trump	2392
59	new_zealand	need	2055

In [37]:
```
df_top_10_words_by_country = df_top_10_words_by_country.groupby(['country', 'word'])['freq'].sum().unstack().
df_top_10_words_by_country
```

Out[37]:

word	amp	case	day	death	health	need	new	people	test	trump
country										
australia	7932	5034	2303	2583	2311	2020	3850	3638	2867	2066
canada	6825	4523	2166	3074	2371	2180	4297	3621	3128	1241
ireland	8293	3287	2901	2896	2467	2404	3365	4302	2671	1086
new_zealand	7937	5607	2709	3244	2271	2055	5742	3621	3661	2392
uk	7244	3008	2281	3129	2062	2173	3636	3906	3399	1601
us	6934	4018	2523	3518	2023	2101	3596	3877	3526	4765

In [38]:
```
plt.rcParams["figure.figsize"] = (10,8)

#set seaborn plotting aesthetics
sns.set(style='white')

#create stacked bar chart
df_top_10_words_by_country.plot(kind='bar', stacked=True, cmap="viridis")
plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')
plt.tight_layout()

plt.show()
```



Latent Dirichlet Allocation (LDA)

In [39]:
```
df_clean_lda = df_clean.copy(deep = True)
df_clean_lda
```

Out[39]:

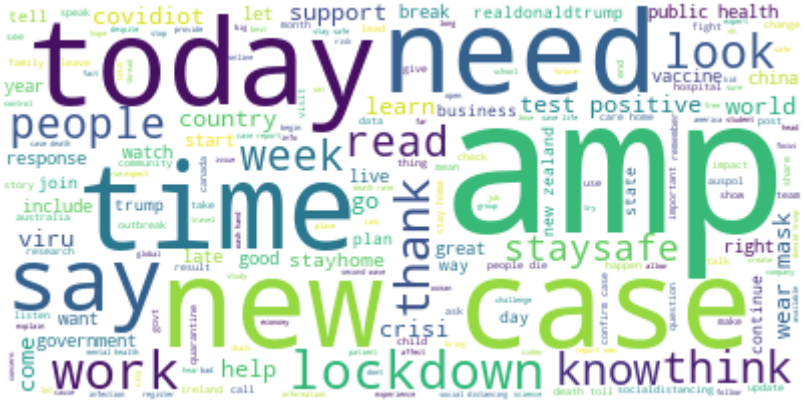
	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_wo
0	Remember the #WuhanCoronaVirus? The pandemic w...	NaN	FALSE	TRUE	WuhanCoronaVirus KillerCuomo	us	267	
1	My sources @WhiteHouse say 2 tactics will be u...	NaN	FALSE	TRUE	Trump	us	281	
2	I'll venture a wild guess: If you were running...	NaN	FALSE	TRUE	COVID19	us	292	
3	#Pakistan (#GreenStimulus = #Nature protection...	NaN	FALSE	TRUE	Pakistan GreenStimulus Nature Green	us	236	
4	🇺🇸 Pandémie de #coronavirus: 30 pasteurs améri...	NaN	FALSE	TRUE	coronavirus COVID__19 COVID —19	us	279	
...	...	...	...	...	...	...	...	
239995	Aa Likes, Retweets yentra 🙏\n🔥🔥🔥 \n#Master	NaN	TRUE	TRUE	Master	new_zealand	39	
239996	Very interesting\nAny thoughts? \n\n#TheFive #T...	NaN	FALSE	TRUE	TheFive Trump2020 KAG2020 mondaythoughts COVID...	new_zealand	142	
239997	As we deal with #COVID19 don't forget that #Ch...	NaN	TRUE	TRUE	COVID19 Christians persecution Nigeria	new_zealand	307	
239998	While we hit 150,000 in #COVID19 deaths, the P...	NaN	FALSE	TRUE	COVID19	new_zealand	115	
239999	This too shall pass #Covid_19 . May remain sta...	NaN	FALSE	TRUE	Covid_19 HopeAlive	new_zealand	280	

240000 rows × 13 columns

In [ ]:

In [40]:
```
plt.rcParams["figure.figsize"] = (10,8)

# Join the different processed tweets together
long_string = ','.join(list(df_clean_lda['text_clean'].values))

# Create a WordCloud object
wordcloud = WordCloud(background_color="white", max_words=5000, contour_color='steelblue')

# Generate a word cloud
wordcloud.generate(long_string)

# Visualize the word cloud
wordcloud.to_image()
```

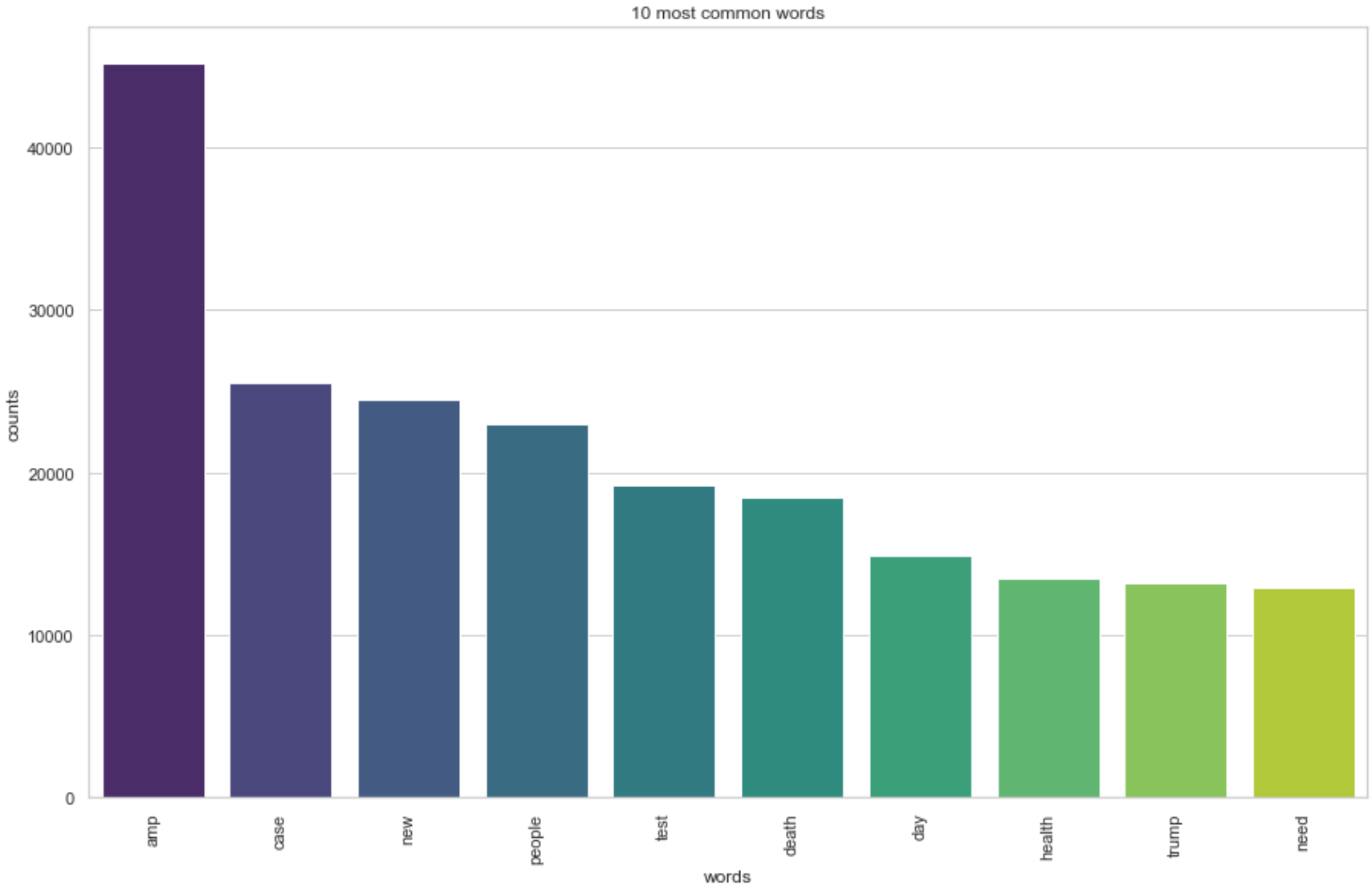
Out[40]:



In [41]:
```
sns.set_style('whitegrid')
%matplotlib inline

# Helper function
def plot_10_most_common_words(count_data, count_vectorizer):
    import matplotlib.pyplot as plt
    words = count_vectorizer.get_feature_names()
    total_counts = np.zeros(len(words))
    for t in count_data:
        total_counts+=t.toarray()[0]
```

```
    count_dict = (zip(words, total_counts))
    count_dict = sorted(count_dict, key=lambda x:x[1], reverse=True)[0:10]
    words = [w[0] for w in count_dict]
    counts = [w[1] for w in count_dict]
    x_pos = np.arange(len(words))

    plt.figure(2, figsize=(15, 15/1.6180))
    plt.subplot(title='10 most common words')
    sns.set_context("notebook", font_scale=1.25, rc=
    {"lines.linewidth": 2.5})
    sns.barplot(x_pos, counts, palette='viridis')
    plt.xticks(x_pos, words, rotation=90)
    plt.xlabel('words')
    plt.ylabel('counts')
    plt.show()

# Initialise the count vectorizer with the English stop words
count_vectorizer = CountVectorizer(stop_words='english')

# Fit and transform the processed titles
count_data = count_vectorizer.fit_transform(df_clean_lda['text_clean'])

# Visualise the 10 most common words
plot_10_most_common_words(count_data, count_vectorizer)
```

```
/Users/haileythanki/.local/lib/python3.9/site-packages/sklearn/utils/deprecation.py:87: FutureWarning: Functi
on get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will be removed in 1.2. Please
use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass
the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be
`data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [42]:
```
p_df = pd.read_csv("df_with_lemmatized_tweets.csv", low_memory = False)
p_df['text_clean'] = p_df['text_clean'].astype('str')
docs = array(p_df['text_clean'])
```

In [43]:
```
def docs_preprocessor(docs):
    tokenizer = RegexpTokenizer(r'\w+')
    for idx in range(len(docs)):
        docs[idx] = docs[idx].lower()  # Convert to lowercase.
        docs[idx] = tokenizer.tokenize(docs[idx])  # Split into words.
```

```
    # Remove numbers, but not words that contain numbers.
    docs = [[token for token in doc if not token.isdigit()] for doc in docs]

    # Remove words that are only one character.
    docs = [[token for token in doc if len(token) > 3] for doc in docs]

    # Lemmatize all words in documents.
    lemmatizer = WordNetLemmatizer()
    docs = [[lemmatizer.lemmatize(token) for token in doc] for doc in docs]

    return docs

docs = docs_preprocessor(docs)
```

In [44]:
```
# Add bigrams and trigrams to docs (only ones that appear 10 times or more)

bigram = Phrases(docs, min_count=100)
trigram = Phrases(bigram[docs])

for idx in range(len(docs)):
    for token in bigram[docs[idx]]:
        if '_' in token:
            # Token is a bigram, add to document.
            docs[idx].append(token)
    for token in trigram[docs[idx]]:
        if '_' in token:
            # Token is a bigram, add to document.
            docs[idx].append(token)
```

In [45]:
```
# Create a dictionary representation of the tweets

dictionary = Dictionary(docs)
print('Number of unique words in initital documents:', len(dictionary))

# Filter out words that occur less than 1000 tweets, or more than 50% of the tweets

dictionary.filter_extremes(no_below=1000, no_above=0.5)
print('Number of unique words after removing rare and common words:', len(dictionary))
```

```
Number of unique words in initital documents: 255752
Number of unique words after removing rare and common words: 498
```

In [46]:
```
corpus = [dictionary.doc2bow(doc) for doc in docs]
print('Number of unique tokens: %d' % len(dictionary))
```

```
print('Number of tweets: %d' % len(corpus))
```

```
Number of unique tokens: 498
Number of tweets: 240000
```

In [47]:
```
# Set training parameters

num_topics = 10
chunksize = 500 # size of the tweet looked at every pass
passes = 20 # number of passes through tweets
iterations = 400
eval_every = 1

# Make an index to word dictionary

temp = dictionary[0]
id2word = dictionary.id2token

%time model = LdaModel(corpus = corpus, id2word = id2word, chunksize = chunksize, alpha = 'auto', \
                       eta = 'auto', iterations = iterations, num_topics = num_topics, passes  =passes, \
                       eval_every = eval_every)
```

```
CPU times: user 9min 43s, sys: 1.04 s, total: 9min 44s
Wall time: 9min 44s
```

In [48]:
```
# feed the LDA model into the pyLDAvis instance

lda_viz = gensimvis.prepare(model, corpus, dictionary)
```

```
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/pyLDAvis/_prepare.py:246: FutureWarning: In a f
uture version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-onl
y.
  default_term_info = default_term_info.sort_values(
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
/Users/haileythanki/opt/anaconda3/lib/python3.9/site-packages/past/builtins/misc.py:45: DeprecationWarning: t
he imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  from imp import reload
```

In [49]:

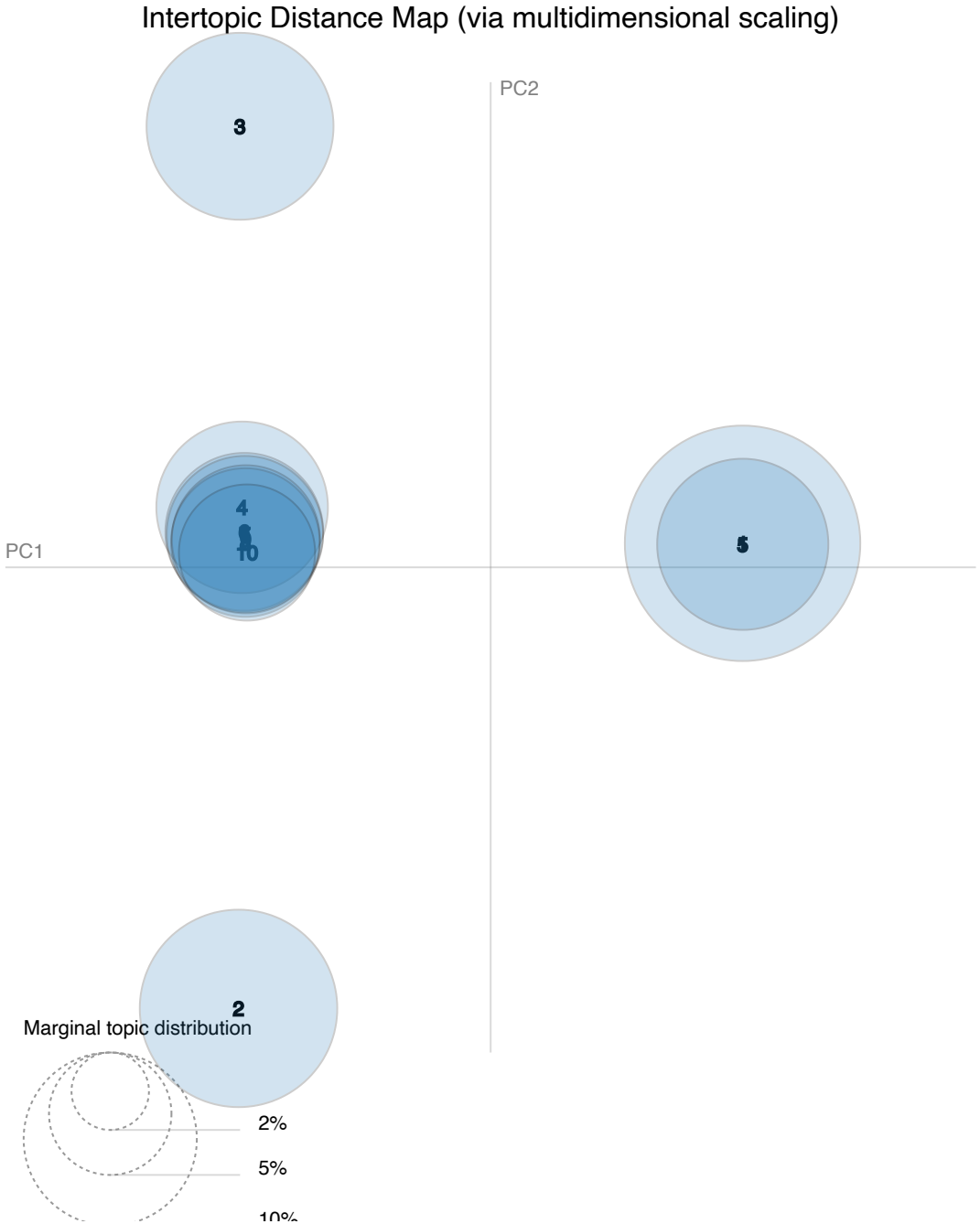
```
lda_viz
```

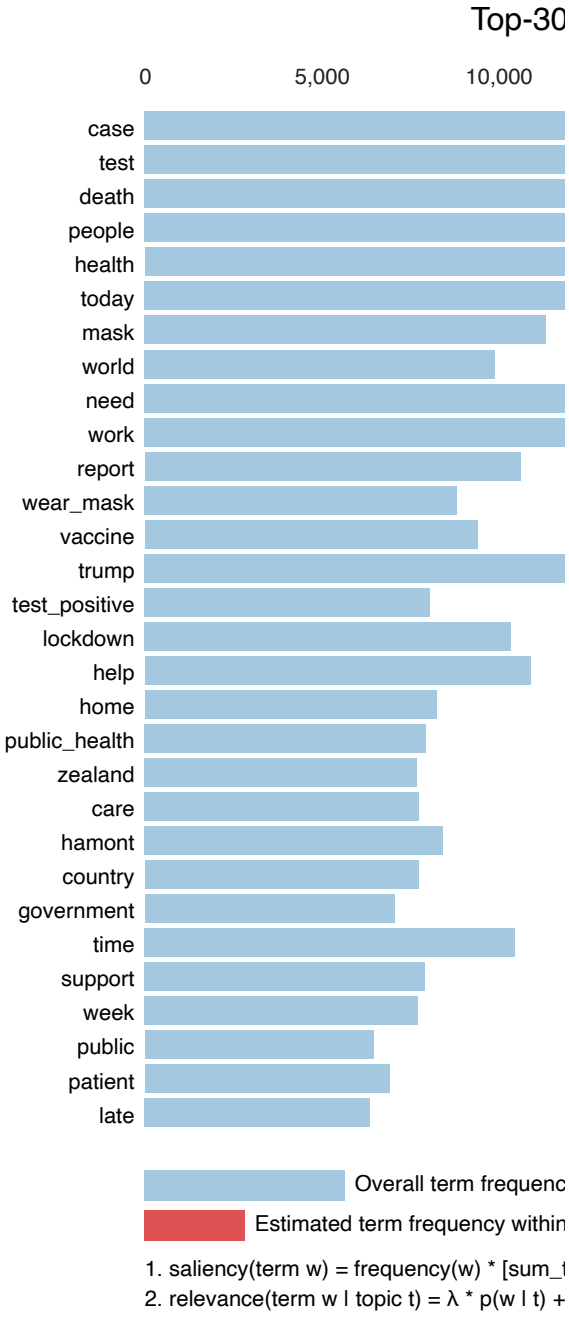
Out[49]:

Selected Topic: 0 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric:(2)

λ = 1





Non-negative matrix factorization

In [50]:
```
# Author: Olivier Grisel <olivier.grisel@ensta.org>
#         Lars Buitinck
#         Chyi-Kwei Yau <chyikwei.yau@gmail.com>
# License: BSD 3 clause


n_samples = 2000
n_features = 1000
n_components = 10
n_top_words = 20
batch_size = 128
init = "nndsvda"


def plot_top_words(model, feature_names, n_top_words, title):
    fig, axes = plt.subplots(2, 5, figsize=(30, 15), sharex=True)
    axes = axes.flatten()
    for topic_idx, topic in enumerate(model.components_):
        top_features_ind = topic.argsort()[: -n_top_words - 1 : -1]
        top_features = [feature_names[i] for i in top_features_ind]
        weights = topic[top_features_ind]

        ax = axes[topic_idx]
        ax.barh(top_features, weights, height=0.7)
        ax.set_title(f"Topic {topic_idx +1}", fontdict={"fontsize": 30})
        ax.invert_yaxis()
        ax.tick_params(axis="both", which="major", labelsize=20)
        for i in "top right left".split():
            ax.spines[i].set_visible(False)
        fig.suptitle(title, fontsize=40)

    plt.subplots_adjust(top=0.90, bottom=0.05, wspace=0.90, hspace=0.3)
    plt.show()
```

```
print("Loading dataset...")
t0 = time()
data = df_clean['text_clean']

data_samples = data[:n_samples]
print("done in %0.3fs." % (time() - t0))

# Use tf-idf features for NMF.
print("Extracting tf-idf features for NMF...")
tfidf_vectorizer = TfidfVectorizer(max_features=n_features, stop_words="english"
)
t0 = time()
tfidf = tfidf_vectorizer.fit_transform(data_samples)
print("done in %0.3fs." % (time() - t0))

# Use tf (raw term count) features for LDA.
print("Extracting tf features for LDA...")
tf_vectorizer = CountVectorizer(max_features=n_features, stop_words="english"
)
t0 = time()
tf = tf_vectorizer.fit_transform(data_samples)
print("done in %0.3fs." % (time() - t0))
print()

# Fit the NMF model
print(
    "Fitting the NMF model (Frobenius norm) with tf-idf features, "
    "n_samples=%d and n_features=%d..." % (n_samples, n_features)
)
t0 = time()
nmf = NMF(
    n_components=n_components,
    random_state=1,
    init=init,
    beta_loss="frobenius",
    alpha_W=0.005,
    alpha_H=0.005,
    l1_ratio=1,
).fit(tfidf)
print("done in %0.3fs." % (time() - t0))


tfidf_feature_names = tfidf_vectorizer.get_feature_names_out()
```

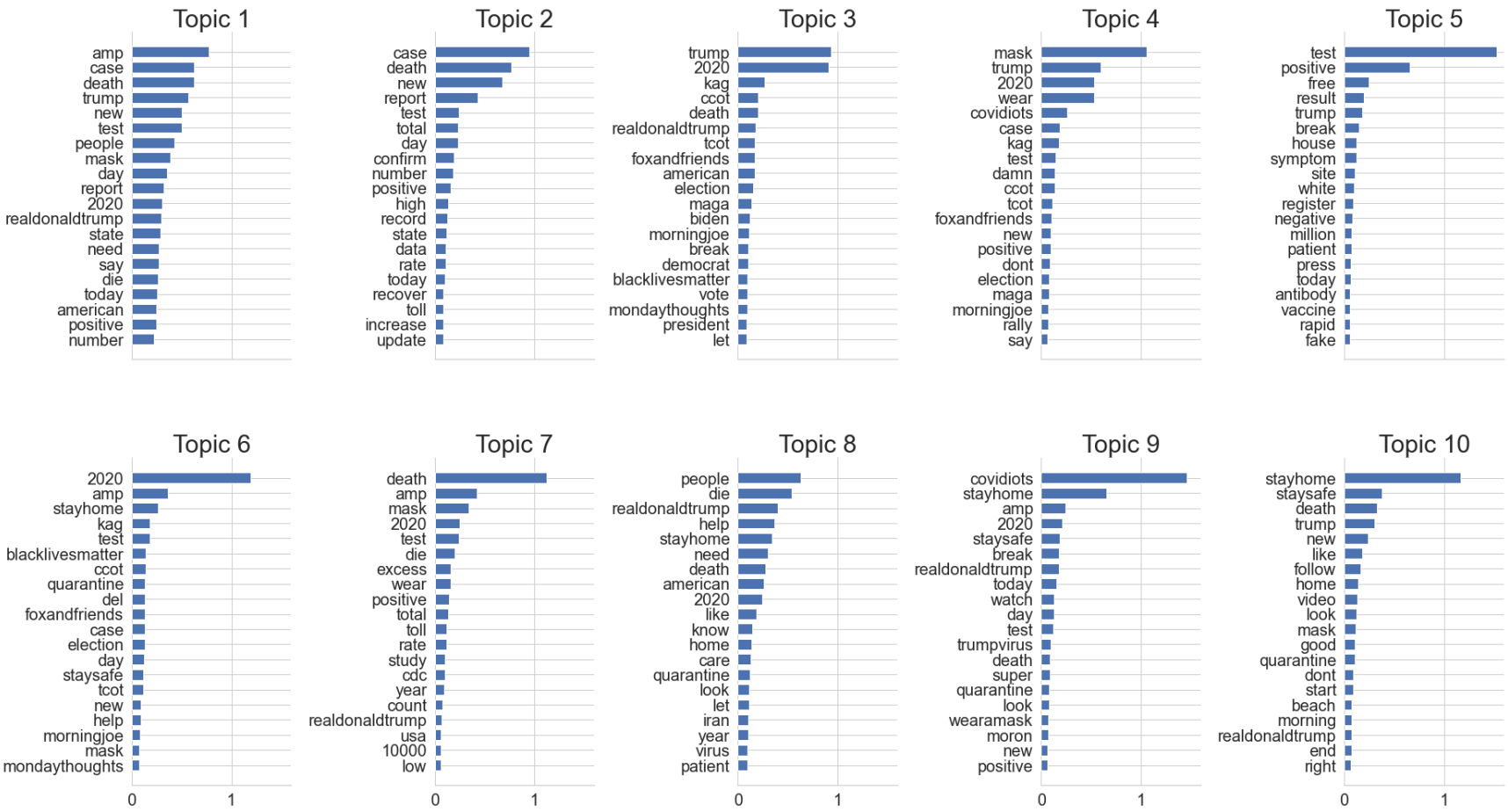
```
plot_top_words(
    nmf, tfidf_feature_names, n_top_words, "Topics in NMF model (Frobenius norm)"
)

# Fit the NMF model
print(
    "\n" * 2,
    "Fitting the NMF model (generalized Kullback-Leibler "
    "divergence) with tf-idf features, n_samples=%d and n_features=%d..."
    % (n_samples, n_features),
)
t0 = time()
nmf = NMF(
    n_components=n_components,
    random_state=1,
    init=init,
    beta_loss="kullback-leibler",
    solver="mu",
    max_iter=1000,
    alpha_W=0.005,
    alpha_H=0.005,
    l1_ratio=0.5,
).fit(tfidf)
print("done in %0.3fs." % (time() - t0))

tfidf_feature_names = tfidf_vectorizer.get_feature_names_out()
plot_top_words(
    nmf,
    tfidf_feature_names,
    n_top_words,
    "Topics in NMF model (generalized Kullback-Leibler divergence)",
)

# Fit the MiniBatchNMF model
print(
    "\n" * 2,
    "Fitting the MiniBatchNMF model (Frobenius norm) with tf-idf "
    "features, n_samples=%d and n_features=%d, batch_size=%d..."
    % (n_samples, n_features, batch_size),
)
t0 = time()
mbnmf = MiniBatchNMF(
    n_components=n_components,
    random_state=1,
    batch_size=batch_size,
```

```
    init=init,
    beta_loss="frobenius",
    alpha_W=0.005,
    alpha_H=0.005,
    l1_ratio=0.5,
).fit(tfidf)
print("done in %0.3fs." % (time() - t0))


tfidf_feature_names = tfidf_vectorizer.get_feature_names_out()
plot_top_words(
    mbnmf,
    tfidf_feature_names,
    n_top_words,
    "Topics in MiniBatchNMF model (Frobenius norm)",
)

# Fit the MiniBatchNMF model
print(
    "\n" * 2,
    "Fitting the MiniBatchNMF model (generalized Kullback-Leibler "
    "divergence) with tf-idf features, n_samples=%d and n_features=%d, "
    "batch_size=%d..." % (n_samples, n_features, batch_size),
)
t0 = time()
mbnmf = MiniBatchNMF(
    n_components=n_components,
    random_state=1,
    batch_size=batch_size,
    init=init,
    beta_loss="kullback-leibler",
    alpha_W=0.00005,
    alpha_H=0.00005,
    l1_ratio=0.5,
).fit(tfidf)
print("done in %0.3fs." % (time() - t0))

tfidf_feature_names = tfidf_vectorizer.get_feature_names_out()
plot_top_words(
    mbnmf,
    tfidf_feature_names,
    n_top_words,
    "Topics in MiniBatchNMF model (generalized Kullback-Leibler divergence)",
)
```

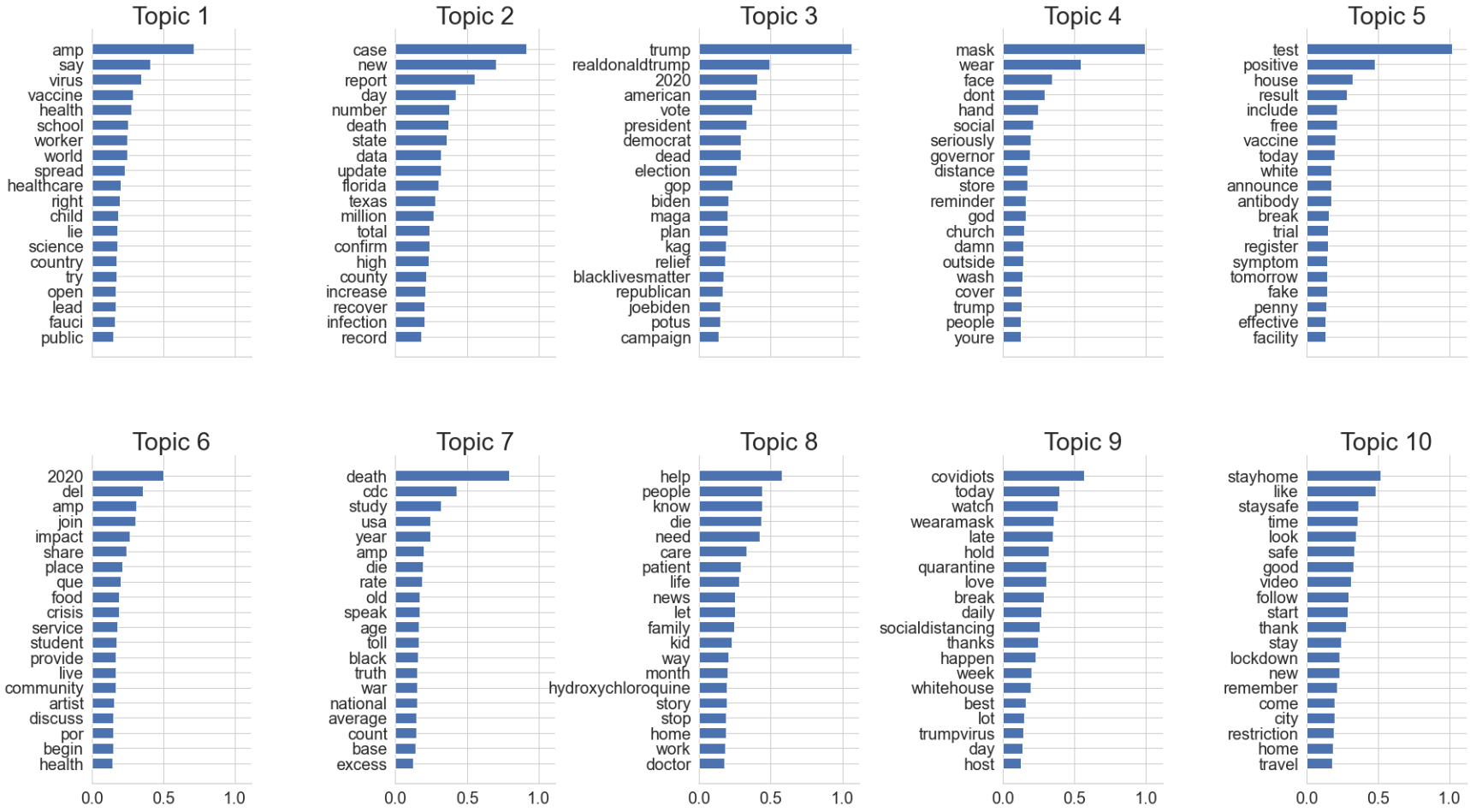
```
print(
    "\n" * 2,
    "Fitting LDA models with tf features, n_samples=%d and n_features=%d..."
    % (n_samples, n_features),
)
lda = LatentDirichletAllocation(
    n_components=n_components,
    max_iter=500,
    learning_method="online",
    learning_offset=0,
    random_state=0,
)
t0 = time()
lda.fit(tf)
print("done in %0.3fs." % (time() - t0))

tf_feature_names = tf_vectorizer.get_feature_names_out()
plot_top_words(lda, tf_feature_names, n_top_words, "Topics in LDA model")
```

```
Loading dataset...
done in 0.000s.
Extracting tf-idf features for NMF...
done in 0.040s.
Extracting tf features for LDA...
done in 0.031s.

Fitting the NMF model (Frobenius norm) with tf-idf features, n_samples=2000 and n_features=1000...
done in 0.095s.
```

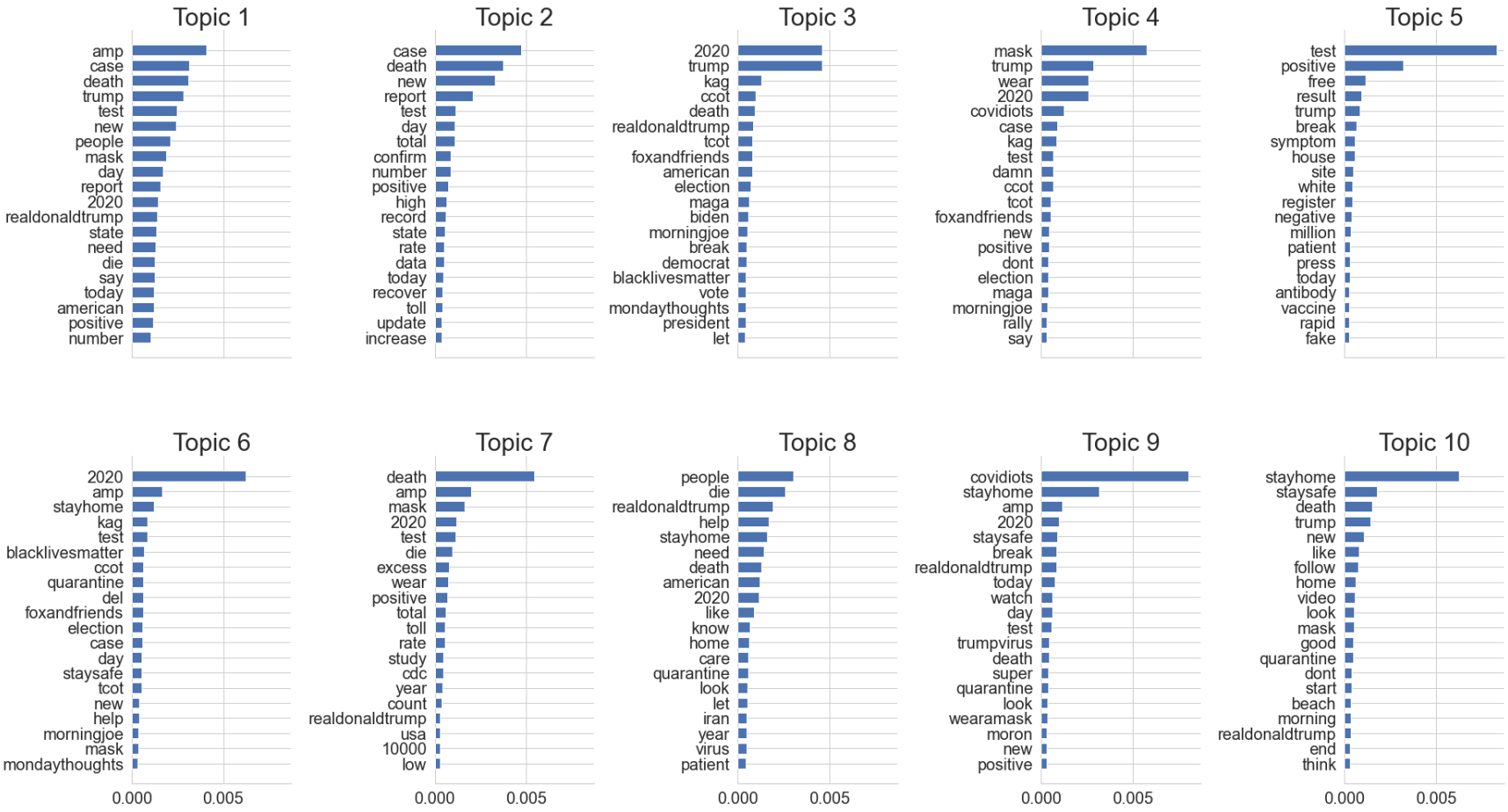
```
/Users/haileythanki/.local/lib/python3.9/site-packages/sklearn/decomposition/_nmf.py:1692: ConvergenceWarnin
g: Maximum number of iterations 200 reached. Increase it to improve convergence.
  warnings.warn(
```



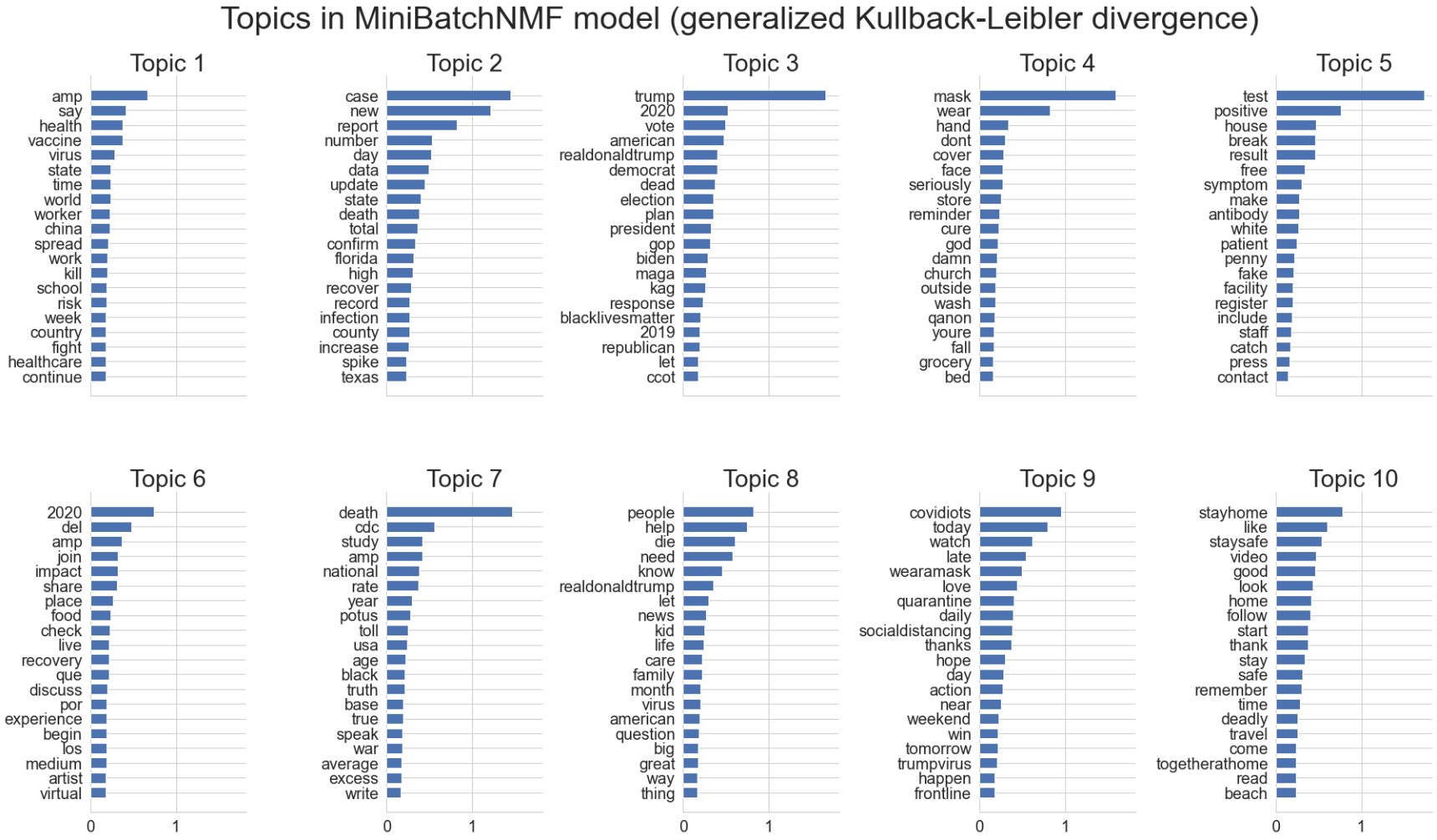

```
 Fitting the NMF model (generalized Kullback-Leibler divergence) with tf-idf features, n_samples=2000 and n_f
eatures=1000...
done in 0.395s.
```



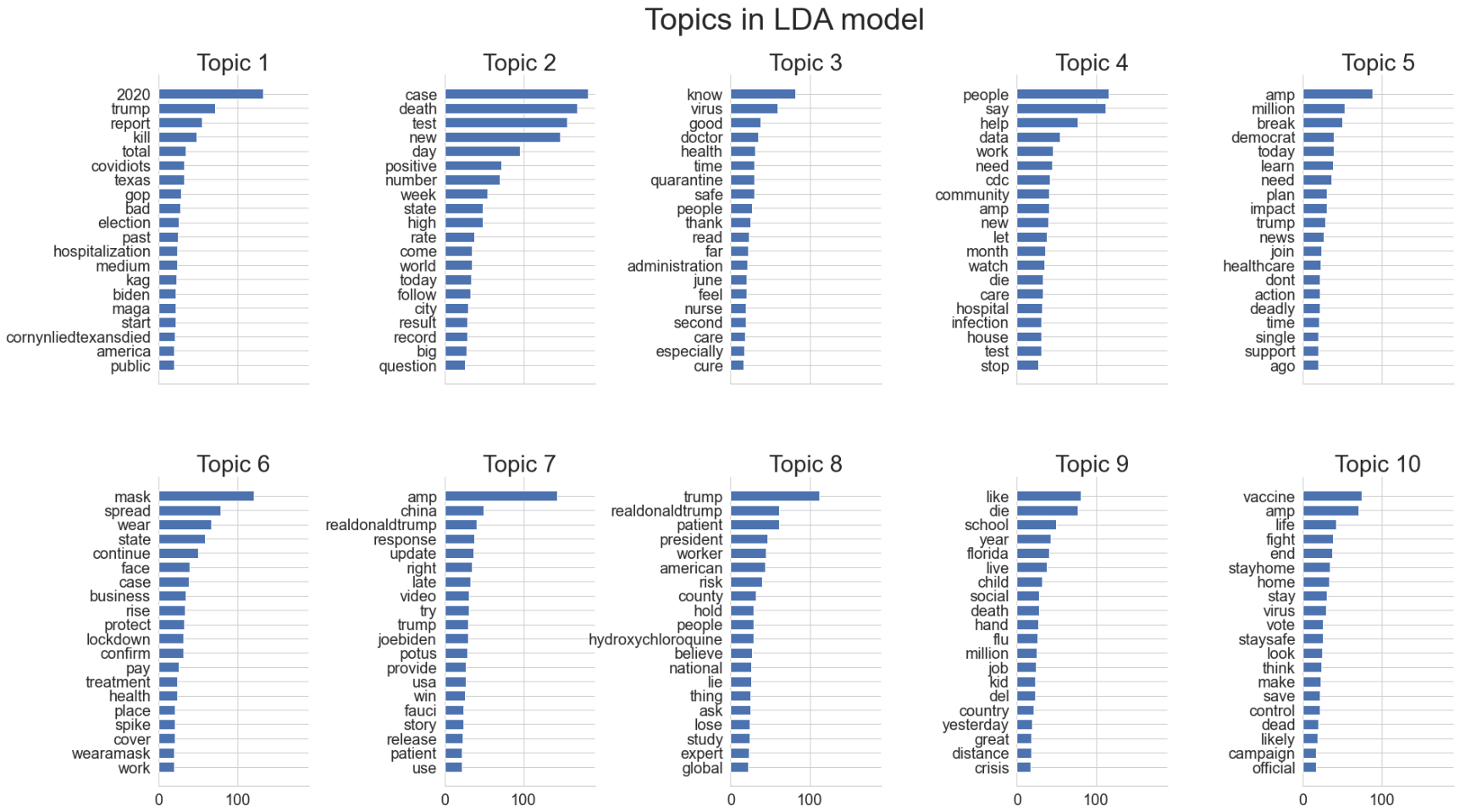

```
 Fitting the MiniBatchNMF model (Frobenius norm) with tf-idf features, n_samples=2000 and n_features=1000, ba
tch_size=128...
done in 0.077s.
```




```
 Fitting the MiniBatchNMF model (generalized Kullback-Leibler divergence) with tf-idf features, n_samples=200
0 and n_features=1000, batch_size=128...
done in 0.150s.
```




```
 Fitting LDA models with tf features, n_samples=2000 and n_features=1000...
done in 59.585s.
```



In [51]:
```
df_clean_lda.to_csv('df_with_lemmatized_tweets.csv', index=False)
```

Cosine similarity

In [52]:
```
df = pd.read_csv('df_with_lemmatized_tweets.csv')
df
```

```
/var/folders/_y/ch74wgzn7s1dxtq4ysb993sr0000gn/T/ipykernel_61326/547425379.py:1: DtypeWarning: Columns (2,3)
have mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv('df_with_lemmatized_tweets.csv')
```

Out[52]:

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_wo
0	Remember the #WuhanCoronaVirus? The pandemic w...	NaN	False	True	WuhanCoronaVirus KillerCuomo	us	267	
1	My sources @WhiteHouse say 2 tactics will be u...	NaN	False	True	Trump	us	281	
2	I'll venture a wild guess: If you were running...	NaN	False	True	COVID19	us	292	
3	#Pakistan (#GreenStimulus = #Nature protection...	NaN	False	True	Pakistan GreenStimulus Nature Green	us	236	
4	🇺🇸 Pandémie de #coronavirus: 30 pasteurs améri...	NaN	False	True	coronavirus COVID__19 COVID —19	us	279	
...	...	...	...	...	...	...	...	
239995	Aa Likes, Retweets yentra 🙏\n🔥🔥🔥 \n#Master	NaN	True	True	Master	new_zealand	39	
239996	Very interesting\nAny thoughts? \n\n#TheFive #T...	NaN	False	True	TheFive Trump2020 KAG2020 mondaythoughts COVID...	new_zealand	142	
239997	As we deal with #COVID19 don't forget that #Ch...	NaN	True	True	COVID19 Christians persecution Nigeria	new_zealand	307	
239998	While we hit 150,000 in #COVID19 deaths, the P...	NaN	False	True	COVID19	new_zealand	115	
239999	This too shall pass #Covid_19 . May remain sta...	NaN	False	True	Covid_19 HopeAlive	new_zealand	280	

240000 rows × 13 columns

In [53]:
```
df.hashtags.dtypes
```

Out[53]:
```
dtype('O')
```

In [54]:
```
df.hashtags[0].split(' ')
```

Out[54]:
```
['WuhanCoronaVirus', 'KillerCuomo']
```

In [55]:
```
# creating an array containing all hastags in the dataset

hashtag_arr = []

for hashtag in df.hashtags:
    hashtag_arr_temp = hashtag.split(' ')
    for hashtag_temp in hashtag_arr_temp:
        hashtag_arr.append(hashtag_temp)

hashtag_arr
```

Out[55]:
```
['WuhanCoronaVirus',
 'KillerCuomo',
 'Trump',
 'COVID19',
 'Pakistan',
 'GreenStimulus',
 'Nature',
 'Green',
 'coronavirus',
 'COVID__19',
 'COVID—19',
 'corona',
 'virus',
 'meme',
 'coronavirusmeme',
 'toilet',
 'paper',
 'coronapocalypse',
 'Coronavirus',
 'blacklifematters',
 'COVID19',
 'StopPoliceBrutality',
 'corona',
 'TheAcademy',
 'AcademyAwards',
 'Oscars',
 'TheOscars',
 'AMPAS',
 'FilmTwitter',
 'WuhanVirus',
 'art',
 'artists',
 'photography',
 'Photographer',
 'coronavirus',
 'Covid_19',
 'COVID19',
 'PAHouse',
 'COVID19',
 'COVID19',
 'COVID19',
 'COVID19',
 'BREAKING',
 'coronavirus',
 'VOTE',
```

```
'COVID19',
'Covid_19',
'NorthCarolina',
'Trump2020',
'KAG2020',
'SaturdayThoughts',
'Ccot',
'Trump',
'Hydroxychloroquine',
'coronavirus',
'Science',
'BrightSpot',
'kswx',
'COVID19',
'IAQ',
'sustainability',
'trane',
'COVID19',
'CoronaVirus',
'AnikaChebrolu',
'ShowMeYourMask',
'MaskUp',
'WeCanStopCOVIDTogether',
'ScienceMatters',
'disability',
'COVID19',
'COVID19',
'fuckchina',
'tyranny',
'coronavirus',
'ACB',
'coronavirus',
'COVID—19',
'CoronavirusPandemic',
'WearAMask',
'cruising',
'WakeUpAmerica',
'HerdImmunity',
'COVID19',
'BillGates',
'BigPharma',
'COVID19',
'Democratic',
'covid',
'dumptrump2020',
```

```
'CoronaVirus',
'Covid_19',
'masks',
'pandemic',
'EatAtHome',
'Quarantine',
'COVID19',
'Covid',
'COVID19',
'pandemic',
'MothersDay',
'Covid_19',
'TyphoidTrump',
'coronavirus',
'PMIS',
'coronavirus',
'Texas',
'coronavirus',
'coronavirus',
'FakeNews',
'PlugandPlay',
'ZprymeNow',
'suntv',
'news',
'sunnews',
'Chennaicorona',
'CoronaUpdatesInIndia',
'CoronavirusIndia',
'CoronaUpdates',
'corona',
'StaySafe',
'Edinburgh',
'FAIL',
'COVID19',
'China',
'covid19',
'COVID19',
'RemoveTrumpNow',
'House',
'coronavirus',
'expensive',
'Democrat',
'NewJersey',
'JaimeHarrison',
'StayAtHome',
```

```
'EarthDay',
'Resist',
'COVID19',
'coronavirus',
'ProjectLincoln',
'TrumpPandemic',
'UBISavesLives',
'StayHome',
'ImpeachedForLife',
'WearADamnMask',
'COVID19',
'COVID—19',
'MakingItHome',
'COVID19',
'pandemic',
'Iran',
'prepper',
'coronavirus',
'COVID19',
'COVID19',
'HEROESACT',
'StudentDebt',
'COVID19',
'StayHome',
'coronavirus',
'coronavirus',
'COVID',
'Halloween',
'Canceled',
'COVID19',
'COVID',
'Nacional',
'Coronavirus',
'COVID19',
'TogetherAtHome',
'COVIDIOTS',
'Polahaku',
'COVID19',
'GreedyGilead',
'COVID19',
'Remdesivir',
'COVID19',
'covid19',
'coronavirus',
'COVID19',
```

```
'Pandemic',
'COVID19',
'covid19',
'Holocaust',
'Coronavirus',
'Indigenous',
'TrumpHasCovid',
'TrumpHasCorona',
'ItIsWhatItIs',
'COVIDIOTS',
'TrumpVirus',
'COVID19',
'TrumpFailedAmerica',
'BidenHarrisToSaveAmerica',
'coronavirus',
'SocialDistancing',
'BalconyJam',
'StarSpangledBanner',
'RT',
'alexsmith',
'covid_19',
'dakprescott',
'nfl',
'sportstalk',
'Covid_19',
'NJ08',
'COVID19',
'COVID19',
'Covid19',
'COVID19',
'Política',
'Covid',
'Congreso',
'CasaBlanca',
'BreakingNews',
'CCPVirus',
'Covid19',
'coronavirusindia',
'Cal',
'Berkeley',
'ONLINE',
'HowTo',
'SEO',
'contentmarketing',
'Presidementia',
```

```
'COVID19',
'COVID—19',
'Covid19',
'Coronavirus',
'anorexia',
'COVID',
'covid19',
'cerls',
'COVID19',
'WuhanVirus',
'HowWeGotHere',
'Coronavirus',
'coronavirus',
'Covid19',
'Covid',
'COVID19',
'EVNews',
'COVID—19',
'FDA',
'Vacuna',
'COVID19',
'innovation',
'protection',
'Covid19',
'WearAMask',
'coronavirus',
'COVID19',
'NewYork',
'OneLove',
'OneHeart',
'StaySafe',
'staysane',
'coronavirus',
'pandemic',
'utility',
'COVID19',
'masks',
'UnitedKingdom',
'TheBatman',
'CoronavirusPandemic',
'COVID19',
'EastHarlem',
'AOC',
'Socialism',
'COVID19',
```

```
'COVID19',
'WearAMask',
'SleepingBeauty',
'COVID19',
'Masks',
'celebrate',
'coronavirus',
'eid',
'muslims',
'coronavirus',
'Daines',
'Trump',
'COVID19',
'COVID19',
'COVID19',
'COVID19',
'COVID19',
'coronavirus',
'Iran',
'Coronavirus',
'COVIDIOTS',
'Hydroxychloroquine',
'coronavirus',
'COVID19',
'BreakingNews',
'Maryland',
'COVID19',
'Coronavirus',
'COVID19',
'COVID19',
'COVID19',
'art',
'artists',
'photography',
'Photographer',
'COVID—19',
'Covid_19',
'FalsePositives',
'Covid19',
'coronavirus',
'COVID19',
'G20',
'covid19',
'TrumpCard',
'China',
```

```
'RedCross',
'COVID19',
'coronavirus',
'Giuliani',
'WakeUpAmerica',
'Trump',
'TrumpCrimeFamily',
'COVID19',
'COVID19',
'CDC',
'Covid19',
'OPENAMERICANOW',
'COVID19',
'COVID19',
'covid19',
'fqhc',
'Covid',
'SARSCoV2',
'disenfectant',
'Covid19',
'COVID19',
'SocialDistancing',
'Auschwitz',
'COVID—19',
'COVID—19',
'COVID19',
'COVID19',
'fishers',
'MoreFemaleWorldLeaders',
'VOTE',
'COVID19',
'China',
'COVID19',
'COVID19',
'TB',
'Coronavirus',
'COVID19',
'GOP',
'SCOTUS',
'facemasks',
'facemasks4all',
'gifts',
'shoppingonline',
'socialdistancing',
'coronavirus',
```

```
'virus',
'coronavirus',
'health',
'vaccine',
'jjencares',
'quarantine',
'coronavirus',
'Coronavirus',
'pandemic',
'COVID19',
'Pandemic',
'AngeloState',
'ramfam',
'COVID19',
'SoANJ',
'COVID19',
'NJ',
'JoeBiden',
'VoteHimOut',
'VoteBidenHarrisToSaveAmerica',
'WuhanCoronaVirus',
'life',
'ourworld',
'inspirationalquotes',
'future',
'wecandoit',
'startingover',
'inspirationalquotes',
'covid19',
'JoeBidenKamalaHarris2020',
'trump2020',
'biden',
'maga',
'berniesanders',
'election',
'COVID19',
'Covid19',
'COVID19',
'SaludTues',
'COVID19',
'COVID19',
'WednesdayMorning',
'ConspiracyTheories',
'Covid19',
'COVID19',
```

```
'Socialmedia',
'NBA',
'NBATwitter',
'COVID19',
'wedding',
'cats',
'dogs',
'cosplay',
'feet',
'COVID19',
'pandemic',
'OneMargarita',
'HickHopPops',
'REMIX',
'COVID_19',
'coronavirus',
'recovery',
'COVID',
'Trump',
'InThisTogether',
'StaySafe',
'maritime',
'shipping',
'covid19',
'coronavirus',
'new',
'COVID',
'earthquake',
'Coronavirus',
'Spotifyplaylists',
'ONEOKROCK',
'stayhome',
'video',
'branding',
'stayhome',
'GodIsGood',
'COVID19',
'journalism',
'EnjoyStayHome',
'Thursday',
'NorthCarolina',
'PeopleHelpingPeople',
'EVNews',
'FederalAgents',
'COVID19',
```

```
 's',
 'TogetherAtHome',
 'COVID19',
 'FactsNotFear',
 'COVID19',
 'WhiteHouseVirus',
 'TrumpIsANationalDisgrace',
 'COVID19',
 'COVIDIOT',
 'COVID19',
 'COVID19',
 'AHRQ',
 'COVID',
 'COVID19',
 'CovidDon',
 'DemocRat',
 'HydroxychloroquineDenier',
 'COVID19',
 'ACA',
 'COVID19',
 'polio',
 'vaccines',
 'RSV',
 'COVID19',
 'covid19',
 'COVID19',
 'BasicIncome',
 'COVID19',
 'CoronavirusPandemic',
 'MemorialDayWeekend',
 'coronavirus',
 'Wuhan',
 'Google',
 'COVID19',
 'AmazonPrime',
 'COVID19',
 'coronavirus',
 'Coronavirus',
 'NowPlaying',
 'Covid_19',
 'WritingCommunity',
 'StaySafe',
 'Chazocrats',
 'TheFive',
 'Trump',
```

```
'KAG',
'Tcot',
'Ccot',
'WednesdayWisdom',
'coronavirus',
'TheStory',
'Tucker',
'COVID19',
'BREAKING',
'TogetherAtHome',
'COVID19',
'MAGA2020',
'WWG1WGA',
'WWG1WGA_WORLDWIDE',
'WeMatter',
'SoTired',
'WeHaveRights',
'MyGrampsDidNotDieForThis',
'coronavirus',
'seniors',
'COVID19',
'ICASSP2020',
'COVID19',
'weed',
'mask',
'coronavirus',
'POTUS45',
'COVID19',
'Grief',
'TrumpIsANationalDisgrace',
'OneWorldTogetherAtHome',
'COVID19',
'USPS',
'Election2020',
'coronavirus',
'GeorgeFloyd',
'NAACP',
'ACLU',
'MSNBC',
'CNN',
'RayshardBrooks',
'GoNavy',
'IlhanOmar',
'COVID19',
'covid19',
```

```
'COVID',
'Michigan',
'Minnesota',
'Trump',
'MAGA',
'PROTEST',
'Covid_19',
'Coronavirus',
'lockdown',
'airlines',
'Texas',
'Florida',
'coronavirus',
'trump',
'healthgis',
'covid19',
'COVID19',
'COVID19',
'COVID19',
'MurderHornets',
'COVID19',
'FoxandFriends',
'Trump2020',
'KAG2020',
'Tcot',
'Ccot',
'COVID19',
'COVID19',
'edchat',
'pension',
'healthcareworker',
'COVID19',
'WearAMask',
'Covid',
'COVID19',
'MyBodyMyChoice',
'WHO',
'NWO',
'COVID19',
'Lost',
'COVID19',
'COVID19',
'Covid',
'Cancer',
'COVID19',
```

```
'SmartNews',
'COVID19',
'coronavirus',
'COVID—19',
'COVID_19',
'COVID',
'Pandemic',
'Whole',
'World',
'COVID19',
'ClimateChange',
'Kentucky',
'COVID—19',
'WearAMask',
'StayAtHome',
'coronavirus',
'coronavirus',
'TrumpIsNotWell',
'MondayMotivaton',
'Covid_19',
'pandemic',
'covid',
'COVID19',
'COVID19',
'COVID19',
'Pfizer',
'SecondWave',
'COVID',
'PANDEMIC',
'COVID19',
'COVID19',
'COVID19',
'MustRead',
'covid',
'COVID—19',
'COVID19',
'COVID19',
'COVID19',
'Tokyo',
'COVID19',
'virus',
'pigs',
'Coronavirus',
'fluvaccine',
'flu',
```

```
'COVID19',
'Washington',
'Cuba',
'COVID19',
'COVID19',
'BREAKING',
'Covid19',
'vaccine',
'COVID19',
'COVID19',
'coronavirus',
'coronavirus',
'Scared',
'Virus',
'SurvivalRate',
'Junkscience',
'coronavirus',
'COVID19',
'socialdistancing',
'patients',
'doctorsoffices',
'SARSCoV2',
'COVID19',
'StayHome',
'Covid_19',
'Pennsylvania',
'COVID19',
'WHIS',
'SDGCities',
'COVID19',
'WuhanLab',
'FASCISTBOOK',
'Facebook',
'COVID19',
'US',
'Hydroxychloroquine',
'COVID19',
'COVID19',
'coronavirus',
'Highway1Dreaming',
'VisitLosOsosBaywood',
'beachdog',
'StaySafe',
'RememberInNovember',
'FoxAndFriends',
```

```
'MorningJoe',
'Ccot',
'Walkaway',
'COVID19',
'KAG2020',
'TuesdayThoughts',
'MAGA',
'COVID—19',
'PandemicOutbreak',
'FoodInsecurity',
'COVID19',
'Arizona',
'Q',
'Qanon',
'TheGreatAwakening',
'WWG1WGA',
'WeAreTheNewsNow',
'Trump2020Landslide',
'DarkToLight',
'FullDisclosure',
'coronavirus',
'medtwitter',
'covid19',
'CoronaVirusUSA',
'CoronaVirusUpdates',
'COVID19',
'GOPBuiltThis',
'TraitorsToAmerica',
'Covid19',
'covid19',
'COVID19',
'Covid_19',
'COVIDIOTS',
'Arizona',
'COVID',
'coronavirus',
'COVID19',
'Empathy',
'TrumpOwnsEveryDeath',
'GOPBetrayedAmerica',
'GOPGenocide',
'COVID19',
'coronavirus',
'facemask',
'COVID19',
```

```
'COVID19',
'SocialDistancing',
'covid',
'coronavirus',
'arizona',
'pandemic',
'Lectionary',
'YearA',
'Easter3',
'antisemitism',
'Acts2',
'1Peter',
'Jesus',
'COVID19',
'COVID19',
'dumptrump',
'COVID19',
'COVID19',
'COVID19',
'COVID19',
'coronavirus',
'COVID19',
'Covid_19',
'FridayVibes',
'MothersDay',
'Sale',
'WednesdayVibes',
'StayHome',
'Shop',
'COVID19',
'COVID19',
'bedgeek',
'selfpleasure',
'COVID19',
'ICYMI',
'Wuhan',
'coronavirus',
'Sunshine',
'Covid19Out',
'COVID19',
'coronavirus',
'Junkscience',
'coronavirus',
'COVID19',
'coronavirus',
```

```
'COVID19',
'Coronavirus',
'HIV',
'lymphoma',
'Coronavirus',
'daffyduck',
'tweety',
'bugsbunny',
'facemask',
'cats',
'kids',
'son',
'mom',
'covid19',
'wearamask',
'COVID19',
'COVID19',
'TrumpCovid',
'WhiteHouse',
'coronavirus',
'COVID',
'Hydroxychloroquine',
'TrumpIsLosing',
'TrumpIsALoser',
'COVID19',
'omaharally',
'Omahastranded',
'trump',
'StayHome',
'BeverlyHills',
'COVID19',
'California',
'polio',
'COVID19',
'coronavirus',
'StayHome',
'COVIDIOTS',
'ArmyFootball',
'AbileneChristian',
'COVID',
'BeatNavy',
'facemask',
'Illinois',
'JBPritzker',
'Coronavirus',
```

```
'COVID19',
'COVID19',
'Coronavirus',
'Floridacoronavirus',
'COVID19',
'StayHome',
'China',
'COVID19',
'coronavirus',
'US',
'Covid',
'COVID19',
'BehavioralEconomics',
'Trump',
'Coronavirus',
'COVID19',
'COVIDreliefIRS',
'COVIDRelieve',
'NancyPelosi',
'COVID19',
'covid',
'debate',
'COVID19',
'COVID19',
'coronavirus',
'Texas',
'America',
'MaskUp',
'Hunger',
'GlobalPrayerForHumanity',
'COVID19',
'NursesWeek',
'collaboration',
'COVID19',
'codeofvets',
'COVID19',
'TaiwanModel',
'COVID19',
'Taiwan',
'COVID',
'coronavirus',
'WhiteHouse',
'COVID19',
'coronavirus',
'pandemic',
```

```
'travelrestrictions',
'transportation',
'coronavirus',
'COVID19',
'China',
'COVID19',
'coronavirus',
'doh',
'COVID19',
'FlattenTheCurve',
'COVID19',
'Trump',
'coronavirus',
'COVID19',
'COVID19',
'Covid_19',
'Covid_19',
'Florida',
'COVID19',
'COVID19',
'CornynLiedTexansDied',
'COVID19',
'CornynLiedTexansDied',
'COVID19',
'CornynLiedTexansDied',
'COVID19',
'COVID19',
'COVID19',
'COVID19',
'coronavirus',
'coronavirus',
'senate',
'COVID19',
'COVID—19',
'COVID19',
'thoughts',
'Quarantine',
'loneliness',
'coronavirus',
'coronavirus',
'coronavirus',
'Trump',
'Americans',
'COVID19',
'BlackLivesMatter',
```

```
'MinneapolisRiot',
'riots',
'GeorgeFloyd',
'coronavirus',
'coronavirus',
'TrumpsJealousOfObama',
'lockdown',
'COVID19',
'Weather',
'Florida',
'coronavirus',
'Makcast',
'COVID',
'noxiousACB',
'healthcare',
'coronavirus',
'COVID19',
'MaskItOrCasket',
'BailoutHumans',
'Trump',
'COVID19',
'coronavirus',
'COVID19',
'COVID19',
'coronavirus',
'covid19',
'BillGates',
'CoronaVirus',
'COVID19',
'celebrate',
'coronavirus',
'eid',
'muslims',
'Coronavirus',
'Liberia',
'COVID19',
'COVID19',
'covidisahoax',
'Covidisover',
'CDCisCorrupted',
'COVID19',
'Puppies',
'COVID19',
'PANdemic2020',
'endlockdown',
```

```
'potus',
'COVID19',
'GOPBetrayedAmerica',
'Masks',
'OssoffForSenate',
'COVID19',
'CovidKills',
'Fauci',
'DrBright',
'OperationWarpSpeed',
'CoronaVirus',
'OPENUPAMERICANOW',
'endthelockdown',
'COVID',
'CDC',
'COVID19',
'Latino',
'unemployment',
'Latino',
'Professionals',
'SCOTUSNomination',
'COVID19',
'pumpkin',
'october',
'autumn',
'fall',
'friday',
'weekend',
'dentistry',
'StayHome',
'Withme',
'COVID19',
'chess',
'socialdistancing',
'covid19',
'coronavirus',
'mask',
'park',
'nyc',
'newyorkcity',
'newyork',
'sunnyside',
'COVID19',
'COVID19',
'coronavirus',
```

```
 'COVID19',
 'COVID19',
 'NancyPelosi',
 'HCQ',
 'GeorgiaRunoffs',
 'COVIDIOT',
 'covid19',
 'armedconflict',
 'Covid19',
 'R2P',
 ...]
```

In [56]:
```
# creating an array containing only the unique hashtags

unique_hastags = set(hashtag_arr)
unique_hastags
```

Out[56]:
```
{'familypreservation',
 'MushroomPowder',
 'BANG',
 'LordHoweIsland',
 'digitalart',
 'CIHR',
 'QuarantineChronicles',
 'GOPLies',
 'Blanquer',
 'pga',
 'patio',
 'TobaccoHarmReduction',
 'Qanons',
 'SuspiciousObserver',
 'noexcuses',
 'wildbirds',
 'EdBartlett',
 'ham',
 'canoncamera',
 'Gambling',
 'covidnurses',
 'makeup',
 'nvac',
 'BCSchoolCovidTracker',
 'TheRookie',
 'DigitalBanking',
 'Kirklees',
 'assad',
 'newbeginnings',
 'personas',
 'cizaseason',
 'icantbreathe',
 'Impeachment',
 'damage',
 'Tits',
 'InformedCommunities',
 'hairstylesforheroes',
 '808day',
 'DragRace',
 'infectadura',
 'PlayingADifferentGame',
 'Protestas',
 'physicianAssistants',
 'DementiaDon',
 'advicefys',
```

```
'SCGuard',
'northernireland',
'techno',
'UKHeatWave',
'Lanarkshire',
'HearFarrakhan',
'rotter',
'WorldNewsDay',
'DCA',
'SweetSixteen',
'TakeAction',
'Product',
'practitioners',
'CoronaVirusIreland',
'LocalPropertyTax',
'Wankers',
'Moodle',
'ReahChakraborty',
'Handmade',
'PEPFARWatch',
'20-May',
'livelifeoutside',
'élèves',
'BarunSobti',
'shannon',
'freeschoolmeals',
'USUAggies',
'aysounited',
'Deal',
'VoteBiden',
'CartwrightKing',
'sinnfein',
'Kz2',
'MDBs',
'Sars',
'authors',
'baobesity',
'Beyonce',
'Installations',
'vaccineTrials',
'7mei',
'prevent',
'MIL',
'conspiracytheorist',
'makelotsofmoney',
```

```
'RECOVER',
'lifelessons',
'tirta',
'ChineseCoronaVirus',
'Delaware',
'PCRTests',
'TrumpsDangerous',
'gkunion',
'innovazione',
'UnitedKingdom',
'HolidaysWithoutHunger',
'TheArtsMatter',
'migrant',
'Humanity',
'bunkerbabytrump',
'ComingTogether',
'LivePD',
'rent',
'wimmin',
'CeolVid',
'ballsbridge',
'COVID—19response',
'CaliforniaForTrump',
'youngones',
'marriedlife',
'USElections',
'bethesdamd',
'greenlanterncorps',
'BeerBods',
'EricGarcetti',
'TrumpIsAPos',
'ASCEND',
'maradona',
'CoVId19',
'pinkart',
'ProtecttheVulnerable',
'boardingschool',
'itme',
'CATS',
'PueblosIndígenas',
'stayafloat',
'norwichfood',
'PostalWorkersDay',
'mechanisms',
'Affinity',
```

```
'AspenMedical',
'Naturebasedsolutions',
'JonathanSwan',
'Ungleichheit',
'Ogiri',
'NoMandatoryCovidVaccine',
'RIPDaveGreenfield',
'Boat',
'AslansCountry',
'MentallyHealthySchools',
'Contagious',
'mehdi',
'BAPsychology',
'TrumpDementiaSyndrome',
'FakeNewsMediaClowns',
'NoBackToNormal',
'coronavirusbc',
'Día77',
'YourCouncilDay',
'teamoffivemillion',
'DrawOurHeroes',
'autoerotism',
'masques',
'CDETB',
'PEDOWOOD',
'foodallergies',
'redcarpet',
'BBBScam',
'oxygenconcentrators',
'JobSeeker',
'PNDHour',
'ByeDon2020',
'DANCE',
'longweekend',
'LoveSomerset',
'inspirational',
'BookstoreNews',
'Cartoons',
'youtubechannel',
'siegfriedroy',
'medstudenttwitter',
'WesternBorder',
'heatwaves',
'FullFact',
'KillerTrump',
```

```
'IndigenousDay',
'DollyParton',
'outsourcing',
'aussies',
'esg',
'Stressed',
'bedford',
'TrustLeaders',
'Kyiv',
'WomenInSport',
'FeedScarborough',
'NamingTheLost',
'kitchensoverchildcare',
'idiotwind',
'cyberrisk',
'ReadtheInsert',
'EnterpriseCentres',
'لبنان',
'houseArrest',
'SDGMoment',
'SystemChange',
'WomenInPolitics',
'OrangeManBad',
'Satanic',
'newtownards',
'BookBound2020',
'Infectadura',
'blackhumour',
'TerryWaite',
'RobinSwann',
'HGV',
'AmazonHolidayBoycott',
'HealthCareRationing',
'presidentelectbiden',
'Leduc',
'dimensionjump',
'SCAM',
'NewsBreakfast',
'CIV',
'coveryourface',
'cleanandsober',
'MentalHealthAwarenessWeek2020',
'shamednation',
'Gardaí',
'covid19hoax',
```

```
'Dallastx',
'TheArchers',
'CorporateManslaughter',
'tearareomāori',
'POLAND',
'globalwarming',
'Allity',
'Qom',
'dev',
'freedownload',
'PostalService',
'brandonrouth',
'supermario',
'GoodNightTwitterWorld',
'cyclosporine',
'MikePenceIsInfected',
'ITV',
'OUTNOW',
'Friends4Ever',
'BillNye',
'lockdownhaircut',
'Email',
'DiplomaciaDigitalRD',
'staycationinspiration',
'NayibBukele',
'BinanceCharity',
'TrumpCrimeSyndicate',
'WhiteMilk',
'CoachellaValley',
'WuhanCoronavirus',
'Floridian',
'breastreconstruction',
'OurOnlyFuture',
'aCareworkersTale',
'AutumnColours',
'Reach',
'CoronavirusBo',
'A14',
'Employeesafety',
'countermeasures',
'ourSIPTU',
'existentiality',
'TMRGUK',
'juicefast',
'oldbayseasoning',
```

```
'meddlesomebrewing',
'BlacklivesMaters',
'singlemen',
'onlyfanspages',
'bantha',
'Rock',
'Shocking',
'SavingLivesInLockdown',
'jerseyfresh',
'funnymemes',
'laval',
'BigScottishBookClub',
'walk',
'teacherhorizons',
'LiberateVirginia',
'Humanitarian',
'caronavirusoutbreak',
'ToiletDuck',
'ANTIVAXERS',
'PremierFord',
'journaal',
'4thofJulyWeekend',
'NengiXShoesByFlora',
'kildarelockdown',
'SAVIEZVOUS',
'Oadby',
'ARiEAL_Researchers',
'VERBORGENARMOEDENL',
'TrumpRallyNC',
'Independents',
'FakeBillionaireTrump',
'Postdoc',
'TheGReatAwakening',
'wealthwarriorfacts',
'screwedbytheRCN',
'Maroc',
'Marks',
'aircargo',
'Adderall',
'WeAreReady',
'HRBarometre',
'brum',
'ACPSEM',
'BlowOutSetUp',
'collaborating',
```

```
'saultnews',
'ESTHER',
'overproduce',
'COVIDAlertApp',
'PrayforGanja',
'ReopenNZ',
'CECRA',
'ChinWag',
'SomersetDay',
'HumpDay',
'OntarioTeachers',
'SuperRugby',
'NewVideo',
'genderparity',
'Ineptocracy',
'VidasNegrasImportam',
'seanad',
'wtpBlue',
'BarackObamaFutureInJeopardy',
'Scunthorpe',
'IranCovidTruth',
'TalbotCounty',
'ArizonaDiamondbacks',
'Ali_Younesi',
'Desantis',
'ThoughtAndPrayers',
'nolivegigsjokesahoy',
'ImmigrationÇaCompte',
'PCD11',
'fortuneteller',
'woulfe',
'BLAME',
'TourDeThanks',
'tired',
'Voted',
'SkyTourMovie',
'sexadvice',
'PopUp',
'cinema4D',
'CoronaWatch',
'UncleJoe',
'Caritastic',
'newmexico',
'modernslavery',
'SonyA7SIII',
```

```
'oursupport',
'MIRONEWS',
'UNO',
'Slovaquie',
'enter',
'madewithanimate',
'BreastScreen',
'封じ込め',
'plea',
'LeaveToRemain',
'ryanairrefunds',
'GoodMorning',
'klingee',
'Hoylake',
'birmingham',
'minutesilence',
'Speaker',
'sundayshred',
'TrumpSpeaderEvent',
'FridayNightSocial',
'ShowMeStrong',
'Populism',
'IWMD2020',
'HappyWednesday',
'ProtectOurCommunity',
'carersrock',
'NO2',
'paisdepandereta',
'新型コロナウイルス',
'zoos',
'ForgottenLordMayor',
'lml',
'runforheroes',
'PeerReview',
'STD',
'barrons',
'假账号',
'N95',
'webcoic',
'CausalWorkers',
'ecpobesity',
'noxiousACB',
'raredigitalart',
'LQvMS',
'sturgisrally',
```

```
'pornharms',
'demand',
'veteran',
'itsallaboutthewood',
'howtofixadrugscandal',
'DerryGirls',
'Americandream',
'michaeljackson',
'quarantineonlineparty',
'shoulder',
'SpaceAdventureCobra',
'InternationalCleanersDay',
'FEET',
'SwiftCurrent',
'AEP',
'amauk',
'PolicyDeciphered',
'BrianClement',
'covidisahoax',
'WeArePDX',
'Fatidabad',
'uniteforfreedom',
'playmoregolf',
'wudu',
'covid19MA',
'bnwphotography',
'PutTheAdsUoJoeBiden',
'MurdochMysteries',
'abq',
'buymyfirsthome',
'InvestigateBillGates',
'Double',
'Víctimas',
'Hypothetically',
'Capitalism',
'kits',
'OttawaGroup',
'taxhaven',
'talkRadio',
'POORmagazine',
'spinnersdancestudio',
'Howrah',
'StopLoiSecuriteGlobale',
'SpiceGirls',
'InternationalDayofOlderPersons',
```

```
'med',
'Wancocks',
'cesarean',
'dogsofinsta',
'FipAcademicLive',
'WeSpeechies',
'PoliticalPandemic',
'wearepharmacy',
'WearAMaskSaveAlife',
'Montevideo',
'tiktok',
'counties',
'CallOfDutyMobile',
'SQAResults',
'Direct',
'socialbubbles',
'turbo',
'AskListenDo',
'MPC',
'Cannabidiol',
'SmallStreamerCommunity',
'pat',
'ILD',
'VoteForBidenHarris',
'assessments',
'Cristobal',
'Balance',
'Nursinghomes',
'hotelquarantine',
'PortlandProud',
'ISEF',
'RFID',
'SP500',
'TrumpKillsAmericans',
'Greenlist',
'trumpgravedancer',
'DefundPolice',
'sobriety',
'拡散希望',
'WeAreSorryNengi',
'Archive',
'LOPEZTraidor',
'services',
'一日一Made',
'histoire',
```

```
'patientsafety',
'OwnerOperators',
'CreepyJoe',
'Colcorona',
'coronapocolypse',
'theirsacrificeisreal',
'MEcfs',
'BusinessImprovement',
'SkeemSaam',
'presidenttrump',
'BMJResearch',
'excludedNI',
'FalunGong',
'BPD',
'sharingthevision',
'JohnKobylt',
'keepingactive',
'MaskUpMichigan',
'Magasins',
'mRNA',
'ONPoli',
'Quantas',
'WYSD20',
'Proms',
'quarantinecats',
'healthliving',
'abetterworld',
'DanskeBankPrem',
'westandwiththesector',
'BackToSport',
'reason',
'eosinophilia',
'MadeInRwanda',
'bigass',
'Garston',
'gouvcan',
'EiffelTower',
'tub',
'ChinaJoe',
'letter',
'CombatVetsforTrump',
'ChooseWell',
'sonos',
'Guatemala',
'saddays',
```

```
'bigboobs',
'stayinghome',
'director',
'Conley',
'SackColbeck',
'sapsMP',
'jonesa',
'SexyArmpits',
'RightsandPromises',
'CherrySeaborn',
'communitystrong',
'totalharms',
'vauxhallcityfarm',
'TuesdayTip',
'UniteLR',
'onlyfansbabe',
'IndependentSAGE',
'here',
'Revelation',
'medlabtx',
'CanadaRecovery',
'globaldrugsurvey',
'modelo',
'AdriaTour',
'RandPaul',
'KenoshaRiot',
'StayTheFuckHome',
'puzzlelover',
'Branding',
'ArtistoftheSummer',
'NetSupportSchool',
'Busdriver',
'Foreverhome',
'NewsMax',
'marketingresearch',
'PoliticalLogics',
'quarantinehealth',
'scales',
'merengue',
'traitor',
'PsychStar',
'redbubbleartist',
'sick',
'Raleigh',
'frontlineheroes',
```

```
'Covid19SafeSystemofWork',
'ColonialHeights',
'tuesdayvibe',
'OHCA',
'VWFC',
'staged',
'ThinkSmarter',
'Burnaboy',
'RadiationPoisoning',
'fox5atl',
'COVIDVaccine',
'coal',
'APA',
'TakeCareOfEachOther',
'TablighiJamaat',
'bloggerstribe',
'treats',
'covid19rftlks',
'backtonormalsoon',
'StayHomeSavesLives',
'Facemask',
'PhoKingBon',
'Walkingthroughthecrisis',
'COVID19Vic',
'vakantieman',
'MJAInSight',
'NoTeDijeAdios',
'Qingdao',
'EndTheOrangMenace',
'putitinthebin',
'MillionsMAGAMarch',
'liquidità',
'SocialProtectionUG',
'everyonetogether',
'PayResearchPaper',
'literary',
'ZoomEvent',
'goslings',
'LekkiMassacre',
'WorldHandHygieneDay2020',
'parentengagement',
'olderpersons',
'aoda',
'MealPrep',
'aintsobad',
```

```
'weskus',
'WearAMaskSaveALife',
'LordMayor',
'Compere',
'stillcelebrating',
'noon',
'paphos',
'ショックドクトリン',
'blacklivesmatter',
'LNPfail',
'covid19science',
'DistanciamientoSocial',
'timesup',
'visual',
'Bucks',
'2hrsNonStopJamz',
'findgod',
'Lombardia',
'keepwarm',
'evidenceecosystem',
'DISINFECT',
'stayactive',
'caresact',
'AffordableHousing',
'Publicsphere',
'密',
'vtpoli',
'nswgov',
'disinfectant',
'postcards',
'Destiny',
'westvillage',
'Groningen',
'BorisHasFailedBritain',
'Past',
'VectorControl',
'tiocfaimidslán',
'SilviaPark',
'USMCA',
'heuteshow',
'interviews',
'DayCare',
'HavelockRoad',
'LawWeek',
'Psychotherapist',
```

```
'ChildLabor',
'ThrowHerOut',
'OpenVirginiaNow',
'eatplaychangeapp',
'antilockdown',
'leadership',
'RecallKenney',
'TDL10',
'Safety_Assessment',
'FrontLine',
'Revelations',
'Listing',
'ResistanceIsNotFutile',
'ProChoice',
'masksforCanada',
'Bundestag',
'ParksCanada',
'saveourjobs',
'GrifterBarbie',
'Society',
'FallRiver',
'SocialCreditSystem',
'UpbeatLockdown',
'westmidsjobs',
'dominicummimgs',
'LonConf20',
'lungdisease',
'HumanServicesRelief',
'atp',
'epsteindidntkillhimself',
'Health',
'VoteRedToSaveAmerica',
'21st',
'BackToWorkSafely',
'CAMHS',
'RelationshipsOverRestrictions',
'CoronavirusDisease',
'20thCenturyHoax',
'Iveson',
'hyperemesis',
'MusicHelps',
'economicsystem',
'bunningskaren',
'NoComplacency',
'maskme',
```

```
'WhateverLevel',
'REOPEN',
'WhileNobodyIsWatching',
'thedoctor',
'MailInBallotFraud',
'healthcareeducation',
'deanogorman',
'BNPL',
'tobeornottobe',
'saveireland',
'avemaria',
'dolphins',
'Riverside',
'gdp',
'APPGSEND',
'Netflix',
'eatwell',
'新加坡',
'ZozimusBar',
'openhousedublin',
'SETCovidFramework',
'kiosks',
'Boards',
'covidcon',
'Eton',
'sextpanther',
'wholesome',
'WarRoomTownHall',
'IWS',
'TorysOut',
'zine',
'coornavirus',
'Kansas',
'DatosCoronaVirus',
'keepgymsopen',
'glasseels',
'Candidates2020',
'tipperary',
'Cuteness',
'obesity',
'ThoughtLeadership',
'DefundCBC',
'cybersafe',
'dårskap',
'TheAthleteDevelopmentShow',
```

```
'StuckwithU',
'grad',
'BuenosDias',
'SelfReg',
'SLAMinutes',
'freeworkout',
'Netherlands',
'Flooding',
'morethanball',
'kilkennyWexfordCarlow',
'CrossroadsOfIdeas',
'USpolitics',
'IVERMIECTIN',
'BikerDown',
'GratitudeAttitude',
'ELINTNews',
'inkblot',
'AbsoFreakinLutely',
'mobilephone',
'fiji',
'haircutsformen',
'Ticats',
'loaf',
'Assassination',
'inducedanxiety',
'covid19tracing',
'clorox',
'trustees',
'MyWarner',
'NovaScotians',
'WednesdayFeeling',
'executiveorder',
'CoadyGrad',
'GCSEs',
'Nepal',
'School',
'EmilyMurphyGSA',
'CoVidiots',
'COVID—19idiots',
'ComPol',
'tlmep',
'TheDoseCBC',
'footfetısh',
'SmokingGun',
'AsianMurderHornets',
```

```
'LMI',
'ShutUpChallenge',
'Kępna',
'ProtectTheVulnerable',
'UKSPINE',
'Optometrist',
'PicOfTheDay',
'portlandmaine',
'swimmandgetsick',
'Museum30',
'NCAAD2',
'DGCIM',
'umf',
'Mossad',
'climatestrike',
'FurFreeBritain',
'allergy',
'U308',
'hypothyroidism',
'MCCQE2',
'questioneverything',
'BlackSupremacy',
'Naked',
'ERGS',
'confinamientoperimetral',
'BN',
'Baylor',
'TAGT',
'SJAPeople',
'DoneWithTrump',
'Assad_Genocide',
'GladysMustGo',
'TheScore',
'Cafés',
'PPI',
'البريميرليغ',
'happylaborday',
'CONTE',
'algebra',
'employeehealth',
'SaturdayNight',
'FPTP',
'Lemmings',
'ENDMMSABUSE',
'CovidMemorialDay',
```

```
'hurricane',
'岡村隆史',
'VEALive',
'flight',
'中共',
'ToryTouretteSyndrome',
'EndCOVIDForAll',
'autopsy',
'covid_19uk',
'COVIDCrisisWayWorseThanEver',
'foto',
'YourLife',
'ESSD',
'NaFianna',
'kungfuschoolshastings',
'WeeklyInvoiceTracker',
'trainthemedparty',
'roman',
'BusinessTransformation',
'CowardinChief',
'SocialDistanceHalloween',
'Lockdown',
'Lanark',
'SASGF',
'TWICE_AAA2020',
'WDoR2020',
'Campania',
'homeworkgap',
'Foodies',
'YemenCantWait',
'PLWNCDs',
'CloroxCURE',
'KC',
'NAI',
'freeze',
'heartless',
'Deutsche',
'ThalapathyVijay',
'charcoal',
'spaceX',
'EmmentalModel',
'PHHarriet',
'streetartist',
'wildlifephotography',
'WorldHeritage',
```

```
'Remdesvir',
'Outback',
'LovelySingh',
'instagramlive',
'SelfishTrump',
'AudigyMedical',
'rhyme',
'EmnaCharqui',
'onlinecourse',
'LongCovid',
'stalled',
'MensMentalHealth',
'Wig',
'Munster',
'travellers',
'theresistance',
'Genocidal',
'VSMPE',
'三浦春馬',
'Doggies',
'creatorsrespond',
'recession',
'SolidarityTracker',
'Sittingbourne',
'TurnOffMSM',
'Tyl',
'onlineretail',
'Buhera',
'TruthIsStrangerThatFiction',
'SecondNarrowsBridge',
'peppermintpatty',
'raisingtherate',
'staywell',
'CoronaVirusHOAX',
'CONICET',
'PsoProtectMe',
'ppploans',
'buyingcontent',
'spirituality',
'genomic',
'RHCP',
'BusesMoveAmerica',
'cybersec',
'CuomoIsAMurderer',
'santarossasoundbites',
```

```
'GiftTheCity',
'bjjgirls',
'AfricansInChina',
'Households',
'OosterhoffResign',
'Kefe',
'Purrsday',
'inspired',
'Verschwoerungstheorien',
'SanQuentin',
'StudMuffin',
'Saudicrimes',
'Ineedhelp',
'DonaldTrumpIsTheTypeOfGuy',
'confinementjour41',
'earthday',
'mfd53',
'indianfood',
'Trumpian',
'ottawatenants',
'WelshMuseums',
'TedrosResign',
'JFKJR',
'TrumpDerangementSyndrome',
'KidsInCrisis',
'PBS',
'λαθρομετανάστες',
'freegunfriday',
'EducationForAll',
'TheFeedZW',
'builtthatwall',
'powerlessness',
'ASHFromHome',
'eucoronavirus',
'bandanastyle',
'1a',
'ThinkSmart',
'GatesFoundation',
'AdaptationWeek',
'TiếngViệt',
'ImVotingfor',
'DoYourJob',
'SuperSpreaderSaturday',
'Droughts',
'président',
```

```
 'Teamsters',
 'NorthernSydney',
 'trainspotting',
 'DétecteurDeRumeurs',
 'FreeAccess',
 'WorldEconomicForum',
 'covd19',
 'concernlongjump',
 'totalrecall',
 'listentoledzeppelin',
 ...}
```

In [57]:
```
# Creating subsets of dataset for the us

df_us = df[df.country == 'us']
df_us
```

Out[57]:

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_word_c
0	Remember the #WuhanCoronaVirus? The pandemic w...	NaN	False	True	WuhanCoronaVirus KillerCuomo	us	267	
1	My sources @WhiteHouse say 2 tactics will be u...	NaN	False	True	Trump	us	281	
2	I'll venture a wild guess: If you were running...	NaN	False	True	COVID19	us	292	
3	#Pakistan (#GreenStimulus = #Nature protection...	NaN	False	True	Pakistan GreenStimulus Nature Green	us	236	
4	🇺🇸 Pandémie de #coronavirus: 30 pasteurs améri...	NaN	False	True	coronavirus COVID__19 COVID —19	us	279	
...	...	...	...	...	...	...	...	
39995	#FluKluxKlan with racist Gadsden flags #StayHo...	NaN	True	False	FluKluxKlan StayHomeSaveLives Reopen Covidiots...	us	124	
39996	Today's the day! @BlueAngels, @AFThunderbirds ...	NaN	False	True	COVID19	us	202	
39997	Contrary to predictions that #Covid19 economic...	NaN	True	False	Covid19	us	283	
39998	WATCH: @GovJanetMills leads Maine's #COVID19 r...	NaN	False	False	COVID19	us	257	

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_word_c
39999	2\n\n#Trump is turning America into a full-blo...	NaN	False	True	Trump dictatorship RemoveTrumpNow TrumpIsANati...	us	299	

40000 rows × 13 columns

In [58]:
```
# Creating a set of unique hashtags for the us

hashtag_arr_us = []

for hashtag in df_us.hashtags:
    hashtag_arr_temp = hashtag.split(' ')
    for hashtag_temp in hashtag_arr_temp:
        hashtag_arr_us.append(hashtag_temp)

unique_hashtags_us = set(hashtag_arr_us)
unique_hashtags_us
```

Out[58]:
```
{'smallbusinesses',
 'GilletsJaunes',
 'Newark',
 'digitalart',
 'QuarantineChronicles',
 'GOPLies',
 'meanmuggz',
 'workingfromhome',
 'COVID19Response',
 'Stupid',
 'HeathenInChief',
 'TroubleBreathing',
 'TrumpKnewVoteBlue',
 'Qanons',
 'SuspiciousObserver',
 'poll',
 'Resign',
 'netde',
 'EmergencyAssistance',
 'workplace',
 'Gambling',
 'covidnurses',
 'makeup',
 'OHP',
 '21May',
 'sunset',
 'givethedrummersome',
 'Innovation',
 'assad',
 'lps',
 'MartialLaw',
 'gospel',
 'Cannabis',
 'postapocalyptic',
 'newbeginnings',
 'KuruluşOsman',
 'Impeachment',
 'damage',
 'HomeHealthcare',
 'GeorgeGate',
 'DragRace',
 'fakedemic',
 'OpenUp',
 'hypocrites',
 'mystery',
```

```
'Protestas',
'KrackenReleased',
'physicianAssistants',
'FrançaisdelEtranger',
'PutinsPuppet',
'Cobra',
'wreathsofinstagram',
'TrumpDevastation',
'HearFarrakhan',
'nude',
'SweetSixteen',
'WISCONSINITES',
'SupportingFarmers',
'onlineclasses',
'LIES',
'Kolkata',
'2AñosDeLegalidad',
'cornavirus',
'트와이스',
'PEPFARWatch',
'livelifeoutside',
'allergyseason',
'bookpromtion',
'bananadaquiri',
'USUAggies',
'VTed',
'aysounited',
'Eagles',
'fuckfriend',
'VoteBiden',
'EllenDeGeneres',
'happymonday',
'sinnfein',
'Tweetorial',
'FelizMartes',
'covidhoax',
'authors',
'Tx',
'Milton',
'AmericaRising',
'Beyonce',
'vaccineTrials',
'prevent',
'AmericaFirstNoReally',
'Baby',
```

```
'HowTo',
'Delaware',
'TrumpsDangerous',
'UnitedKingdom',
'gnome',
'freckles',
'CPMGcovidsolutions',
'40Millionoutofwork',
'MaskOn',
'bunkerbabytrump',
'LivePD',
'CoronavirusIndia',
'Avoid',
'Chloroquine',
'rent',
'CaliforniaForTrump',
'covid1984',
'bethesdamd',
'IndiaFightsCoronavirus',
'freedom',
'freshprince',
'keithurban',
'videoviral',
'Calculus',
'prochoice',
'essaypay',
'navy',
'Socialism',
'mnmean',
'apnode',
'CoVId19',
'Husbands',
'Blacklivesmatter',
'pinkart',
'CATS',
'PueblosIndígenas',
'SaudiTrump',
'dayjobs',
'PhysicalDistancing',
'SNTEunoAuno',
'preteen',
'buddybench',
'FORSCOM',
'SND2020',
'COMPTON',
```

```
'25AmendmentNow',
'mtpol',
'Boston',
'mentalillness',
'TrumpDementiaSyndrome',
'websday',
'Nutrition',
'Minister',
'NDChallengeNetwork',
'NoBackToNormal',
'NationalSportsDay',
'RSV',
'GaryBarnett',
'Wyoming',
'IStandWithFauci',
'RIOT',
'funathome',
'TrumpForPrison2020',
'IWearAMask',
'foodallergies',
'CombatCovid',
'redcarpet',
'TrumpRiots',
'Hochschulen',
'OrangeShitGibbon',
'ByeDon2020',
'DANCE',
'LockHerUp',
'horseshoecrab',
'Lost',
'HappyMemorialDay',
'DrBirx',
'OpenNYNow',
'TheView',
'inspirational',
'LoveIsEssential',
'BookstoreNews',
'OUTLOUD',
'youtubechannel',
'siegfriedroy',
'OpenAmericaUp',
'cdnpoli',
'dad',
'antiviral',
'schoolreopening',
```

```
'ministry',
'TrumpKnewAndDidNothing',
'TrumpVirus',
'NegligentHomicide',
'FreeComicBookDay2020',
'PamelaTurner',
'ObamaGate2020',
'usaCoronavirus',
'POW',
'Stressed',
'covidproblems',
'ROTH',
'schools',
'NobleJennJenn',
'who',
'pasoapaso',
'CoronaVirusCorruption',
'Azithromycin',
'DarwinSelfSelectButton',
'NamingTheLost',
'BARS',
'ScienceClearAndVivid',
'RockCandy',
'HomeRule',
'stayathomeprotests',
'RLF100',
'BunkerBoy',
'COVID—19Testing',
'Muharam',
'okleg',
'DerangedDonald',
'OrangeManBad',
'LordOfTheRings',
'JonRappoport',
'PandemicsReport',
'Danbury',
'BAEKHYUN',
'Gapol',
'HealthCareRationing',
'DUHBOYGUD',
'tiktokdown',
'FI',
'SCAM',
'ACTogether',
'Immunity',
```

```
'blacklifematters',
'covid19hoax',
'Dallastx',
'ATS2020',
'ChildTrafficking',
'globalwarming',
'HCP',
'Qom',
'LAtraffic',
'BlueWave2020',
'supermario',
'GBED',
'procreate',
'Corona',
'Trumpchecks',
'Democracy',
'tweetiatrician',
'heyarnold',
'Custody',
'BidensRiots',
'died',
'ZimbabweanLivesMatter',
'September11',
'DrCarrieMadej',
'DiplomaciaDigitalRD',
'staycationinspiration',
'freedumb',
'TeamMaryland',
'ReadyAF',
'NepotismBarbie',
'CoachellaValley',
'Live',
'WuhanCoronavirus',
'IDWeek2020',
'Floridian',
'breastreconstruction',
'teamnewyork',
'SAVETHECHILDERN',
'ForMe',
'6FeetApart',
'protection',
'SEC',
'BUFFALO',
'Adrenochrome',
'BeVeryVeryAfraid',
```

```
'CovidScam',
'OpportunityZone',
'ftc',
'clothfacecoverings',
'partytime',
'Mentorship',
'goodtoknow',
'zoonoses',
'treason',
'art',
'ChildrensLivesMatter',
'BuenasNoticias',
'meddlesomebrewing',
'WhoWillUKnownThatDiesU',
'courtpacking',
'RonDeSantisIsAMonster',
'Ireland',
'StopCOVID19',
'Top5ThingsToKnowToday',
'bantha',
'Queen',
'kidneydisease',
'SavingLivesInLockdown',
'funnymemes',
'Vehicle',
'BostonGlobe',
'walk',
'GODFIRST',
'SkilledTrade',
'nurseappreciation',
'race',
'DemandAction',
'MAGICSTICK',
'野球女子',
'TrumpRallyNC',
'ubtruth',
'Independents',
'TRUMPMORON',
'Gofundme',
'getaway',
'LifeGoesOn',
'TheGReatAwakening',
'Maroc',
'BABYMETAL',
'Costco',
```

```
'Adderall',
'trends',
'STEMdaily',
'kidsincages',
'GISH',
'overproduce',
'FREE',
'PrayforGanja',
'WHO',
'OpenUpAmericaAgain',
'HSI',
'HumpDay',
'SuperRugby',
'ProtectEducation',
'nyclockdown',
'MondayAfternoon',
'nashvill',
'Cardiotwitter',
'GetToTheBunker',
'TheReidOut',
'TrumpPandemic',
'Congrats',
'worklife',
'NEWS',
'AlecBaldwin',
'BarackObamaFutureInJeopardy',
'BlindSheeple',
'TalbotCounty',
'TREAT',
'Mercy',
'Desantis',
'HR1044MustPassNow',
'Ellen',
'ThoughtAndPrayers',
'FAMILYCARE',
'Magdalena',
'Лавров',
'SputnikV',
'CupOfJoe',
'rocknroll',
'stress',
'Liban',
'NonEssential',
'Vienna',
'GreatAmericanBash',
```

```
'Brazil',
'modernslavery',
'GOPDollarsOverLives',
'enter',
'DemCreeper',
'TECHGIANTS',
'Salud',
'GoodMorning',
'klingee',
'CRISPR',
'Tadpoles',
'TrumpSpeaderEvent',
'ShowMeStrong',
'권영진',
'momsofinstagram',
'COpublichealth',
'follo4folloback',
'EndTimes',
'notreally',
'cardio',
'LPT',
'crooked',
'Mundschutzboykott',
'streaming',
'NoJudticeNoPeace',
'StayHome',
'biotechnews',
'fitness',
'FRIDAY',
'barrons',
'OneVoice1',
'SleepyJoeBiden',
'N95',
'Christians',
'savethetatas',
'noxiousACB',
'Europa',
'SorosFundedRiots',
'bookmarketing',
'Rebellion',
'psilocybin',
'shoulder',
'SpaceAdventureCobra',
'TrumpSuperSpreader',
'Hermosillo',
```

```
'Palestinian',
'FEET',
'BigTechControl',
'stayconnectedtogether',
'mytopfans',
'JAMA',
'challengeaccepted',
'ONLYFANS',
'ThankYouHealthCareWorkers',
'governorsassertyourdominance',
'covidisahoax',
'BBB',
'djing',
'theft',
'warriornunwednesday',
'uniteforfreedom',
'playmoregolf',
'covid19MA',
'TeamCanada',
'trumpers',
'玉森裕太',
'Cuba',
'halloween',
'VoteSaferSD',
'SafeToLearn',
'NursingHome',
'abq',
'MedicReSCardiology',
'death',
'MurdochMysteries',
'FiscalConservativism',
'donorforlife',
'Oakland',
'川尻蓮',
'TraumaInformedSchools',
'Capitalism',
'taxhaven',
'KeepYourChildHome',
'POORmagazine',
'giftED',
'gopconvention',
'MyHeroAcademia',
'pulmonaryrehab',
'dogsofinsta',
'orangutan',
```

```
'goforward',
'PoliticalPandemic',
'stephenamell',
'GreenTech',
'mother',
'crise',
'KAGA2020',
'tiktok',
'USED',
'filmmaking',
'counties',
'qrcode',
'turbo',
'ДНЯО',
'SupplyChain',
'CDC',
'Germany',
'Cuomosexual',
'SmallStreamerCommunity',
'ChooseLove',
'BuyAmerican',
'leaked',
'climatechange',
'hydroxychroloquine',
'smoothskin',
'behavior',
'WorldOceansDay',
'ExpandTheCourt',
'GiveawayAlert',
'Cristobal',
'hazleton',
'StrangerThings',
'Amazônia',
'tweety',
'JanAndolan',
'SpanishFlu',
'ISEF',
'RFID',
'SP500',
'RadicalRedistribution',
'Patreon',
'TrumpKillsAmericans',
'NNSA',
'trumpgravedancer',
'DefundPolice',
```

```
'sobriety',
'delta',
'PAYTHEM',
'federal',
'photographyonline',
'SuffolkAlerts',
'WednesdayMood',
'patientsafety',
'OwnerOperators',
'CreepyJoe',
'OCD',
'9News',
'coronapocolypse',
'fitfam',
'MEcfs',
'donating',
'OperationLegend',
'presidenttrump',
'Unhappiness',
'improves',
'Murphy',
'bestofbergen',
'Venezuela',
'holidayseason',
'Kirwanforum',
'EarthDay50',
'HatchAct',
'cantidad',
'Honduras',
'HurricaneDelta',
'Togetherwedeliver',
'联储会',
'fox5ny',
'MaskUpMichigan',
'safety',
'mRNA',
'Followback',
'nicetry',
'CoronavirusImpact',
'BankOfAmericaFailedME',
'RACISM',
'quarantinecats',
'TrueColors',
'BloodPressure',
'cooking',
```

```
'ArsTechnica',
'abetterworld',
'resisters',
'bakirkoy',
'youcandoit',
'murderer',
'Fenway',
'HOMEDRIVE',
'schoolsreopening',
'bigass',
'Seniors',
'Rochester',
'ChinaJoe',
'JoeKamala2020',
'ClintonFoundation',
'Okinawa',
'NIAW2020',
'Guatemala',
'USconsumers',
'bigboobs',
'stayinghome',
'CampWarnersBros',
'BenCarson',
'NEWSMEDIAISTHEVIRUS',
'SexyArmpits',
'onlineshopping',
'QPOST',
'likeforlikes',
'technocracy',
'COVID19outbreak',
'totalharms',
'UniteLR',
'onlyfansbabe',
'streetscene',
'CoronaDidntKillIItself',
'here',
'BlackFridayAmazon',
'china',
'CVS',
'usrc',
'CorruptTrump',
'dollar',
'bigdickgang',
'SorosGate',
'RandPaul',
```

```
'KenoshaRiot',
'StayTheFuckHome',
'SIDS',
'Whistleblowers',
'ps4',
'lmao',
'DeadBody',
'xXX',
'govegan',
'MaskUpMonday',
'biocytogen',
'KamalaHarrisForVP',
'QT2020',
'06strong',
'NationalHospitalWeek',
'vitamins',
'CoronaVirusHoax',
'Weird',
'scales',
'TodaysMedicalUpdate',
'CompassionateRelease4Reality',
'b2b',
'sick',
'NYCTestandTrace',
'Chazocrats',
'PoliceAreNecessary',
'BTS',
'AntiVaccine',
'tuesdayvibe',
'WednesdayThoughts',
'memorialday',
'OHCA',
'bahcesehir',
'staged',
'WildlifeWednesday',
'pigs',
'MAGAt',
'Nurses4HIT',
'RecallWhitmer',
'fox5atl',
'volusiacounty',
'COVIDVaccine',
'Liarinchief',
'paxex',
'TablighiJamaat',
```

```
'Facemask',
'ObamaWasBetterAtEverything',
'EltonJohn',
'COVID19Vic',
'FireFauciNow',
'mycousincalledfromjail',
'Americans',
'Qingdao',
'CO',
'YourFutureIsAtYourFingertips',
'MillionsMAGAMarch',
'Tests',
'Noticias',
'SocialProtectionUG',
'PayResearchPaper',
'wearing',
'Author',
'goslings',
'brainfog',
'riporganizedcrime',
'southerncolumbia',
'CancelStudentDebt',
'EducationCannotWait',
'WearAMaskSaveALife',
'bigsarge',
'noon',
'ReleaseArnabNow',
'PTSD',
'hydroxycholoroquine',
'blacklivesmatter',
'martinpradenasviolador',
'fibrofog',
'SexySaturday',
'Bucks',
'findgod',
'avcilar',
'DISINFECT',
'DCHHS',
'caresact',
'hazukihairbrooklyn',
'Publicsphere',
'myinvisiblems',
'FinishtheThread',
'defcon28',
'disinfectant',
```

```
'MewGulf',
'Ahmedabad',
'cryptocurrency',
'howto',
'Telangana',
'VaccinesWork',
'Pediatric',
'SARSCOV2',
'westvillage',
'NannyState',
'FlipTheSenateBlue',
'Maskup',
'Jobloss',
'BreathingExercises',
'stream',
'KHive',
'BookReview',
'TheCure',
'boycott',
'PostIntensiveCareSyndrome',
'antivirus',
'Past',
'LiberalHypocrisy',
'chalk',
'rtj4',
'Cops',
'MarxeFaculty',
'QuikLab',
'victorberrios',
'BurialpittsNYC',
'USMCA',
'forestpark',
'ReOpenOregon',
'coroanvirus',
'friends',
'poshmark',
'CoronaTNUpdate',
'DayCare',
'Disparities',
'Togo',
'ML',
'OpenVirginiaNow',
'tattooart',
'firearm',
'steal',
```

```
'leadership',
'HealthIT',
'SuperNasty',
'whgfkeepitmoving',
'MakingCountermeasures',
'KanyeWest',
'Revelations',
'Folllow',
'influenza',
'usaf',
'Nacionales',
'JusticeForGloriaBambo',
'Resource',
'flu',
'ocmd',
'PayItForward',
'MJ',
'FallRiver',
'CenterforDiseaseControl',
'Tip',
'HumanServicesRelief',
'mcm',
'Sick',
'BeatCorona',
'karm',
'Southwest',
'Health',
'VoteRedToSaveAmerica',
'BackToWorkSafely',
'Alachua',
'MondayMotivaton',
'NeverKamala',
'snapchat',
'Disease',
'Biohazard',
'LGBTQI',
'drkennethbenjaminhughes',
'HonorOurFallen',
'momlife',
'Marijuana',
'GreenLightNY',
'Covid19Out',
'PPEworks',
'VirusCure',
'Air7HD',
```

```
'slide',
'antisocial',
'BNPL',
'Buckeye',
'video_conferencing',
'machinelearning',
'MedEd',
'dolphins',
'PEDOGATEISREAL',
'healthyfood',
'AllinForOhio',
'Cruise',
'Netflix',
'新加坡',
'ootd',
'laflorida',
'Conversatorios',
'BloodAndWater',
'Fall2020',
'NMSDC',
'EvaIllouz',
'hair',
'camgirl',
'beautiful',
'coronaupdatesindia',
'ImranKhan',
'MasksOff',
'DrainIt',
'Gratification',
'NotTrumpFirst',
'CuomoKilledSeniors',
'ASIA',
'Kansas',
'agenda21',
'Devil',
'DatosCoronaVirus',
'KARCHER',
'coornavirus',
'Cuteness',
'obesity',
'AGrooveAlmostEverydayWillPlayTillThePandemicGoAway',
'eu',
'legalimmigrant',
'fuckingmachines',
'rockandroll',
```

```
'morethanball',
'WeDeyKnack',
'CrossroadsOfIdeas',
'montgomerycountymd',
'Books',
'ReopenCT',
'TrumpVirusNov3TheresACure',
'inkblot',
'OER',
'TuesdayThought',
'NWIowa',
'Universities',
'Biotechnology',
'AnybodyButTrump',
'TogetheratHome',
'RMFansEnCasa',
'DowntownCincinnati',
'trumpScience',
'Window',
'CoronaVirusUK',
'covid19tracing',
'from',
'clorox',
'culture2020',
'bethechange',
'MyopiaAward',
'bidenharris2020',
'WednesdayFeeling',
'CovidVic',
'GCSEs',
'spreaderinchief',
'BidenCabinet',
'GOPhypocrisy',
'Nepal',
'BoycottChineseProducts',
'TheBuckSextonShow',
'MDCounties',
'School',
'letTheDoctorsSpeak',
'EmilyMurphyGSA',
'NosesOn',
'AMNET',
'Chanukah',
'CovidTesting',
'ComPol',
```

```
'COVID19SA',
'無限大',
'LGBTQPride',
'AsianMurderHornets',
'ProtectTheVulnerable',
'فيريرا_جروس_يامرتضي',
'Leftist',
'personaldata',
'NCAAD2',
'Niger',
'climatestrike',
'LysolTrump',
'Recycle',
'Wheaties',
'allergy',
'Football',
'lalege',
'Fiverr',
'Attack',
'GODBLESS',
'Tudor',
'Naked',
'GUNviolence',
'fyp',
'CoronaDeath',
'TAGT',
'oso',
'PPI',
'البريميرليغ',
'chase',
'SaturdayNight',
'UMNProud',
'NaturePhotography',
'hairdressers',
'hurricane',
'flight',
'QARMY',
'中共',
'Dashboard',
'Biological',
'OssoffForSenate',
'beachdog',
'COVIDCrisisWayWorseThanEver',
'princegeorgescounty',
'CEUs',
```

```
'Hannity',
'yogalife',
'Borat2',
'MaskUpOmaha',
'IvankasCoffins',
'NoMaskNoEntry',
'Transplantation',
'LGBTQ',
'Lockdown',
'SASGF',
'WhatAMoron',
'homeworkgap',
'OurRemedyIsARMY',
'4Oct',
'Berlin',
'BidenCares',
'HWRA',
'Foodies',
'YourBusiness',
'HopeInUSA',
'KC',
'openamericanow',
'BigBrotherGR',
'SantaBarbaraCounty',
'ICAFarmville',
'Supersite',
'KaranJohar',
'shadowban',
'Search',
'Deutsche',
'Brevard',
'spaceX',
'instantpot',
'wildlifephotography',
'LockHimUp',
'Remdesvir',
'Parcels2Pack',
'IA',
'CupofJoe',
'ArmyFootball',
'theatre',
'BasicIncome',
'flights',
'SMEs',
'LongCovid',
```

```
'WheresOurBailout',
'WakeUpAmerica',
'Manafort',
'PreExistingCondition',
'abigaildisney',
'VDI',
'HydroxychloroquineWorks',
'FlintWaterCrisis',
'travellers',
'theresistance',
'mhealth',
'healing',
'Civil',
'recession',
'TurnOffMSM',
'CountEveryLegalVote',
'Orders',
'LouisianaScholarsCollege',
'SecondNarrowsBridge',
'staywell',
'CoronaVirusHOAX',
'Granny',
'washingtondc',
'Millenials',
'ppploans',
'spirituality',
'Duke',
'Israeli',
'BusesMoveAmerica',
'Review',
'CuomoIsAMurderer',
'Puppies',
'leimertpark',
'TheCarNerdTalks',
'birthdaygreetings',
'FacesOfCoronavirus',
'SedgwickCounty',
'ecotype',
'Liberty',
'Trails',
'UnmaskingFlynn',
'Purrsday',
'localauthor',
'SanQuentin',
'friendstogetherinc',
```

```
 'StudMuffin',
 'akbatı',
 'airconditioners',
 'tuckercarlsontonight',
 'OBAMAGATE',
 'DonaldTrumpIsTheTypeOfGuy',
 'trail',
 'TrumpVirusDeathToll235K',
 'Spanish',
 'strike',
 ...}
```

In [59]:
```
# Creating subsets of dataset for the uk

df_uk = df[df.country == 'uk']
df_uk
```

Out[59]:

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_word_count	hashtags_
40000	Hardware has been more negatively impacted by ...	NaN	False	False	COVID19	uk	170	22	
40001	This week, hackers made headlines for targetin...	NaN	False	True	COVID19	uk	234	29	
40002	In the last week, refugee camps in #Syria and ...	NaN	False	True	Syria Greece COVID19	uk	235	33	
40003	Yes. You get to start your own for-profit #Cov...	NaN	False	True	Covid	uk	92	12	
40004	I've cared for somebody with #Covid_19 in my h...	NaN	False	True	Covid_19	uk	284	56	
...	...	...	...	...	...	...	...	...	...
79995	"We are more and more anxious. We can't go out...	NaN	False	True	CCPVirus	uk	278	40	
79996	Across the globe, #universities have been very...	NaN	False	True	universities COVID19	uk	304	34	

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_word_count	hashtags_
79997	What's new in #pharmacy and #healthcare? Check...	NaN	False	False	pharmacy healthcare	uk	173	21	
79998	Give back, Red Pill someone today and every da...	NaN	False	True	Trump	uk	230	42	
79999	It is time to consider your Xmas #heatingoil p...	NaN	False	True	heatingoil Covid19	uk	308	47	

40000 rows × 13 columns

In [60]:
```
# Creating a set of unique hashtags for the uk

hashtag_arr_uk = []

for hashtag in df_uk.hashtags:
    hashtag_arr_temp = hashtag.split(' ')
    for hashtag_temp in hashtag_arr_temp:
        hashtag_arr_uk.append(hashtag_temp)

unique_hashtags_uk = set(hashtag_arr_uk)
unique_hashtags_uk
```

Out[60]:
```
{'Create',
 'SRM',
 'familypreservation',
 'Islamophobia',
 'smallbusinesses',
 'STRIDE4localisation',
 'Blanquer',
 'QuarantineChronicles',
 'pga',
 'workingfromhome',
 'patio',
 'PANDAS',
 'Problem',
 'LancasterCountyPA',
 'supermarkets',
 'TrumpKnewVoteBlue',
 'Qanons',
 'ham',
 'LeavingNoOneBehind',
 'cvp709',
 'FSBConnect',
 'poll',
 'netde',
 'Transgender',
 'workplace',
 'PRrequest',
 '7maj',
 'EPicos',
 'CJIBobde',
 '21May',
 'sunset',
 'salemtogether',
 'Kirklees',
 'Innovation',
 'CCSEConversation',
 'AANadvocacy',
 'NHSPharmacies',
 'Cannabis',
 'postapocalyptic',
 'newbeginnings',
 'mortalidad',
 'Impeachment',
 'TechForGood',
 'ChiefRabbi',
 'icantbreathe',
```

```
'wecarewesupportweachieve',
'hairstylesforheroes',
'MigrantsMakeOurNHS',
'infectadura',
'InfectiousDiseases',
'CaptainTomMoore',
'Cabbies',
'XcomChimeraSquad',
'mystery',
'VE75',
'weekendreads',
'SCGuard',
'northernireland',
'studentvoice',
'techno',
'Cobra',
'Seagulls',
'Lanarkshire',
'PutinsPuppet',
'rotter',
'evaluate',
'nude',
'digitalcitizenship',
'SweetSixteen',
'TakeAction',
'Childline1098',
'notnormal',
'medieval',
'ReahChakraborty',
'CovidCon',
'Coops',
'Handmade',
'NCYT',
'트와이스',
'cornavirus',
'Drones',
'U2community',
'CallForCode',
'freeschoolmeals',
'CentralPark',
'NIHR',
'Warwick',
'VoteBiden',
'influencers',
'disley',
```

```
'happymonday',
'EllenDeGeneres',
'Tweetorial',
'NorthWales',
'Peckham',
'covidhoax',
'authors',
'MillionMAGAMarch',
'MSOA',
'Tx',
'curatorthoughts',
'7mei',
'FollowingThe',
'Baby',
'HowTo',
'yogaathome',
'Catwoman',
'domestic',
'TheDiplomat',
'tirta',
'SharkTank',
'RECOVER',
'Delaware',
'ForwardInUnity',
'innovazione',
'UnitedKingdom',
'HolidaysWithoutHunger',
'HonourBasedAbuse',
'Psychiatric',
'covidrestrictions',
'MaskOn',
'migrant',
'bunkerbabytrump',
'Motability',
'ghosts',
'rent',
'CoronavirusIndia',
'LivePD',
'Chloroquine',
'bmx',
'salmankhan',
'KeepInTouch',
'FBPE',
'WordPress',
'covid1984',
```

```
'USElections',
'freedom',
'ぬりえ',
'Rutte',
'BradParscale',
'terplove',
'suspendthetest',
'Socialism',
'AntilopenGang',
'CoVId19',
'WWIII',
'Blacklivesmatter',
'ecological',
'Zionists',
'Reconcile',
'ASCEND',
'stayafloat',
'tern',
'relaxation',
'SaudiTrump',
'ChurchillFellows',
'substanceusedisorders',
'EdenvilleDam',
'mechanisms',
'borisbaby',
'HMIEducators',
'epic',
'Synairgen',
'PhysicalDistancing',
'Affinity',
'onpageseo',
'7NEWS',
'creatively',
'Ungleichheit',
'TransformacionDigital',
'RecordingStudio',
'Boston',
'Opportunity',
'mentalillness',
'christmaspudding',
'DDoS',
'Nutrition',
'heartsforheros',
'StatisticalCoordination',
'Foto',
```

```
'Deranged',
'IStandWithFauci',
'ReleaseTheRussiaReport',
'funathome',
'WhackAMole',
'toptiptuesdays',
'masques',
'stigma',
'foodallergies',
'TrumpRiots',
'oxygenconcentrators',
'JobSeeker',
'OrangeShitGibbon',
'pctl',
'universalbasicincome',
'ByeDon2020',
'ConnectingScotland',
'ForgottenShielders',
'Tuesdaythought',
'routine',
'DrBirx',
'childnutrition2020',
'ToryRebels',
'usnavy',
'youtubechannel',
'booklove',
'cdnpoli',
'dad',
'schoolreopening',
'KillerTrump',
'1000names',
'TrumpKnewAndDidNothing',
'TrumpVirus',
'dominiccummigs',
'NegligentHomicide',
'Herefordshire',
'vinylcommunity',
'Scouts',
'corporatebonds',
'usaCoronavirus',
'covidproblems',
'GOT7_DYE',
'schools',
'texturehuntergatherer',
'bedford',
```

```
'Essaydue',
'TrustLeaders',
'who',
'thankateacherday',
'Azithromycin',
'SurreyHeath',
'festival',
'visitors',
'Kyiv',
'bushfire',
'ENECOVID',
'Stayathomechallenge',
'UnemploymentInsurance',
'NamingTheLost',
'3TierLockdown',
'CancelAnimalAg',
'NorthernPowerhouse',
'لبنان',
'BunkerBoy',
'ToxicShockSyndrome',
'SDGMoment',
'NHSVolunteerResponder',
'RunningOfTheBulls',
'Desdemona',
'WomenInPolitics',
'RehabLegend',
'DerangedDonald',
'reentry',
'DraftDodger',
'Satanic',
'PandemicsReport',
'BookBound2020',
'Flights',
'HGV',
'BAEKHYUN',
'Gapol',
'londonprotest',
'presidentelectbiden',
'LLC',
'Clare',
'tiktokdown',
'everydayisfriday',
'ACTogether',
'Immunity',
'cleanandsober',
```

```
'coveryourface',
'blacklifematters',
'TheArchers',
'CorporateManslaughter',
'ATS2020',
'IpswichHospital',
'BlueWave2020',
'dev',
'freedownload',
'UKTrendsAndFutures',
'Breezy',
'ISO27001',
'jobretentionscheme',
'MikePenceIsInfected',
'Corona',
'bbsendafriend',
'Democracy',
'tweetiatrician',
'Soundcloud',
'ToryBritain',
'ITV',
'OUTNOW',
'Kill4Fun',
'ZimbabweanLivesMatter',
'Uncategorized',
'September11',
'MAGAMillionMarch',
'arts',
'IGF',
'CulturalMarxism',
'WuhanCoronavirus',
'hometome',
'Ps5',
'IDWeek2020',
'covidview',
'Live',
'StAndrewsDay',
'LameDonaldDuck',
'mustfollow',
'allkeyshop',
'Soriano',
'protection',
'SEC',
'IBB',
'Adrenochrome',
```

```
'NAM',
'ruusu',
'CostaExpress',
'Backatcha',
'CovidTestingScandal',
'Bodoh',
'A14',
'ATO',
'countermeasures',
'island',
'aupair',
'HMICommunity',
'TMRGUK',
'councils',
'art',
'JusticeIsTheNextNormal',
'cellphone',
'westwoodfranklin',
'LaLaguna',
'VidasQuilombolasImportam',
'jagoslondon',
'meatball',
'singlemen',
'Ireland',
'DifferentFuturesWales',
'ACEP20',
'destress',
'HS2',
'OneLife',
'brexitshambles',
'Queen',
'Rock',
'LoveYourNeighbour',
'conman',
'Shocking',
'jerseyfresh',
'funnymemes',
'murderhornets',
'BostonGlobe',
'BukeleDictador',
'LuciferNetflix',
'walk',
'Humanitarian',
'ToiletDuck',
'journaal',
```

```
'4thofJulyWeekend',
'race',
'YULIN',
'alllivesmattered',
'Vaccines2020',
'Piglet',
'EdébatsCNB',
'Oadby',
'TPTB',
'Cardiff',
'wealthwarriorfacts',
'goodeats',
'Postdoc',
'BigBrotherNaijaLockdown2020',
'Trumpeo',
'trends',
'Costco',
'putyourmaskon',
'BAPRAS',
'Light',
'sostupid',
'brum',
'TTM',
'BlowOutSetUp',
'collaborating',
'Aerosol',
'FREE',
'WHO',
'CummingsOnTour',
'eidmubarak2020',
'LibrarySpaces',
'MOREandMORE',
'BovineBully',
'HEOR',
'lockdowner',
'EarlyOnline',
'HSI',
'NewVideo',
'Centralbanks',
'nyclockdown',
'FalseFacts',
'ShimmeringOcean',
'Ineptocracy',
'WNBC',
'Cardiotwitter',
```

```
'TheReidOut',
'ToryGenocide',
'TrumpPandemic',
'NBABubble',
'NEWS',
'AlecBaldwin',
'ShropshireBusiness',
'Scunthorpe',
'lifestyles',
'PresidenteDoBrasil',
'ArizonaDiamondbacks',
'miracles',
'atlas',
'PCD11',
'Presidental_Candidate',
'SputnikV',
'MyFavouriteThings',
'JacobReesMogg',
'ViernesDeAcción',
'ليلة_القدر',
'stress',
'CoronavirusPlaylist',
'WhereToBritain',
'FDARedTape',
'Breixt',
'Collaboration',
'Vienna',
'heatingoil',
'cinema4D',
'Buenosdías',
'Brazil',
'Hindi',
'Dail',
'abc730',
'UncleJoe',
'WineDelivery',
'modernslavery',
'SonyA7SIII',
'ReciprocalBuy',
'MIRONEWS',
'castlefield',
'enter',
'madewithanimate',
'Eurovision',
'spy',
```

```
'GoodMorning',
'halloweengifts',
'CRISPR',
'PoliticalScience',
'covidmaternity',
'minutesilence',
'birmingham',
'Hoylake',
'PublicSectorContracts',
'FridayNightSocial',
'TrumpSpeaderEvent',
'Speaker',
'Luciferian',
'internationalTravel',
'whatsappstokvel',
'NO2',
'paranormal',
'blackandred',
'zoos',
'cardio',
'新型コロナウイルス',
'UniversalCredit',
'ClemencyNow',
'legendlockdown',
'streaming',
'lml',
'StayHome',
'Shelters',
'university',
'runforheroes',
'UntitledAtelier',
'fitness',
'shopsmallbusiness',
'メキシカン料理',
'OneVoice1',
'N95',
'ecpobesity',
'Raffle',
'DeloresCahill',
'Europa',
'veteran',
'KRACHE',
'howtofixadrugscandal',
'quarantineonlineparty',
'Palestinian',
```

```
'KeepNCSafe',
'mytopfans',
'WorkSpaceSolutions',
'gettymuseum',
'ONLYFANS',
'AEP',
'Margate',
'PolicyDeciphered',
'BrianClement',
'WASHPAP2030',
'covid19MA',
'britishairways',
'Malton',
'CovidTaskForce',
'Cuba',
'halloween',
'NursingHome',
'SafeToLearn',
'bnwphotography',
'PeterHermann',
'death',
'familyfun',
'CollegeSports',
'NoEsBroma',
'flavourtown',
'kits',
'Capitalism',
'saftyfirst',
'LCHF',
'KeepYourChildHome',
'Survivability',
'talkRadio',
'spinnersdancestudio',
'DigitalToolbox',
'InternationalDayofOlderPersons',
'Ask_Spectrum',
'LaCienaga',
'veggie',
'cesarean',
'RCNNQN',
'offpageseo',
'PCCs',
'April17',
'wearepharmacy',
'DoingAWorldOfGood',
```

```
'mother',
'WearAMaskSaveAlife',
'tiktok',
'CoronavirusItalia',
'counties',
'KAGA2020',
'Coronavid19',
'LockdownFatigue',
'SQAResults',
'CallOfDutyMobile',
'insurtech',
'bojo',
'tory',
'turbo',
'drugabuse',
'SupplyChain',
'CDC',
'Germany',
'wearenotgoingaway',
'MPC',
'SmallStreamerCommunity',
'TulsaCovidFest2020',
'leaked',
'climatechange',
'hydroxychroloquine',
'mcr',
'ILD',
'FreddieMercury',
'VoteForBidenHarris',
'Balance',
'lovenetball',
'KempKILLS',
'DARBAR_Ni_Deli',
'Nursinghomes',
'workingfromhomewithatoddler',
'Casualties',
'Wada',
'U2',
'SpanishFlu',
'GetCrewChangeDone',
'Shahidafridi',
'authenticlearning',
'sobriety',
'HomeInstead',
'KisanKiBaat',
```

```
'hairdressing',
'businessIT',
'Neet',
'services',
'nevertrumper',
'DansFault',
'plasmadonors',
'PuertoRican',
'ImpeachBarrNOW',
'photographyonline',
'patientsafety',
'9News',
'SocialDistancingKitchen',
'OCD',
'scottishfootball',
'MEcfs',
'donating',
'Northamptonshire',
'SummerBreeze',
'Venezuela',
'MyHealthChat',
'solutionproviders',
'civildisobedience',
'excludedNI',
'geopolitica',
'emotional',
'Honduras',
'kingofventilators',
'Taliban',
'TheBruvs',
'safety',
'mRNA',
'software',
'singleuse',
'RoadToRiding',
'CoronavirusImpact',
'Proms',
'FRIENDS',
'newhair',
'BloodPressure',
'cooking',
'sw',
'resisters',
'murderer',
'Fenway',
```

```
'Nurse',
'schoolsreopening',
'closetheborder',
'MadeInRwanda',
'Garston',
'Osteoporosis',
'PPEs',
'healthprofessionals',
'Seniors',
'tub',
'ChinaJoe',
'MaryShelley',
'ClintonFoundation',
'Okinawa',
'Guatemala',
'Winter',
'CoronaBasKarona',
'director',
'Pink',
'stayinghome',
'greetingcards',
'Conley',
'BenCarson',
'sapsMP',
'ShareCulture',
'Researchers',
'CovidOrganics',
'funnyvideos',
'SexyArmpits',
'onlineshopping',
'StayStrongNC',
'IRAQ',
'iykyk',
'COVID19outbreak',
'economicrecovery',
'358Walkingsticks',
'www',
'HADDOCK',
'totalharms',
'vauxhallcityfarm',
'TuesdayTip',
'GambleResponsibly',
'sleepapnea',
'bbcyourquestions',
'Awake',
```

```
'uvc',
'Revelation',
'china',
'vaping',
'CVS',
'ThePostElection',
'CanadaRecovery',
'KCR',
'inclusivetheatre',
'TaxHavens',
'CorruptTrump',
'Decentralization',
'ESRAASRA2020',
'digitalpatient',
'ProWrestling',
'Branding',
'SinCienciaNoHayFuturo',
'NetSupportSchool',
'payfreeze',
'MaskUpMonday',
'dojo',
'merengue',
'vitamins',
'maskclips',
'PsychStar',
'state',
'Aon',
'NYCTestandTrace',
'pedagoofriday',
'sick',
'Raleigh',
'sackcunmings',
'sundaylunch',
'Chazocrats',
'BTS',
'shaking',
'tuesdayvibe',
'WednesdayThoughts',
'Ballymena',
'supportindieartists',
'ThinkSmarter',
'infusedwater',
'Burnaboy',
'Bojo',
'NGAW20',
```

```
'Nightingale',
'pigs',
'LeedsUnited',
'coal',
'UpstateNY',
'APA',
'FBRPARTY',
'devastating',
'TablighiJamaat',
'bloggerstribe',
'TakeCareOfEachOther',
'treats',
'covid19rftlks',
'backtonormalsoon',
'FutureFund',
'FinallyFriday',
'justasking',
'unlock1',
'Processing',
'COVID19Vic',
'vakantieman',
'Americans',
'myMCNY',
'madeinusa',
'Finland',
'therealplague',
'Tenders',
'Noticias',
'AntiFacemask',
'literary',
'woky',
'263Chat',
'grlpwr',
'Parcel2Pack',
'EducationCannotWait',
'WearAMaskSaveALife',
'Compere',
'borishasfailedtheuk',
'flushots',
'PTSD',
'scotland',
'hydroxycholoroquine',
'nationalfirstrespondersday',
'blacklivesmatter',
'RINA',
```

```
'LNPfail',
'NotMyPM',
'gorillaglue',
'Bucks',
'DavidIcke',
'FabChange20',
'Lombardia',
'Socrates',
'interfaith',
'OOH',
'DISINFECT',
'Case',
'caresact',
'vtpoli',
'biodiversityday',
'greyliterature',
'disinfectant',
'Ahmedabad',
'cryptocurrency',
'PubOpening',
'Telangana',
'VaccinesWork',
'Zicla',
'Gas',
'QuaranTunes',
'Destiny',
'SARSCOV2',
'TheShowMustBePaused',
'NannyState',
'boycott',
'FlipTheSenateBlue',
'excited',
'cep',
'stream',
'leepcovid',
'BookReview',
'SongbirdMovie',
'BorisHasFailedBritain',
'VectorControl',
'Past',
'CMCpodcast',
'UTP',
'KO',
'interviews',
'SASE',
```

```
'besties',
'friends',
'heuteshow',
'TImorLeste',
'rspb',
'layoffs',
'Disparities',
'ML',
'50swomen',
'HavelockRoad',
'UofStirling',
'AccessManagement',
'Janamashtami2020',
'antilockdown',
'RIPIrrfanKhan',
'leadership',
'HealthIT',
'webscammers',
'PES2021',
'nursesareheroes',
'Listing',
'TDL10',
'Revelations',
'Safety_Assessment',
'imaccustomer',
'influenza',
'longtail',
'HealthIsKey4All',
'ParksCanada',
'flu',
'alwaysplaying',
'lockdownbirthday',
'PayItForward',
'MJ',
'Society',
'WalesAtRisk',
'Covid_19uk',
'YCDAB',
'dominicummimgs',
'CarnetEHESS',
'ferry',
'PassTheBill',
'LonConf20',
'westmidsjobs',
'InclusiveInnovation',
```

```
'atp',
'epsteindidntkillhimself',
'Sharia',
'GiveBloodSaveLives',
'Health',
'physicalhealth',
'21st',
'CoronavirusIsDifferent',
'JobRetention',
'Citations',
'Assisteddying',
'BackToWorkSafely',
'CAMHS',
'doityourself',
'MondayMotivaton',
'TrumpDeathToll152K',
'20thCenturyHoax',
'Disease',
'CML',
'LGBTQI',
'Boobs',
'rhino',
'MSCOVID19',
'momlife',
'spacex',
'LifeSaving',
'justaddwater',
'niñosycoronavirus',
'thedoctor',
'litchat',
'machinelearning',
'MedEd',
'poison',
'MITSloanExperts',
'APPGSEND',
'Lovelondon',
'EquityforAll',
'Netflix',
'SimplifyingSelection',
'healthyfood',
'YorkCVS',
'tennis',
'shadows',
'OnCourse',
'SafeQuarantine',
```

```
'ootd',
'greenspace',
'CoronaInPak',
'in',
'lFuckedUrDad',
'wasting',
'covidcon',
'obooks',
'hair',
'beautiful',
'TrumpDeathToll',
'BigBoobs',
'Eton',
'coronaupdatesindia',
'WarRoomTownHall',
'MasksOff',
'September2020',
'ChapelHill',
'TopOfTheWorld',
'zumbavirtual',
'vulnérabilité',
'Somizi',
'WhitePrivilege',
'ThoughtLeadership',
'obesity',
'Somalis',
'skype',
'Twitchstream',
'cybersafe',
'lluminati',
'eu',
'RedFSLN',
'grad',
'BarnardCastle',
'Netherlands',
'Flooding',
'biomarker',
'hairsalonwatford',
'dailyrender',
'Hancockmustgo',
'jewellerymaking',
'BikerDown',
'IVERMIECTIN',
'COVID19challange',
'Books',
```

```
'swineflu',
'cndpoli',
'Universities',
'AbsoFreakinLutely',
'mobilephone',
'haircutsformen',
'Adivasis',
'Biotechnology',
'thoughtoftheday',
'TogetheratHome',
'RMFansEnCasa',
'Brewerania',
'Assassination',
'ukmortgageprisoners',
'attacks',
'hardrockcafe',
'CoronaVirusUK',
'covid19prevention',
'novels',
'executiveorder',
'confused',
'bidenharris2020',
'CovidVic',
'KimberlyGuilfoyle',
'homealone',
'bbqt',
'Orkney',
'PreventableDiseases',
'beki',
'School',
'EmilyMurphyGSA',
'CoVidiots',
'nationalthankateacherday',
'pink',
'habitatloss',
'CovidTesting',
'ProningPatients',
'COVID19SA',
'tyrannei',
'ACEsupported',
'CDNBusinesses',
'jackwimperis',
'Mika',
'Susceptible',
'LMI',
```

```
 'duringthewar',
 'newyorktough',
 'Chevrolet',
 'twibbon',
 'judo',
 'Clydach',
 'GasChambers',
 'Optometrist',
 'PicOfTheDay',
 'islingtonhairsalon',
 ...}
```

In [61]:
```
# Creating subsets of dataset for australia

df_australia = df[df.country == 'australia']
df_australia
```

Out[61]:

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_word_count
120000	Australia is at the forefront of efforts to de...	NaN	True	True	COVID19	australia	266	39
120001	Wow! Look at these numbers! Can't wait to hear...	NaN	False	True	coronavirus	australia	205	38
120002	sometimes it's not the strength\nbut the gentl...	NaN	False	True	myphoto photo	australia	272	31
120003	-The US set a record of over 53,000 new #COVID...	NaN	True	True	COVID—19	australia	296	46
120004	Do we have to change our corporeal habits in o...	NaN	False	True	MondayMotivaton	australia	247	28
...	...	...	...	...	...	...	...	...
159995	ARRIA gives language to data & voice repo...	NaN	FALSE	TRUE	freetrial nlg excel powerbi alexa qlik results...	australia	272	34
159996	Excellent video on vitamin D Vs #Covid_19 \n@C...	NaN	TRUE	FALSE	Covid_19	australia	113	12
159997	@NrsgMutualAid Why explore\n————-CIVILITY?\n\...	NaN	FALSE	TRUE	NursingEducation Nursing Simulation MedEd	australia	319	43
159998	New #VPN Trends Driven by #COVID - whether its...	NaN	FALSE	FALSE	VPN COVID DDoS	australia	237	36

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_word_count
159999	Th #UK governments public approval is begging ...	NaN	FALSE	FALSE	UK staysafe covid19 workingathome	australia	134	15

40000 rows × 13 columns

In [62]:
```
# Creating a set of unique hashtags for australia

hashtag_arr_australia = []

for hashtag in df_australia.hashtags:
    hashtag_arr_temp = hashtag.split(' ')
    for hashtag_temp in hashtag_arr_temp:
        hashtag_arr_australia.append(hashtag_temp)

unique_hashtags_australia = set(hashtag_arr_australia)
unique_hashtags_australia
```

Out[62]:
```
{'sleepinginsweat',
 '19Nov',
 'LordHoweIsland',
 'digitalart',
 'LOPEZGenocida',
 'SociallyDistant',
 'reto',
 'mosquitoes',
 'workingfromhome',
 'Stupid',
 'EsNoticia',
 'TrumpKnewVoteBlue',
 'Qanons',
 'BruceSwedien',
 'slogangraffiti',
 'poll',
 'LoveIt',
 'AnnaBligh',
 'workplace',
 'Gambling',
 'makeup',
 'BCSchoolCovidTracker',
 'sunset',
 'Innovation',
 'MartialLaw',
 'cityofpg',
 'Cannabis',
 'personas',
 'Geisterspiele',
 'Tits',
 'TechForGood',
 'icantbreathe',
 'Impeachment',
 'covidwalks',
 'time2call',
 'PlayingADifferentGame',
 'InfectiousDiseases',
 'mystery',
 'ButHisHair',
 'advicefys',
 'FM20',
 'wegotthis',
 'GoldCoast',
 'landforsale',
 'weekendreads',
```

```
'techno',
'wreathsofinstagram',
'TrumpDevastation',
'agedcarestaff',
'ThreeTimes',
'WorldHypertensionDay',
'skinsense',
'nude',
'submissive',
'DCA',
'Product',
'BitcoinCashBCH',
'mushy',
'installers',
'Kolkata',
'CovidCon',
'Coops',
'cornavirus',
'PHAC',
'20-May',
'BarunSobti',
'bushfirecrisis',
'コスプレ',
'istandwithdan',
'CentralPark',
'VoteBiden',
'Warwick',
'wearewarriors',
'Top20',
'FelizMartes',
'digitalshow',
'covidhoax',
'authors',
'MillionMAGAMarch',
'MIL',
'KeepAmericaFree',
'domestic',
'TiffanyCircle',
'UnitedKingdom',
'gnome',
'LestWeForget',
'AccessToInformation',
'Humanity',
'TedWheeler',
'MaskOn',
```

```
'PortlandOregon',
'rent',
'CoronavirusIndia',
'trachie',
'wimmin',
'FBPE',
'ge2020',
'USElections',
'covid1984',
'IndiaFightsCoronavirus',
'freedom',
'TorresStrait',
'GOV',
'greenlanterncorps',
'NACCHO',
'smallgroups',
'Rutte',
'CoronavirusSupplement',
'UnimelbPursuit',
'cuomoprimetime',
'CoVId19',
'Blacklivesmatter',
'Zionists',
'boardingschool',
'PueblosIndígenas',
'Melton',
'EdgarAllenPoe',
'PhysicalDistancing',
'AspenMedical',
'zimbabwe',
'Sanook',
'Contagious',
'Boston',
'Opportunity',
'mentalillness',
'TrumpDementiaSyndrome',
'FakeNewsMediaClowns',
'Nutrition',
'Aborigines',
'DDoS',
'CentralBank',
'pills',
'Uyghur',
'RSV',
'RIOT',
```

```
'masques',
'globalwebinar',
'anewwayforward',
'Slave',
'PEDOWOOD',
'BanLiveExport',
'TrumpRiots',
'JobSeeker',
'OrangeShitGibbon',
'SackDanAndrews',
'PNDHour',
'horseshoecrab',
'Lost',
'DrBirx',
'SaveAssangeCovid19',
'Hiteshi',
'Medtwitter',
'UU',
'arterialstiffness',
'OpenAmericaUp',
'WesternBorder',
'cdnpoli',
'heatwaves',
'antiviral',
'qldelection',
'TrumpKnewAndDidNothing',
'TrumpVirus',
'NegligentHomicide',
'IndigenousDay',
'Demonstrasi',
'VIOLA',
'CatchMeOutside',
'outsourcing',
'BidenWasNotElected',
'cryptotrading',
'aussies',
'ElderlyAustralians',
'BCElection2020',
'usaCoronavirus',
'Vale',
'esg',
'schools',
'UKpol',
'who',
'Azithromycin',
```

```
'festival',
'visitors',
'covidbc',
'bushfire',
'kitchensoverchildcare',
'NamingTheLost',
'UnemploymentInsurance',
'idiotwind',
'Frazzamatazz',
'NorthernPowerhouse',
'RLF100',
'CancelAnimalAg',
'ToxicShockSyndrome',
'SystemChange',
'DerangedDonald',
'BadaDashain',
'Budget2021',
'act4CleanAir',
'herbaltea',
'Mauritanie',
'CovidLonghaul',
'BAEKHYUN',
'TrabinLaw',
'AmazonHolidayBoycott',
'housingcrisis',
'londonprotest',
'removethecap',
'dimensionjump',
'μενουμε_ασφαλεις',
'bulge',
'NewsBreakfast',
'GATES',
'FedChair',
'ChildTrafficking',
'EM',
'Allity',
'globalwarming',
'BlueWave2020',
'Qom',
'QuarantineLifeBeLike',
'ChrisSmithTonight',
'letsplay',
'Corona',
'TamworthCastle',
'howmanydeathsareenough',
```

```
'ZimbabweanLivesMatter',
'Email',
'genderdata',
'September11',
'painpatientsvote',
'BinanceCharity',
'arts',
'WhiteMilk',
'Ciudad',
'DanniiMinogue',
'WuhanCoronavirus',
'lestweforget',
'CovidWhistleblower',
'BitcoinBank',
'soupkitchens',
'ResearchTogether',
'whore',
'SEC',
'gamegrumos',
'BuildSocialHousing',
'pressforprogress',
'TourismStrong',
'makeMoreAmericansDead',
'OpportunityZone',
'WesternAustralia',
'ATO',
'LockdownDownZim',
'fist',
'Alien',
'aupair',
'saferathome',
'equalrights',
'art',
'pharmacovigilance',
'Ireland',
'StopCOVID19',
'HS2',
'ChakalakaVideo',
'kidneydisease',
'Queen',
'murderhornets',
'Blue',
'mothernature',
'warwick',
'illegalproxyvotes',
```

```
'Egyptologist',
'econhist',
'LiberateVirginia',
'caronavirusoutbreak',
'NotALeaderJustALiberal',
'ANTIVAXERS',
'interiordesigners',
'Advierte',
'YULIN',
'Shivpuri',
'race',
'hurricanes',
'mismanage',
'pandemicduck',
'agedcarerc',
'isolyfe',
'ESRcareers',
'TaoistTaiChi',
'HongKongYouth',
'getaway',
'aircargo',
'covid19comms',
'DaddysGirl',
'Darebin',
'trends',
'Costco',
'putyourmaskon',
'WAs',
'iniso',
'ESTHER',
'PeterGutwein',
'FREE',
'WHO',
'eidmubarak2020',
'HEOR',
'HSI',
'NCPrison',
'NewVideo',
'MLBOpeningDay',
'WorldPopulation',
'Conferencing',
'MorningBuzz',
'TwitterPoll',
'TAXI',
'NBABubble',
```

```
'ToryGenocide',
'TheReidOut',
'TrumpPandemic',
'Benefits',
'NEWS',
'worklife',
'OneNote',
'IranCovidTruth',
'Ali_Younesi',
'NewEra4Food',
'drinkwine',
'sweet',
'nolivegigsjokesahoy',
'manditorymasks',
'fortuneteller',
'rocknroll',
'JacobReesMogg',
'MyFavouriteThings',
'AroundtheWeb',
'stress',
'FamilyTRAIN',
'tired',
'Voted',
'Collaboration',
'DEFCON',
'PopUp',
'Brazil',
'Coercivecontrol',
'abc730',
'modernslavery',
'ClimateBreakdown',
'Slovaquie',
'豆原一成',
'Cubrebocas',
'isodining',
'Virgin',
'Eurovision',
'plea',
'Salud',
'GoodMorning',
'vicfabe',
'uniteandrecover',
'CRISPR',
'JNNP',
'TrumpSpeaderEvent',
```

```
'Populism',
'smashburger',
'3DEXPERIENCE',
'ProtectOurCommunity',
'paisdepandereta',
'covidaustrailia',
'paranormal',
'Tassie',
'cardio',
'Spoonie',
'UniversalCredit',
'follo4folloback',
'新型コロナウイルス',
'streaming',
'StayHome',
'university',
'FreeRefugees',
'fitness',
'BarcoClickshare',
'假账号',
'OneVoice1',
'N95',
'CausalWorkers',
'BeLikeAmyGDala',
'pornharms',
'Europa',
'veteran',
'BearBile',
'watergate',
'again',
'InternationalCleanersDay',
'Tetriseffect',
'MonashLens',
'Palestinian',
'cometogethermelbourne',
'Karpathos',
'mytopfans',
'gettymuseum',
'allthebest',
'RaceRiots',
'WeArePDX',
'uniteforfreedom',
'NursingHome',
'PutTheAdsUoJoeBiden',
'Cuba',
```

```
'halloween',
'frightening',
'buymyfirsthome',
'Víctimas',
'death',
'techradio',
'economists',
'shares',
'Oakland',
'SpreadTheWord',
'Capitalism',
'KeepYourChildHome',
'med',
'dashboard',
'ReparationsNOW',
'GarmentIndustry',
'MyHeroAcademia',
'OpenThePub',
'CRACKERS',
'Day29',
'WeSpeechies',
'wizkidxEbro',
'Messi700',
'Montevideo',
'tiktok',
'Coronavid19',
'drugabuse',
'SupplyChain',
'CDC',
'Germany',
'3Danimation',
'wearenotgoingaway',
'TulsaCovidFest2020',
'DiedFromCOVID',
'กรุงเทพธุรกิจ',
'climatechange',
'hydroxychroloquine',
'smoothskin',
'rfdsSANT',
'CrimeMinister',
'JoanChen',
'InternationalLabourDay',
'DARBAR_Ni_Deli',
'hotelquarantine',
'topmodel',
```

```
'PortlandProud',
'SpanishFlu',
'mmorpg',
'karmavirus',
'RFID',
'SP500',
'IGSS',
'authenticlearning',
'mocked',
'trumpgravedancer',
'Parafield',
'MagaCLOWNS',
'拡散希望',
'parisjetaime',
'processingindex',
'DefundPolice',
'LOPEZTraidor',
'services',
'一日一Made',
'DansFault',
'federal',
'patientsafety',
'9News',
'fitfam',
'coronapocolypse',
'BusinessImprovement',
'SkeemSaam',
'Murphy',
'BMJResearch',
'Venezuela',
'WorldBreastfeedingWeek',
'fintechs',
'isolationstories',
'全力選手権',
'holidayseason',
'FalunGong',
'EarthDay50',
'emotional',
'Honduras',
'HurricaneDelta',
'waittimes',
'JohnKobylt',
'YogaTherapy',
'Zulfiqar',
'Neoliberal',
```

```
'safety',
'mRNA',
'kellyanneconwayhascovid',
'JuliaGillard',
'Microbiota',
'TrueColors',
'ナナニジ',
'USASUCKS',
'BackToSport',
'Fenway',
'Nurse',
'schoolsreopening',
'Osteoporosis',
'healthprofessionals',
'Seniors',
'CombatVetsforTrump',
'ClintonFoundation',
'Guatemala',
'BaghdadPost',
'bigboobs',
'director',
'stayinghome',
'Mink',
'queenvictoriamarket',
'trinitybellwoodspark',
'BenCarson',
'APAC',
'jonesa',
'SackColbeck',
'ConstitutionalRights',
'TheFloorIsLava',
'IRAQ',
'RightsandPromises',
'onlineshopping',
'MindfulnessMeditationApps',
'COVID19outbreak',
'SIDA',
'sleepwear',
'totalharms',
'Cause',
'onlyfansbabe',
'uvc',
'UnknownTransmission',
'CoronaDidntKillIItself',
'china',
```

```
'vaping',
'UnitedNotDivided',
'CorruptTrump',
'globaldrugsurvey',
'FlipTheClinic',
'OPV',
'modelo',
'AdriaTour',
'lifestyletipsbyjc',
'SIDS',
'Whistleblowers',
'EarlyHeadStart',
'SIGGRAPH',
'CubaSalvaVidas',
'amacrinecells',
'chrissyamphlett',
'CoitosDe2minutos',
'NewsMax',
'KamalaHarrisForVP',
'10QuestionsWithAcademics',
'vitamins',
'conferencia',
'Injury',
'CoronaVirusHoax',
'sensitivepeople',
'CompassionateRelease4Reality',
'traitor',
'state',
'2020WasFunUntil',
'sick',
'Drilly4Mayor',
'Chazocrats',
'toxic',
'BTS',
'AntiVaccine',
'Regidora',
'OHCA',
'WednesdayThoughts',
'ABCNEWS',
'CrimeFiction',
'stamp',
'Reynosa',
'beer',
'pigs',
'IranDeal',
```

```
'Nightingale',
'coal',
'TablighiJamaat',
'NotiPaz',
'bloggerstribe',
'StayHomeSavesLives',
'Facemask',
'FireFauciNow',
'COVID19Vic',
'fuckyoufauci',
'jobkeeper',
'Americans',
'NoTeDijeAdios',
'Qingdao',
'putitinthebin',
'MillionsMAGAMarch',
'Tests',
'backstreetboys',
'Noticias',
'OregonWay',
'landing',
'SackBoris',
'fooddrive',
'everyonetogether',
'wearing',
'brainfog',
'TheDaily',
'MealPrep',
'marinelife',
'UNGSII',
'WearAMaskSaveALife',
'whatmattersnow',
'paphos',
'SupportLocalBusinesses',
'PTSD',
'scotland',
'hydroxycholoroquine',
'blacklivesmatter',
'LNPfail',
'PolicyForumPod',
'DistanciamientoSocial',
'DavidIcke',
'unesco',
'SeabirdSaturday',
'Chrysanthemums',
```

```
'FarmersProtests',
'vtpoli',
'nswgov',
'TEMAS',
'Georgiagovernor',
'disinfectant',
'cryptocurrency',
'ecology',
'VaccinesWork',
'Telangana',
'Pediatric',
'Gas',
'SARSCOV2',
'boycott',
'Maskup',
'TakeDownCCP',
'stream',
'BookReview',
'LiberalHypocrisy',
'lockdownmadness',
'IndustryNews',
'coroanvirus',
'friends',
'LawWeek',
'50swomen',
'antilockdown',
'leadership',
'HealthIT',
'LiberalAus',
'NSWPol',
'ResistanceIsNotFutile',
'Blender3d',
'ProChoice',
'bigpharma',
'influenza',
'Bundestag',
'USAelection2020',
'anakinra',
'flu',
'NW',
'PayItForward',
'Society',
'failureofleadership',
'SanayaIrani',
'lungdisease',
```

```
'NZL',
'grumpycat',
'anonymus',
'newjob',
'IndiaAgainstCorona',
'Health',
'CountryMusic',
'MondayMotivaton',
'RelationshipsOverRestrictions',
'musicnews',
'snapchat',
'Disease',
'indiegames',
'LGBTQI',
'Biohazard',
'rhino',
'bunningskaren',
'MadeinLagos',
'trumprally',
'maskme',
'PetSciencePodcast',
'Werribee',
'iStandWithEmperorDan',
'MailInBallotFraud',
'deanogorman',
'JikingeWakingeWengine',
'machinelearning',
'dolphins',
'MedEd',
'gdp',
'poison',
'Waurnponds',
'Netflix',
'corn',
'tennis',
'eatwell',
'roxburghpark',
'italia',
'CovidDiversion',
'kiosks',
'hair',
'camgirl',
'beautiful',
'TrumpDeathToll',
'sextpanther',
```

```
'wholesome',
'ImranKhan',
'TransientSpacesBlog',
'September2020',
'AFLCATSMAGPIES',
'Staycation2020',
'Jordans',
'wanderlust',
'DonaldLiedPeopleDied',
'Kansas',
'agenda21',
'QLDnurses',
'WhitePrivilege',
'Candidates2020',
'obesity',
'Covidaus',
'BuenosDias',
'SOUTHAUSTRALIA',
'lastyear',
'vanmorrison',
'COVID19challange',
'Books',
'swineflu',
'cndpoli',
'FreeAltan',
'Universities',
'hocuspocus',
'chickenpox',
'pearlmethyst',
'fiji',
'TaiwanCDC',
'ManyVids',
'TogetheratHome',
'inducedanxiety',
'garmentindustry',
'Window',
'bigtitsathome',
'AF5',
'syringes',
'tariff',
'WhereToGo',
'from',
'kawaii',
'NovaScotians',
'CovidVic',
```

```
'Egyptians',
'FoodWars',
'Kreuzberg',
'Nepal',
'Orkney',
'School',
'EmilyMurphyGSA',
'chocolate',
'NationalEmergencyPowers',
'Brunch',
'GrahamAshton',
'nationalthankateacherday',
'COVID−19idiots',
'CovidTesting',
'COVID19SA',
'STARTUP',
'footfetısh',
'無限大',
'AOD',
'اعدام',
'portlandmaine',
'Belgique',
'SkyNews',
'swimmandgetsick',
'personaldata',
'Covid19Zim',
'Recycle',
'Rema',
'ACCHOs',
'climatestrike',
'GiantEagleOwl',
'allergy',
'Football',
'narratives',
'lorijeanfinnila',
'robholmes',
'Tudor',
'Aboriginal',
'Renewables',
'FashRev',
'confinamientoperimetral',
'StormGiantsThunder',
'fyp',
'besmart',
'STOLEN',
```

```
'Baylor',
'GladysMustGo',
'HBPRCA_WS2020',
'Ageist',
'geolocation',
'foreigncorrespondent',
'saveaustralia',
'joerogan',
'FPTP',
'Lemmings',
'NaturePhotography',
'reintegration',
'flight',
'QARMY',
'EndCOVIDForAll',
'PENDEJONOMICS',
'DominicCummingsThis',
'ironore',
'peur',
'Hannity',
'Cybercrime',
'LGBTQ',
'CowardinChief',
'Lockdown',
'25Jun',
'logging',
'trust',
'propoganda',
'Dashain',
'lanecove',
'Berlin',
'TheBrianNoeShow',
'ARIAs',
'SteveMnuchin',
'occulthomewear',
'PLWNCDs',
'goingcrazy',
'QueeslandPolice',
'Jotul',
'INTR12214',
'DeFi',
'Search',
'scummo',
'CBSM2020',
'websiteexclusive',
```

```
'PublicPolicy',
'ThalapathyVijay',
'Brevard',
'MindControl',
'Bullstop',
'pioneersofchange',
'LockHimUp',
'Outback',
'ForeverAlone',
'GlobalMeditation',
'OCCOVID19',
'FamilyLoveFaith',
'Flemington3031',
'stayprepared',
'theatre',
'bread',
'BasicIncome',
'flights',
'SMEs',
'LongCovid',
'WakeUpAmerica',
'MensMentalHealth',
'manel',
'JnJCheekyTuesday',
'BitcoinPredictions',
'forensic',
'medrxiv',
'ClapForTheNHS',
'darwinawards',
'HydroxychloroquineWorks',
'Tool',
'三浦春馬',
'VicDiBitetto',
'ActorsLife',
'markloganmp',
'SackHancock',
'BoycottJimsMowing',
'NewsTrading',
'recession',
'onlineretail',
'airside',
'day60oflockdown',
'RACING',
'raisingtherate',
'SPHP',
```

```
'staywell',
'ShortBlack',
'DrugsOffTheStreets',
'Wellcometrust',
'allthingsconsidered',
'Millenials',
'buyingcontent',
'RCAction',
'gaycum',
'RHCP',
'Comply',
'Israeli',
'Fronteras',
'Review',
'santarossasoundbites',
'MagnificentScotland',
'AMOEU',
'LaalSinghChaddha',
'AfricansInChina',
'10May',
'GoBackToAfrica',
'Verschwoerungstheorien',
'workingathome',
'SanQuentin',
'healthtechnology',
'OBAMAGATE',
'NorthTynesideCCG',
'STEMeducation',
'indianfood',
'ScheerStupidity',
'Alzheimer',
'test',
'JFKJR',
'carolebaskin',
'COVIDPacific',
'TrumpDerangementSyndrome',
'DayofMourning',
'λαθρομετανάστες',
'promisedcommunicationskept',
'AustralianSeedFederation',
'PBS',
'dominiccunmings',
'TheFeedZW',
'DeepState',
'WakeUp',
```

```
 'EducationForAll',
 'Humour',
 'fostercarers',
 'designer',
 'OpenSchools',
 'ElDecidor',
 'hole',
 'powerlessness',
 'CovidRelief',
 'baking',
 'builtthatwall',
 'bandanastyle',
 'Ausbiz',
 'GeneralStrike',
 'GatesFoundation',
 'Madurai',
 'onetownatatime',
 'induecard',
 'FreeAccess',
 'AUSTRALIA',
 'totalrecall',
 'FrontLineHeroes',
 'EMRIP',
 'AACRCovid',
 'africansagain',
 'GovernmentLies',
 'dumbfuck',
 'KempKills',
 'playbowls',
 'fingerprintexpert',
 'PCRTest',
 'SCUMMO',
 'Maradona',
 'positves',
 'genetherapy',
 'aliensarereal',
 'EdwardColston',
 'TrumpEconomyDisaster',
 'SuperSpreaderEvent',
 'DEDICA20',
 'Extroverts',
 'CovidChristmas',
 'jdpower',
 'coronavirus',
 'insolvency',
```

```
 'BiggBoss14',
 'Emotes',
 'Countdown',
 'JeffKennett',
 'Karen',
 'VaccineAgenda',
 '10kgold',
 'Queensland',
 'Matrimonio',
 'GiftTax',
 ...}
```

In [63]:
```
# Creating subsets of dataset for ireland

df_ireland = df[df.country == 'ireland']
df_ireland
```

Out[63]:

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_word_count	ha
160000	#COVID19 UPDATE: New Jersey has 796 new positi...	NaN	FALSE	TRUE	COVID19	ireland	250	34	
160001	**25 voices** announce the Government's lockdo...	NaN	FALSE	TRUE	COVID19 Kildare Offaly Laois lockdown	ireland	131	15	
160002	German doctors are sending this around on Twit...	NaN	FALSE	TRUE	coronavirus Covid_19	ireland	97	11	
160003	Hugely proud of our #COVID SLT team for their ...	NaN	FALSE	TRUE	COVID	ireland	313	48	
160004	If you are cocooning, vulnerable or isolated a...	NaN	FALSE	FALSE	Roscommon CommunityCall	ireland	300	43	
...	...	...	...	...	...	...	...	...	
199995	#rtept #Harris hypocrisy personified	NaN	True	False	rtept Harris	ireland	36	4	
199996	#SkillsConnect supports workers who have lost ...	NaN	False	False	SkillsConnect COVID19	ireland	281	40	
199997	1/18 🇸🇪 4th update. Shorter, many aspects are ...	NaN	False	True	covid19	ireland	302	53	

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_word_count	ha
199998	Just when you thought #DominicWest was just a ...	NaN	False	True	DominicWest Covid19	ireland	186	28	
199999	@colettebrowne @SimonHarrisTD has too many let...	colettebrowne	True	False	Covid_19	ireland	170	19	

40000 rows × 13 columns

In [64]:
```
# Creating a set of unique hashtags for ireland

hashtag_arr_ireland = []

for hashtag in df_ireland.hashtags:
    hashtag_arr_temp = hashtag.split(' ')
    for hashtag_temp in hashtag_arr_temp:
        hashtag_arr_ireland.append(hashtag_temp)

unique_hashtags_ireland = set(hashtag_arr_ireland)
unique_hashtags_ireland
```

Out[64]:
```
{'Create',
 'Swedish',
 'SRM',
 'greenelectricity',
 'MandatoryMaskMonday',
 'unitedstate',
 'smallbusinesses',
 'Futurelern',
 'digitalart',
 'idontlikemondays',
 'craic',
 'workingfromhome',
 'Stupid',
 'sandiafolk',
 'TrumpKnewVoteBlue',
 'wildbirds',
 'poll',
 'Resign',
 'VIMEO',
 'workplace',
 'WorldEnvironmentDay2020',
 'TheRookie',
 'sunset',
 'DigitalBanking',
 'Innovation',
 'MartialLaw',
 'coronabonds',
 'Cannabis',
 'ROTFS',
 'TechForGood',
 'ChiefRabbi',
 'covidwalks',
 'InformedCommunities',
 'InfectiousDiseases',
 'CaptainTomMoore',
 'northernireland',
 'Cobra',
 'Autumnwatch',
 'wreathsofinstagram',
 'WorldHypertensionDay',
 'UltraHD',
 'nude',
 'practitioners',
 'CoronaVirusIreland',
 'onlineclasses',
```

```
'LocalPropertyTax',
'medieval',
'ExcellentEight',
'Coops',
'NCYT',
'LeisureComplexatLoughLannagh',
'cornavirus',
'shannon',
'HeartOfTheCommunity',
'ChallengePovertyWeek',
'SpinUnit',
'TakeOnTheBank',
'TrustTransland',
'VoteBiden',
'influencers',
'CartwrightKing',
'sinnfein',
'YEADON',
'britain',
'FelizMartes',
'covidhoax',
'Hounslow',
'baobesity',
'livingwithHIV',
'HowTo',
'conspiracytheorist',
'domestic',
'BiodiversityWeek',
'ChineseCoronaVirus',
'Viagra',
'PCRTests',
'Kerryhour',
'Housingassociation',
'disabilitytwitter',
'HumourConFinement',
'UnitedKingdom',
'gnome',
'MaskOn',
'Humanity',
'cocktailhour',
'GretaThunberg',
'ghosts',
'PEP',
'CeolVid',
'ballsbridge',
```

```
'rent',
'Chloroquine',
'eLibrary',
'SouthamptonHospital',
'ge2020',
'Banbridge',
'youngones',
'marriedlife',
'covid1984',
'JonathanVanTam',
'festive',
'Snow',
'freedom',
'pharmacyapp',
'potus2020',
'autismireland',
'prochoice',
'BeerBods',
'CoronaSurvivors',
'Socialism',
'maradona',
'Fulham',
'maternityrights',
'CoVId19',
'Zionists',
'yonex',
'PostalWorkersDay',
'mechanisms',
'borisbaby',
'Affinity',
'PhysicalDistancing',
'monthlycineclub',
'RIPDaveGreenfield',
'Opportunity',
'RTEFakeNews',
'BAPsychology',
'mentalillness',
'StayConnectedTogether',
'TrumpFamily',
'Minister',
'NoBackToNormal',
'DigitalProtection',
'YourCouncilDay',
'GARYLINEKER',
'IStandWithFauci',
```

```
'DrawOurHeroes',
'masques',
'CDETB',
'CORONAHOAX',
'WelcomeApp',
'irishart',
'TrumpRiots',
'OrangeShitGibbon',
'LockHerUp',
'Lost',
'inspirational',
'routine',
'LOLITAEXPRESS',
'medstudenttwitter',
'cdnpoli',
'schoolreopening',
'Gaillimh',
'TrumpVirus',
'Leavingcert',
'Kindlymask',
'usaCoronavirus',
'ReduceYourContacts',
'schools',
'No10',
'who',
'thankateacherday',
'gynae',
'UKDictatorship',
'LCbusiness',
'Azithromycin',
'ENECOVID',
'festival',
'CommunityResponse',
'doitforDan',
'Stayathomechallenge',
'metals',
'FriendshipDay',
'ReadtheInsert',
'EnterpriseCentres',
'Quarantunes',
'ThisMeansMore',
'Esoteric',
'balmoralshow',
'PrimaryCareCentre',
'beenthere',
```

```
'NPHET',
'LordOfTheRings',
'Satanic',
'Budget2021',
'newtownards',
'PandemicsReport',
'motd2',
'blackhumour',
'Flights',
'RobinSwann',
'TerryWaite',
'housingcrisis',
'londonprotest',
'Clare',
'JulyStimulus',
'SCAM',
'shamednation',
'coveryourface',
'Immunity',
'Gardaí',
'blacklifematters',
'covid19hoax',
'ChildTrafficking',
'paralytics',
'RonMalka',
'Belfasthour',
'weak',
'GoodNightTwitterWorld',
'geriatrician',
'carersweek',
'Corona',
'Deluca',
'Democracy',
'ITV',
'ToryBritain',
'AlcoholAwarenessWeek',
'Kill4Fun',
'jockdown',
'genderdata',
'Down',
'arts',
'NepotismBarbie',
'TrumpCrimeSyndicate',
'Cummgate',
'AmazonMoratorium',
```

```
'CulturalMarxism',
'WuhanCoronavirus',
'OurOnlyFuture',
'MaterHospital',
'Seamus',
'Tullamore',
'soblessedtolivehere',
'SEC',
'Adrenochrome',
'Cityscape',
'GeertWilders',
'aCareworkersTale',
'metaanalysis',
'MorningFocus',
'AutumnColours',
'CoronavirusBo',
'Sunny',
'golfswing',
'ourSIPTU',
'hilarymantel',
'BarrowinFurness',
'juicefast',
'art',
'FingersCrossed',
'ChangingIreland',
'ThatsFootball',
'Negativity',
'pharmacovigilance',
'AsthmaTopTips',
'Ireland',
'StopCOVID19',
'destress',
'Queen',
'Shocking',
'HeretoHelp',
'funnymemes',
'luas',
'covidbutterflies',
'walk',
'ProBonoWeek',
'Laffer',
'JalilMuntaqim',
'screener',
'needtotalk',
'kildarelockdown',
```

```
'MissingIt',
'YULIN',
'FENTLINDI',
'HaveGreatWeekend',
'VERBORGENARMOEDENL',
'LifeGoesOn',
'BABYMETAL',
'trends',
'HRBarometre',
'fermanagh',
'FREE',
'WHO',
'InternationalDayoftheGirl',
'MLBOpeningDay',
'InnovateTogether',
'HighlightOnlineConcert',
'VidasNegrasImportam',
'statathome',
'ToryGenocide',
'meath',
'seanad',
'NEWS',
'worklife',
'beautyoftheordinary',
'Scunthorpe',
'SputnikV',
'MissHolly',
'woulfe',
'JacobReesMogg',
'stress',
'Benandjerrys',
'wibble',
'CoronaWatch',
'Brazil',
'Dail',
'UncleJoe',
'xtendbarre',
'abc730',
'Visors',
'StopBensDeportation',
'enter',
'SpudGunDesign',
'Eurovision',
'olivetreekitchen',
'CapturingHistory',
```

```
'FrontlineHeroes',
'ryanairrefunds',
'LeaveToRemain',
'TheMentalHealthFund',
'GoodMorning',
'CRISPR',
'Luciferian',
'covidmaternity',
'sundayshred',
'IWMD2020',
'carersrock',
'ForgottenLordMayor',
'新型コロナウイルス',
'TLM',
'zoos',
'pubsreopen',
'UniversalCredit',
'rebel',
'streaming',
'MICKJAGGER',
'StayHome',
'Telanganaassembly',
'university',
'Shelters',
'fitness',
'BigBetting',
'CastlewellanJoey',
'Christians',
'N95',
'thankyoutony',
'ecpobesity',
'DeloresCahill',
'itsallaboutthewood',
'StayingVirtuallyConnected',
'perspex',
'koreanwar70',
'DerryGirls',
'Americandream',
'coffebreak',
'sdg6',
'ODD1',
'quarantineonlineparty',
'shoulder',
'kazatomprom',
'CopingWithCovid',
```

```
'Palestinian',
'GetTheFluShot',
'mytopfans',
'SCO',
'RaceRiots',
'Overweight',
'LakersWin',
'InvestigateBillGates',
'NursingHome',
'PeterHermann',
'COVIDGATE',
'Cuba',
'halloween',
'biopharma',
'Double',
'salt',
'HeroesAtSeaShoutout',
'DRFAUCI',
'CPT',
'Something',
'Oakland',
'quack',
'kits',
'commonsense',
'SVPireland',
'LCHF',
'Donegal',
'FUNFACT',
'Wancocks',
'Cooperatives',
'PCCs',
'Commentator',
'LoveSPGC',
'tiktok',
'Coronavid19',
'PD',
'Direct',
'socialbubbles',
'bojo',
'Translink',
'SupplyChain',
'CDC',
'Germany',
'wexford',
'CakeSale',
```

```
'Cuomosexual',
'wearenotgoingaway',
'PrayforNigeria',
'WilliamShakespeare',
'climatechange',
'hydroxychroloquine',
'ChooseLove',
'esai20',
'Earths',
'FreddieMercury',
'GiveawayAlert',
'DARBAR_Ni_Deli',
'heroic',
'LSR',
'StrangerThings',
'U2',
'MyKindOfTown',
'NewYorkDailyNews',
'buymedia',
'RFID',
'Greenlist',
'true',
'speeding',
'gobshites',
'stayclear',
'services',
'SeptemberToRemember',
'autonomic',
'purecork',
'patientsafety',
'9News',
'insightwomen',
'80s',
'BilderbergGroup',
'dart',
'rtept',
'affordableaccommodation',
'BMJResearch',
'EveryFriday1pm',
'Venezuela',
'excludedNI',
'GitHubSatellite',
'nhsvolunteers',
'FalunGong',
'Alive',
```

```
'sharingthevision',
'keepingactive',
'Magasins',
'supplyteachers',
'safety',
'software',
'mRNA',
'Quantas',
'WYSD20',
'Cuddles',
'cooking',
'oneisland',
'DanskeBankPrem',
'Mudanjiang',
'schoolsreopening',
'coronavirusbriefing',
'EiffelTower',
'KerryOpenForBusiness',
'ChooseWell',
'SoftwareTesting',
'livingNOTlockdown',
'Pandession',
'Guatemala',
'stayinghome',
'Pink',
'Conley',
'RTEIrelandRemembers',
'IE',
'BenCarson',
'RTEPT',
'ShareCulture',
'Researchers',
'onlineshopping',
'brightonprotest',
'severance',
'NLRP3',
'WeAreWithYou',
'Skerries',
'COVID19outbreak',
'economicrecovery',
'eamonryan',
'CherrySeaborn',
'totalharms',
'bbcyourquestions',
'GemmaoDoherty',
```

```
'PRIMERDESINGLTD',
'onlyfansbabe',
'Awake',
'crpd',
'Covid19hoax',
'china',
'vaping',
'Revelation',
'coviddays',
'TaxHavens',
'PfizerBiotech',
'KCR',
'Jaythan',
'LFCatHome',
'AchievingImpact',
'RandPaul',
'BEER',
'SIDS',
'Whistleblowers',
'ArtistoftheSummer',
'Busdriver',
'Wexfordhour',
'Foreverhome',
'animalliberation',
'92newsireland',
'NewYorkTimes',
'rteinvestigates',
'CompassionateRelease4Reality',
'Dontbecomecomplacent',
'Sensible',
'newross',
'ENGIRL',
'frontlineheroes',
'Phase4',
'sackcunmings',
'BTS',
'habits',
'Covid19SafeSystemofWork',
'Didomustgo',
'CrimeFiction',
'tuesdayvibe',
'WednesdayThoughts',
'beer',
'pigs',
'RadiationPoisoning',
```

```
'Nightingale',
'coal',
'Taoiseach',
'irelanddaily',
'communitypharmaciesweek',
'NottingHillCarnival',
'Enterprise',
'Facemask',
'EltonJohn',
'Walkingthroughthecrisis',
'COVID19Vic',
'bidenharis2020',
'HousingExclusion2020',
'Americans',
'pqa20',
'Finland',
'CountyAntrim',
'IrishEducation',
'SackBoris',
'OdeToJoyIE',
'Tests',
'originalcharacter',
'physios',
'PESSPA',
'ZoomEvent',
'lonliness',
'DDOR',
'OnMyMind',
'CraigavonHospital',
'Swift',
'WearAMaskSaveALife',
'LordMayor',
'stillcelebrating',
'StickGrandpaByTheWindow',
'SupportLocalBusinesses',
'Abba',
'blacklivesmatter',
'yourewelcometolaois',
'badger',
'timesup',
'covidrecovery',
'DavidIcke',
'apeuro',
'keepwarm',
'stayactive',
```

```
'animalcruelty',
'cjrs',
'disinfectant',
'Ahmedabad',
'cryptocurrency',
'Telangana',
'VaccinesWork',
'SinnFéin',
'redflag',
'TUICUMembers',
'ecology',
'SARSCOV2',
'Campbelltown',
'Groningen',
'antivirus',
'stream',
'BorisHasFailedBritain',
'newballsplease',
'lockdownmadness',
'tiocfaimidslán',
'MathsWeek2020',
'interviews',
'coroanvirus',
'idratherbeinmikethepies',
'friends',
'Psychotherapist',
'ML',
'homesafety',
'SmokingJeane',
'pokeronline',
'pastaFriday',
'CeannSibéalfromadistance',
'covidpressbriefing',
'DariusGuppy',
'gemmaodoherty',
'eatplaychangeapp',
'antilockdown',
'leadership',
'Rothchild',
'MaithThú',
'TDL10',
'KanyeWest',
'influenza',
'primaryhealthcare',
'Bundestag',
```

```
'FutureEchoes',
'ChildPoverty',
'HoldHope',
'saveourjobs',
'flu',
'PayItForward',
'NW',
'communitytracker',
'Wockhardt',
'OneMillion',
'Sick',
'Health',
'CovidIreland',
'Bantry',
'BackToWorkSafely',
'cluster',
'StabilityFund',
'MondayMotivaton',
'Dentist',
'LeoTheLoser',
'snapchat',
'vaccinebrothers',
'CoronavirusDisease',
'negazionisti',
'indiegames',
'hyperemesis',
'soldonitsfirstouting',
'MARXIST',
'rhino',
'economicsystem',
'NoComplacency',
'cruel',
'coronajustice',
'trumprally',
'WhateverLevel',
'nurses2020',
'Troublemaker',
'REOPEN',
'gbMSM',
'juliehealy',
'WhileNobodyIsWatching',
'Airbus',
'saveThechildren',
'saveireland',
'machinelearning',
```

```
'dolphins',
'MedEd',
'Fatphobia',
'Netflix',
'DiabetesAwarenessMonth',
'tennis',
'ZozimusBar',
'openhousedublin',
'SETCovidFramework',
'greenspace',
'Fall2020',
'hair',
'beautiful',
'Eton',
'MasksOff',
'Beninprotest',
'Staycation2020',
'AAC',
'NewcastleNews',
'Dunlavin',
'TorysOut',
'zine',
'tipperary',
'WhitePrivilege',
'Tenerife',
'obesity',
'skype',
'SkillsChallenge',
'gameOVER',
'WeekOfMourning',
'codeword',
'Netherlands',
'freeworkout',
'100DaysOfChaos',
'kilkennyWexfordCarlow',
'LIBERAL',
'InternationalCrisisGroup',
'Hancockmustgo',
'Books',
'swineflu',
'OER',
'ELINTNews',
'Universities',
'lymphopenia',
'objectivity',
```

```
'TOV',
'WebinarPandemic',
'UKfishingindustry',
'RVH',
'thiswillallendsoon',
'clorox',
'NovaScotians',
'confused',
'TeamWAST',
'bidenharris2020',
'COVID2019IRELAND',
'brunofernandes',
'bakingtheworldabetterplace',
'Nepal',
'Orkney',
'98FMDublinTalks',
'School',
'HappyTail',
'EmilyMurphyGSA',
'childsexualabuse',
'familysupport',
'epa',
'MaskUpSaveLives',
'info',
'CovidTesting',
'Wembley',
'RussleBurton',
'Covid19disease',
'Mermaids',
'RipoffIreland',
'Kępna',
'Belgique',
'Leftist',
'SkyNews',
'midwives2020',
'communitymatters',
'Museum30',
'Devolved',
'jungle',
'dataprotection',
'Mossad',
'legomovie',
'agroecology',
'climatestrike',
'Burren',
```

```
'Football',
'ResearchersNight',
'Chelsea',
'U308',
'questioneverything',
'glengormley',
'Journalism',
'FillYourHeartWithIreland',
'WetMarket',
'hypothyroidism',
'Tudor',
'ToryCorruption',
'SharedIreland',
'antiseptic',
'besmart',
'SkillsMatch',
'TreeCheers',
'TheScore',
'seeyouinAugust',
'PPI',
'MOTD',
'employeehealth',
'ENDMMSABUSE',
'NaturePhotography',
'SOTEU2020',
'scammers',
'hairdressers',
'billion',
'Ayurvedickit',
'Ofcom',
'tradmusic',
'CYBERFLAG',
'TougherTogether',
'ESSD',
'nillskillcrew',
'NaFianna',
'TobiasEllwood',
'Just',
'yogalife',
'TruthMovement',
'WeeklyInvoiceTracker',
'Cybercrime',
'notinmyname',
'Foreigners',
'Lockdown',
```

```
'Luckykhambule',
'uisneach',
'schoolreopeningIreland',
'WDoR2020',
'Campania',
'artistoninstagram',
'BidenCares',
'Berlin',
'Foodies',
'stormhour',
'healthyworkinglives',
'openamericanow',
'sharethejourney',
'americastrong',
'heartless',
'PublicPolicy',
'HopeNotHate',
'WorldHeritage',
'golfgate',
'judges',
'BasicIncome',
'flights',
'SMEs',
'LongCovid',
'winetasting',
'WakeUpAmerica',
'onlinecourse',
'manel',
'TypeIIR',
'Munster',
'gilimedidas',
'travellers',
'VicDiBitetto',
'healing',
'ProfJackLambert',
'TripleSets',
'GuyAdams',
'freddieking',
'Buhera',
'RICKYDEARMAN',
'recession',
'TashaK',
'TruthIsStrangerThatFiction',
'Oviedo',
'GlobalSparks',
```

```
'SelfRegulating',
'Stmarksschool',
'staywell',
'CoronaVirusHOAX',
'Dying',
'HaveAGreatWeekend',
'DrugsOffTheStreets',
'Blind',
'buyingcontent',
'Wikipedia',
'savewithstories',
'Israeli',
'karens',
'lockdownskill',
'smellthecoffee',
'GiftTheCity',
'Households',
'inspired',
'Verschwoerungstheorien',
'misunderstanding',
'FoodWaste',
'Engagement',
'Ineedhelp',
'Saudicrimes',
'twiglets',
'dublinbus',
'earthday',
'mfd53',
'Thessaloniki',
'Spanish',
'policiers',
'StopHidingISL',
'test',
'TrumpDerangementSyndrome',
'SensoryProcessing',
'punkrock',
'PBS',
'Ironman4Crumlin',
'WakeUp',
'EducationForAll',
'designer',
'OpenSchools',
'SustainableFinanceEU',
'GoldmanSachs',
'eucoronavirus',
```

```
'baking',
'irishweddingchat',
'franchise',
'GatesFoundation',
'covd19',
'HopenotHate',
'KeepSafe',
'AUSTRALIA',
'WorldEconomicForum',
'concernlongjump',
'FrontLineHeroes',
'prerecorded',
'GAAD',
'CovidCurfew',
'belgianchocolate',
'surviving2020',
'RyanairRobbers',
'casualworkers',
'TY20_21',
'moriartygatheringdust',
'SuperSpreaderEvent',
'WorldTourismDay',
'MUSIC',
'psoriasis',
'Trauma',
'hiimsimontoo',
'coronavirus',
'globalmindtravel',
'LKA',
'Torino',
'VaccineAgenda',
'SWResponds',
'Karen',
'Queensland',
'GrosvenorRoadSurgery',
'HRBConf2020',
'kegs',
'DarkWeb',
'Backtoworksafely',
'patient',
'CoronavirusUK',
'ALGORE',
'Virtual',
'Kashmiris',
'outdoors',
```

```
'TFCLIVE',
'Covid19Germany',
'clusters',
'Centra',
'gouvernement',
'BeHapoy',
'OliverJeffers',
'COVIDView',
'southendbeach',
'Millwall',
'Indiana',
'L2Day',
'DarknessintoLightSligo',
'mask4all',
'corcaighabu',
'sunny',
'helpme',
'SAGEScandal',
'Betterworld',
'abyss',
'canaryislands',
'GatesForPrison2020',
'professors',
'apfsd',
'Málaga',
'bluebell',
'BusinessGrowth',
'RAFLuton',
'Downsizing',
'park',
'rte',
'DailyMirror',
'environmental',
'Rothschild',
'Patriots',
'clapforNHS',
'Somalia',
'bar',
'ICIC20Virtual',
'livetheirbestlife',
'lowtide',
'CivicDuty',
'mondaymorning',
'Rothshcild',
'Brixton',
```

```
 'CabInLondon',
 'movieart',
 'PressConference',
 'worldkindnessday2020',
 'WoulfeMUSTgo',
 'Stress',
 'ChristianLivesMatter',
 'safehands',
 'officespace',
 'Hungarian',
 ...}
```

In [65]:
```
# Creating subsets of dataset for new zealand

df_new_zealand = df[df.country == 'new_zealand']
df_new_zealand
```

Out[65]:

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_word_
200000	HEARTBREAKING—Dr. Carlos Araujo-Preza treated ...	NaN	False	True	COVID19	new_zealand	303	
200001	Do you think the media is biased to the Left o...	NaN	False	True	nzpol COVID—19	new_zealand	76	
200002	#coronavirus #BreakingNews #BREAKING #COVID19 ...	NaN	True	False	coronavirus BreakingNews BREAKING COVID19 Covi...	new_zealand	111	
200003	In response to #COVID19, #LegalDepartments pla...	NaN	False	False	COVID19 LegalDepartments	new_zealand	176	
200004	President #Trump is Speaking and EXPOSING the ...	NaN	False	True	Trump coronavirus	new_zealand	191	
...	...	...	...	...	...	...	...	...
239995	Aa Likes, Retweets yentra 🙏\n🔥🔥🔥 \n#Master	NaN	True	True	Master	new_zealand	39	
239996	Very interesting\nAny thoughts? \n\n#TheFive #T...	NaN	False	True	TheFive Trump2020 KAG2020 mondaythoughts COVID...	new_zealand	142	
239997	As we deal with #COVID19 don't forget that #Ch...	NaN	True	True	COVID19 Christians persecution Nigeria	new_zealand	307	
239998	While we hit 150,000 in #COVID19 deaths, the P...	NaN	False	True	COVID19	new_zealand	115	
239999	This too shall pass #Covid_19 . May remain sta...	NaN	False	True	Covid_19 HopeAlive	new_zealand	280	

40000 rows × 13 columns

In [66]:
```
hashtag_arr_new_zealand = []

for hashtag in df_new_zealand.hashtags:
    hashtag_arr_temp = hashtag.split(' ')
    for hashtag_temp in hashtag_arr_temp:
        hashtag_arr_new_zealand.append(hashtag_temp)

unique_hashtags_new_zealand = set(hashtag_arr_new_zealand)
unique_hashtags_new_zealand
```

Out[66]:
```
{'Swedish',
 'Islamophobia',
 'dokter',
 'gloryhole',
 'digitalart',
 'knit',
 'FunFacts',
 'scarytimes',
 'workingfromhome',
 'patio',
 'TobaccoHarmReduction',
 'Problem',
 'TrumpKnewVoteBlue',
 'noexcuses',
 'poll',
 'CoronavirusEnArgentina',
 'canoncamera',
 'workplace',
 'makeup',
 'WorldEnvironmentDay2020',
 'WeTakeCareofEachOther',
 '21May',
 'ResourceLibrary',
 'sunset',
 'Innovation',
 'MartialLaw',
 'TestPilot',
 'Cannabis',
 'postapocalyptic',
 'SirRichardDearlove',
 'icantbreathe',
 'TechForGood',
 'TestingTarget',
 'DSBN',
 'rollei',
 'cizaseason',
 'DragRace',
 'infectadura',
 'CaptainTomMoore',
 'hypocrites',
 'mystery',
 'Protestas',
 'wehaveiteasy',
 'wegotthis',
 'Seagulls',
```

```
'WorldNewsDay',
'wreathsofinstagram',
'75UN',
'submissive',
'nude',
'KylemorePassHotel',
'Wankers',
'Moodle',
'1stFollowUpCallBack',
'CovidCon',
'MissArdern',
'PHAC',
'cornavirus',
'KCC',
'Bikes',
'3in4',
'CentralPark',
'agricultural',
'VoteBiden',
'CanadiEM',
'happymonday',
'ConnectedHealth',
'Kz2',
'Sars',
'covidhoax',
'VeniceSkatePark',
'MillionMAGAMarch',
'mhfnz',
'CivicLongWeekend',
'canadaintheworld',
'dumpthetrump',
'Unschooling',
'yogaathome',
'lifelessons',
'expert',
'stopthevirus',
'UnitedKingdom',
'gnome',
'2020Grad',
'MaskOn',
'Humanity',
'migrant',
'ComingTogether',
'rent',
'COVID—19response',
```

```
'Chloroquine',
'bmx',
'FBPE',
'NWT',
'IndiaFightsCoronavirus',
'festive',
'freedom',
'YourSayNZ',
'mobileshooting',
'videoviral',
'Rutte',
'MenstruationMatters',
'EricGarcetti',
'gdp2020',
'covid19vaccine',
'CoVId19',
'pinkart',
'ドラクエウォーク',
'Zionists',
'itme',
'SaudiTrump',
'product',
'NZDataOps',
'everest',
'PhysicalDistancing',
'CB',
'UnMask',
'Ogiri',
'fordnation',
'NoMandatoryCovidVaccine',
'creatively',
'memesforclimate',
'Boat',
'AslansCountry',
'Boston',
'hamilton',
'mentalillness',
'TrumpDementiaSyndrome',
'Nutrition',
'Uyghur',
'covid19NZ',
'teamoffivemillion',
'RSV',
'ReleaseTheRussiaReport',
'IStandWithFauci',
```

```
'masques',
'heating',
'modernart',
'BBBScam',
'TrumpRiots',
'OrangeShitGibbon',
'animalresearchsaveslives',
'ByeDon2020',
'longweekend',
'HealthResearchCouncil',
'inspirational',
'DrBirx',
'مرزوقة',
'HamiltonOntario',
'clothmask',
'NASH',
'horseshoecrab',
'LoveSomerset',
'watered',
'cdnpoli',
'heatwaves',
'PIFBLM',
'HRCommunity',
'antiviral',
'TrumpKnewAndDidNothing',
'TrumpVirus',
'NegligentHomicide',
'CarriedInterest',
'BidenWasNotElected',
'usaCoronavirus',
'contemporaryteamskills',
'schools',
'who',
'befree',
'Kawashocki',
'CRYPTO',
'Azithromycin',
'festival',
'RedAlert',
'NamingTheLost',
'BunkerBoy',
'houseArrest',
'FarmBoyRefresh',
'Desdemona',
'WomenInPolitics',
```

```
'LordOfTheRings',
'PandemicsReport',
'lofihiphop',
'Infectadura',
'DensityDoneWell',
'Flights',
'housingcrisis',
'londonprotest',
'GlobalGovernance',
'STOPLYINGTOUS',
'SCAM',
'ACTogether',
'Immunity',
'MentalHealthAwarenessWeek2020',
'POLAND',
'blacklifematters',
'ChildTrafficking',
'tearareomāori',
'paralytics',
'globalwarming',
'brandonrouth',
'PostalService',
'tpswednesday',
'GoodNightTwitterWorld',
'carersweek',
'procreate',
'Corona',
'Soundcloud',
'Canon',
'ITV',
'ToryBritain',
'Democracy',
'lockdownhaircut',
'TwitchFam',
'RHPS',
'arts',
'smallchanges',
'Interfaith',
'AmazonMoratorium',
'CulturalMarxism',
'WuhanCoronavirus',
'Live',
'gorilla',
'SAVETHECHILDERN',
'protection',
```

```
'upnorth',
'SchoolsNotBars',
'Adrenochrome',
'BatvVrus',
'AutumnColours',
'ViseUp',
'AAOSNow',
'Employeesafety',
'existentiality',
'treason',
'art',
'ReSS2020',
'ChildrensLivesMatter',
'EnterpriseCanada',
'Day30',
'Ireland',
'onlyfanspages',
'venicebeach',
'HS2',
'Queen',
'funnymemes',
'FundingOpportunity',
'Blue',
'OneTeamOneDream',
'libertà',
'walk',
'PremierFord',
'JalilMuntaqim',
'MAid',
'全球大流行',
'race',
'ARiEAL_Researchers',
'Scamdiemic',
'braincancerwarrior',
'lostforwords',
'TrumpRallyNC',
'Solidarität',
'FakeBillionaireTrump',
'Postdoc',
'medlabON',
'Costco',
'trends',
'WeAreReady',
'ACPSEM',
'Rumble',
```

```
'Rewilding',
'HumpDaaaaaayyyy',
'kpss2020',
'tuesdaytrivia',
'BigHousingThankYou',
'nanoscience',
'COVIDmemes',
'Aerosol',
'FREE',
'PakistaniGirl',
'WHO',
'ReopenNZ',
'SomersetDay',
'TuneIn2InTune',
'SANZAR',
'OntarioTeachers',
'BUFvsTEN',
'MLBOpeningDay',
'genderparity',
'NHLMascotGameNight',
'TheReidOut',
'ToryGenocide',
'TrumpPandemic',
'dougford',
'MatthewHooton',
'RWCfinals',
'Level2',
'NoVaseline',
'faversham',
'Desantis',
'wearafrickinmask',
'BlursDay',
'CoronavirusWar',
'menneedtoact',
'ImmigrationÇaCompte',
'sweet',
'ابـوقـحط',
'Presidental_Candidate',
'SputnikV',
'malieveld',
'rocknroll',
'stress',
'SkyTourMovie',
'Voted',
'Elsevier',
```

```
'Brazil',
'minigolf',
'NEC',
'Masturbate',
'newmexico',
'covidsigns',
'BOJ',
'ClimateBreakdown',
'HomeTogether',
'filmeverything',
'Bay',
'playing',
'Salud',
'GoodMorning',
'HappySeptember1st',
'CRISPR',
'BigHearts',
'DigitalInclusionNZ',
'TrumpSpeaderEvent',
'whatsappstokvel',
'righttofood',
'McMasterPerspective',
'PS4share',
'paranormal',
'新型コロナウイルス',
'ServingHumanity',
'UniversalCredit',
'legendlockdown',
'streaming',
'HumberHeights',
'StayHome',
'university',
'fitness',
'STD',
'ArtOfQuarantine',
'shopsmallbusiness',
'NationalLampoonsEuropeanVacation',
'OneVoice1',
'austrianeconomics',
'StandUp',
'N95',
'TeamofFiveMillion',
'Christians',
'extremeweather',
'sturgisrally',
```

```
'pubquiz',
'武汉病毒',
'Europa',
'veteran',
'Chatham',
'michaeljackson',
'Goats',
'again',
'TrumpSuperSpreader',
'chemcarenz',
'CopingWithCovid',
'Palestinian',
'NoBlueNoGreen',
'PeerRevWeek20',
'mytopfans',
'Margate',
'covidisahoax',
'BBB',
'Deliberately',
'ToryScumbags',
'8kun',
'fracking',
'Pembangkang',
'wudu',
'schoolsbeforebars',
'NursingHome',
'giditraffic',
'TeamCanada',
'covid19MA',
'Cuba',
'halloween',
'CovidSurge',
'salt',
'EcosystemRestoration',
'model3',
'HeroesAtSeaShoutout',
'death',
'thermography',
'OttawaGroup',
'Capitalism',
'KeepYourChildHome',
'Sociological',
'houses',
'BaldyMan',
'MyHeroAcademia',
```

```
'EveryoneNeedsALittleTLC',
'Day29',
'sweepstakes',
'nitwit',
'stephenamell',
'crise',
'MayoClinic',
'tiktok',
'CallOfDutyMobile',
'KAGA2020',
'filmmaking',
'qrcode',
'Coronavid19',
'SupplyChain',
'CDC',
'Germany',
'tannka',
'SmallStreamerCommunity',
'wexford',
'gang',
'climatechange',
'hydroxychroloquine',
'Fifa',
'Lamborn',
'pruebas',
'FreddieMercury',
'GiveawayAlert',
'playitsafe',
'Cristobal',
'assessments',
'vvd',
'digitaltwin',
'SpanishFlu',
'Lofihiphop',
'MARCH',
'RFID',
'TrumpKillsAmericans',
'trumpgravedancer',
'reopenNZ',
'DefundPolice',
'Parafield',
'Neet',
'internship',
'DansFault',
'PhysEd',
```

```
'TheChive',
'9News',
'OCD',
'Surgery',
'coronapocolypse',
'theirsacrificeisreal',
'80s',
'KeepCookingCarryOn',
'blurtblog',
'BMJResearch',
'Venezuela',
'EarthDay50',
'Ayer',
'Honduras',
'COVIDscandal',
'AIintegrated',
'safety',
'software',
'singleuse',
'mRNA',
'ONPoli',
'Block',
'healthliving',
'ForEveryChild',
'Mudanjiang',
'Nurse',
'schoolsreopening',
'Osteoporosis',
'PPEs',
'Seniors',
'sonos',
'letter',
'ClintonFoundation',
'Okinawa',
'SASSACARES',
'saddays',
'bigboobs',
'greetingcards',
'Hype',
'trinitybellwoodspark',
'BenCarson',
'ConstitutionalRights',
'TousContreMacronJour5',
'COVID19outbreak',
'WaterIsLife',
```

```
'likeforlikes',
'NoMoreFlu',
'totalharms',
'onlyfansbabe',
'IndependentSAGE',
'medlabtx',
'vaping',
'CVS',
'china',
'CanadaRecovery',
'KenoshaRiot',
'ps4',
'stgeorgesday',
'govegan',
'marketingresearch',
'KamalaHarrisForVP',
'vitamins',
'HKIA',
'Injury',
'LawOfAttraction',
'NewYorkTimes',
'CompassionateRelease4Reality',
'2020WasFunUntil',
'Chazocrats',
'frontlineheroes',
'disarmament',
'BTS',
'WednesdayThoughts',
'memorialday',
'Destiny2NewLight',
'notobacco',
'beer',
'Nightingale',
'IamCanadian',
'coal',
'TakeCareOfEachOther',
'TablighiJamaat',
'treats',
'StayHomeSavesLives',
'Facemask',
'APBspeaker',
'COVID19Vic',
'Americans',
'AgeingWellNZ',
'Qingdao',
```

```
'Pacientes',
'EndTheOrangMenace',
'pornaddict',
'Tests',
'HamiltonOnt',
'landing',
'M18',
'coronapanic',
'justrecoveryforall',
'アカペラ',
'Weskus',
'WorldHandHygieneDay2020',
'parentengagement',
'COVID19remdesivir',
'olderpersons',
'Liberated',
'aoda',
'OnMyMind',
'aintsobad',
'thalidomide',
'weskus',
'WearAMaskSaveALife',
'VTExclusive',
'ショックドクトリン',
'COVIDvaccines',
'PTSD',
'scotland',
'SupportLocalBusinesses',
'INDICTTHETRAITORS',
'hydroxycholoroquine',
'blacklivesmatter',
'visual',
'23May',
'londonprotests',
'DavidIcke',
'BotswanaElephants',
'evidenceecosystem',
'stayactive',
'密',
'OBR',
'sexbuddy',
'ObiWan',
'disinfectant',
'cryptocurrency',
'Telangana',
```

```
'VaccinesWork',
'Destiny',
'KindnessInScience',
'casinomylivelihood',
'SARSCOV2',
'NannyState',
'SadiqKhanResignNow',
'excited',
'TakeDownCCP',
'PostIntensiveCareSyndrome',
'stream',
'Tenet',
'duchenne',
'BorisHasFailedBritain',
'SilviaPark',
'Africa4Palestine',
'interviews',
'trumplightbulb',
'friends',
'EastLancashire',
'Disparities',
'Psychotherapist',
'ML',
'TD1',
'ThrowHerOut',
'Day28',
'leadership',
'HealthIT',
'RecallKenney',
'Arduino',
'BLMCanada',
'masksforCanada',
'ProChoice',
'bigpharma',
'Coyoacán',
'influenza',
'SocialWorker',
'flu',
'GrifterBarbie',
'Society',
'SocialCreditSystem',
'NZL',
'dominicummimgs',
'Tiziana',
'CAREMassey',
```

```
'AntiChristHaters',
'Health',
'physicalhealth',
'LoveLossLocdown',
'LiSA',
'dancemama',
'MondayMotivaton',
'SmallpoxBlankets',
'snapchat',
'Disease',
'MusicHelps',
'Encampments',
'rhino',
'momlife',
'cruel',
'VirtualRamadan',
'Covid19Out',
'regenerativetourism',
'antisocial',
'healthcareeducation',
'dietitians',
'IsraeliCrimes',
'MedEd',
'dolphins',
'covid_19usa',
'healthyfood',
'Netflix',
'corn',
'MayLongWeekend',
'tennis',
'greenspace',
'Trainees',
'Boards',
'beautiful',
'nosocialdistancinghere',
'ImranKhan',
'MasksOff',
'Breastfeeding',
'CathKidston',
'AAC',
'wanderlust',
'tubadadettolkadhulahai',
'obesity',
'DefundCBC',
'Abstract',
```

```
'TheAthleteDevelopmentShow',
'dårskap',
'Powell',
'MacEngMotivation',
'BuenosDias',
'StuckwithU',
'EDAC2020',
'SelfReg',
'shopify',
'Netherlands',
'Sevenoaks',
'GratitudeAttitude',
'KushnerKnew',
'swineflu',
'OER',
'cndpoli',
'jcphillips',
'mobilephone',
'Ticats',
'roadtoaffiliate',
'TogetheratHome',
'loaf',
'Matheson',
'Window',
'globalSouth',
'trustees',
'MyWarner',
'CovidVic',
'KimberlyGuilfoyle',
'Nepal',
'Orkney',
'BoycottChineseProducts',
'School',
'CollisionfromHome',
'EmilyMurphyGSA',
'chocolate',
'nationalthankateacherday',
'pink',
'CovidTesting',
'PureVPN',
'TheDoseCBC',
'無限大',
'SmokingGun',
'Kamala',
'hastings',
```

```
'ProtectTheVulnerable',
'PicOfTheDay',
'nomorecovid',
'communitymatters',
'wuhanvirologylab',
'dataprotection',
'agroecology',
'Recycle',
'climatestrike',
'ghoststory',
'egomaniac',
'Covid19solution',
'Football',
'CarStar',
'WetMarket',
'MCCQE2',
'allergy',
'BlackSupremacy',
'Chelsea',
'ToryCorruption',
'SmallCaps',
'CheckThenShare',
'Tudor',
'lovescotland',
'fyp',
'besmart',
'WorstHit',
'DoneWithTrump',
'happylaborday',
'algebra',
'NaturePhotography',
'scammers',
'hairdressers',
'hurricane',
'中共',
'flight',
'movements',
'covid_19uk',
'Annibyniaeth',
'beachdog',
'peur',
'Hannity',
'hikes',
'TobiasEllwood',
'kungfuschoolshastings',
```

```
'LGBTQ',
'CowardinChief',
'Lockdown',
'SocialDistanceHalloween',
'SASGF',
'backtobetter',
'trust',
'OurRemedyIsARMY',
'NationalHousingDay',
'Berlin',
'BidenCares',
'CloroxCURE',
'ChurchBazaars',
'KoronawirusWPolsce',
'fukushima',
'deanmartin',
'CGTN',
'ebmtechno',
'ThalapathyVijay',
'actorslife',
'EmmentalModel',
'PHHarriet',
'sloganbogan',
'Queenstown',
'HockeyCard',
'LockHimUp',
'instagramlive',
'VPChen',
'SelfishTrump',
'theatre',
'bread',
'BasicIncome',
'flights',
'SMEs',
'LongCovid',
'thist',
'WakeUpAmerica',
'SuperspreaderACB',
'podiatry',
'HydroxychloroquineWorks',
'ClapForTheNHS',
'FlintWaterCrisis',
'bloodclots',
'healing',
'recession',
```

```
'Sittingbourne',
'votebluer',
'HKpolice',
'CoronaVirusHOAX',
'Dying',
'CONICET',
'MichaelCohen',
'Millenials',
'buyingcontent',
'cornovirusaustralia',
'Wikipedia',
'Israeli',
'pwwnz',
'DerelictionOfDuty',
'GoldenHorseShoe',
'AfricansInChina',
'GülistanDoku',
'Liberty',
'OosterhoffResign',
'Kefe',
'inspired',
'AmericaNeedsJoeBiden',
'window',
'DonaldTrumpIsTheTypeOfGuy',
'OBAMAGATE',
'Spanish',
'UltimateWarrior',
'STEMeducation',
'ScheerStupidity',
'Trumpian',
'test',
'Wetherspoon',
'COVIDPacific',
'cisktime',
'TrumpDerangementSyndrome',
'DayofMourning',
'PatriotsInTune',
'punkrock',
'covid19rural',
'soccorritori',
'freegunfriday',
'ElderAbuse',
'WakeUp',
'DeepState',
'slp',
```

```
'designer',
'savethechildren',
'baking',
'GatesFoundation',
'ImVotingfor',
'FreeAccess',
'covd19',
'XRPCommunity',
'KeepSafe',
'probiotics',
'FrontLineHeroes',
'PenceSpreadsTheVirus',
'socialchange',
'threat',
'COVID19MA',
'ThePower',
'PCRTest',
'PresidentT',
'PhilosophyOfTimeTravel',
'TrumpsGOP',
'BoomBap',
'WECharity',
'SuperSpreaderEvent',
'Brant',
'cashier',
'ダッフィー',
'coronavirus',
'cyclical',
'LifeLines',
'Karen',
'ToryCriminalCovidMismanagement',
'PRINCIPLETrial',
'Queensland',
'BabyShark',
'LAO',
'wet',
'AirNZ',
'outdoors',
'Alberta',
'the',
'wedonotconsent',
'evicted',
'coronavirusaustralia',
'Bill_Gates',
'KendimiçinNot',
```

```
'ワーキングホリデー',
'كذبة_كورونا',
'UKpolitics',
'SundaySpotlight',
'COVIDView',
'HWDSBmothers',
'4agosto',
'Indiana',
'MentalHealthCzar',
'mask4all',
'Opinion',
'sunny',
'helpme',
'MitchMcConnell',
'NoreenTufele',
'TheyAreAlsoUs',
'RoadToll',
'GatesForPrison2020',
'banners',
'jackfruit',
'JeremyVine',
'Lmao',
'whogotbeats',
'masksforOntario',
'FiY1',
'VIBES',
'environmental',
'DELvsFDA',
'3moremonths',
'Patriots',
'clapforNHS',
'topshelf',
'Silencetrump',
'WinItWednesday',
'Osekabasheba',
'erotica',
'sinagua',
'CivicDuty',
'mondaymorning',
'PandemicParenting',
'OER4Covid',
'georgia',
'Injecting',
'costumephotography',
'Stress',
```

```
'KeepEvanSafe',
'safehands',
'couples',
'FOIA',
'whakawhetai',
'autosure',
'Steampunk',
'QudsDay',
'DemocratParty',
'atr',
'Virus2020',
'nanofibers',
'HoldFirm',
'cabinfever',
'COVIDiots',
'StayAtHome',
'4thofJuly',
'SheikhMohamedbinZayed',
'Day34ofLockdown',
'alien',
'AyudaEnEspanol',
'P5',
'CovidLikesThis',
'finalfantasyxii',
'disproportionate',
'GeoPolitiks',
'Breathe',
'Sussex',
'covidNZ',
'TruthMatters',
'WalkAwayCampaign',
'winteriscoming',
'flowerphotography',
'nzpol',
'SCOTUSNominee',
'evacuation',
'IgboYouths',
'ChinaVirus',
'LongHaul',
'RacialMinorities',
'NDLB2020',
'ClimateCrises',
'Bacon',
'Veteranspouses',
'doubledotsquash',
```

```
 'MilesCoffee',
 'CardiacSurgery',
 'daughter',
 'CommunityEngagement',
 'SibStories',
 'USSRoosevelt',
 'COVIDpumpkin',
 'SRHR',
 'NowPlaying',
 'monochrome',
 ...}
```

In [67]:
```
# Creating subsets of dataset for new zealand

df_canada = df[df.country == 'canada']
df_canada
```

Out[67]:

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_word_count
80000	The decision to visit with family or friends d...	NaN	False	True	COVID	canada	294	40
80001	Under Trump's erratic mishandling of the #coro...	NaN	False	True	coronavirus	canada	303	42
80002	Around the world critical events are being ove...	NaN	True	False	COVID19	canada	279	47
80003	Visiting our parks this weekend? Please #Stay...	NaN	False	True	StaySafe socialdistancing	canada	254	36
80004	Join us for a live session that will focus on ...	NaN	False	True	Covid19	canada	289	42
...	...	...	...	...	...	...	...	...
119995	It's great to have a hopeful perspective from ...	NaN	True	False	COVID19 healthcare hope	canada	108	17
119996	An important article reporting that #SARSCoV2 ...	NaN	False	True	SARSCoV2	canada	297	34

	text	reply_to_screen_name	is_quote	is_retweet	hashtags	country	text_char_count	text_word_count
119997	Stay safe this #BonfireNight2020 🎆 \n\nWith org...	NaN	False	True	BonfireNight2020 Covid19	canada	299	45
119998	195 new cases of #COVID19 reported;\n\n80-Lago...	NaN	False	True	COVID19	canada	299	39
119999	Discussion avec @BillGates sur le financement ...	NaN	False	True	coronavirus	canada	170	24

40000 rows × 13 columns

In [68]:
```
hashtag_arr_canada = []

for hashtag in df_canada.hashtags:
    hashtag_arr_temp = hashtag.split(' ')
    for hashtag_temp in hashtag_arr_temp:
        hashtag_arr_canada.append(hashtag_temp)

unique_hashtags_canada = set(hashtag_arr_canada)
unique_hashtags_canada
```

Out[68]:
```
{'Islamophobia',
 'PSPCFamily',
 'MushroomPowder',
 'BANG',
 'doublytragic',
 'smallbusinesses',
 'TestShortage',
 'inflammasomes',
 'CIHR',
 'FTE',
 'Blanquer',
 'workingfromhome',
 'Stupid',
 'closeChinasBorders',
 'scaleups',
 'fitboy',
 'TrumpKnewVoteBlue',
 'Qanons',
 'RunChat',
 'Canadá',
 'hypocritepelosi',
 'EdBartlett',
 'Father',
 'studentsvspandemics',
 'poll',
 'CivilServiceLive',
 'workplace',
 'makeup',
 'WorldEnvironmentDay2020',
 'sunset',
 'HLINblog',
 'Innovation',
 'cityofpg',
 'désopasdéso',
 'Tidy',
 'newbeginnings',
 'Cannabis',
 'MartialLaw',
 'postapocalyptic',
 'icantbreathe',
 'Impeachment',
 'TechForGood',
 'ManchesterMarch',
 'bombings',
 '808day',
```

```
'DragRace',
'InfectiousDiseases',
'CaptainTomMoore',
'mystery',
'PrivatizationFailure',
'Vape',
'DementiaDon',
'YouLost',
'weekendreads',
'UKHeatWave',
'grammar',
'wreathsofinstagram',
'premiumsnapchat',
'bisquesoup',
'nude',
'pathologist',
'whattowatchonnetflix',
'onlineclasses',
'installers',
'PLZ',
'Kolkata',
'NCYT',
'PHAC',
'cornavirus',
'élèves',
'SpawnInLaw',
'コスプレ',
'istandwithdan',
'babyboss',
'onfr',
'3in4',
'Deal',
'agricultural',
'influencers',
'AECOPD',
'sackDominic',
'NorthWales',
'Peckham',
'gaytwitter',
'promotion',
'MDBs',
'covidhoax',
'Sars',
'authors',
'MillionMAGAMarch',
```

```
'baobesity',
'Milton',
'Installations',
'ResilientLeadership',
'makelotsofmoney',
'furchild',
'expert',
'gkunion',
'sanctionskill',
'UnitedKingdom',
'TheArtsMatter',
'HonourBasedAbuse',
'LestWeForget',
'gnome',
'migrant',
'GretaThunberg',
'cocktailhour',
'bunkerbabytrump',
'collection',
'MaskOn',
'rent',
'COVID45',
'WordPress',
'Humanity',
'danandshay',
'FBPE',
'NWT',
'fails',
'USElections',
'Chantilly',
'covid1984',
'freedom',
'surfing',
'TrumpChump',
'Calculus',
'cancelanimalag',
'Socialism',
'covid19vaccine',
'OttawaGolfExaminer',
'ProtecttheVulnerable',
'Blacklivesmatter',
'CoVId19',
'CATS',
'SaudiTrump',
'UnsafeSeptember',
```

```
'product',
'makeitmakesense',
'epic',
'StonerLife',
'veday75thanniversary',
'PhysicalDistancing',
'AspenMedical',
'MyHighSt',
'UnMask',
'Naturebasedsolutions',
'JonathanSwan',
'fordnation',
'zimbabwe',
'MentallyHealthySchools',
'mehdi',
'Boston',
'hamilton',
'mentalillness',
'Nutrition',
'BritishColunbia',
'CLWR',
'coronavirusbc',
'Día77',
'WorkingClass',
'giftcard',
'RSV',
'Uyghur',
'IStandWithFauci',
'ReleaseTheRussiaReport',
'funathome',
'deserves',
'autoerotism',
'WhackAMole',
'masques',
'menstyle',
'stigma',
'HappySummerDay',
'OrangeShitGibbon',
'JobSeeker',
'SackDanAndrews',
'Justintrudeau',
'longweekend',
'horseshoecrab',
'inspirational',
'routine',
```

```
'Mizoram',
'Cartoons',
'medstudenttwitter',
'larnaca',
'cdnpoli',
'SystemicAgeism',
'dad',
'heatwaves',
'antiviral',
'FullFact',
'safetysolutions',
'TrumpKnewAndDidNothing',
'axelhappy',
'TrumpVirus',
'Herefordshire',
'DollyParton',
'decriminalization',
'BidenWasNotElected',
'RMTs',
'BCElection2020',
'usaCoronavirus',
'Pride2020',
'LiberalStaff',
'Politician',
'schools',
'texturehuntergatherer',
'who',
'No10',
'larriesarehot',
'thankateacherday',
'CRYPTO',
'WomenInSport',
'Kyiv',
'FeedScarborough',
'NamingTheLost',
'covidbc',
'RedAlert',
'FriendshipDay',
'gives',
'cyberrisk',
'CancelAnimalAg',
'BTR',
'Hotels2Homes',
'whitehousebriefing',
'FarceMasks',
```

```
'DerangedDonald',
'tell',
'OrangeManBad',
'LordOfTheRings',
'Budget2021',
'reentry',
'motd2',
'PandemicsReport',
'anxietycure',
'CovidLonghaul',
'Stage2',
'YouthVerdict',
'removethecap',
'londonprotest',
'Leduc',
'SCAM',
'CIV',
'MentalHealthAwarenessWeek2020',
'Covidlife',
'Immunity',
'ACTogether',
'ChildTrafficking',
'BBNaijaReunion2020',
'Saskatchewan',
'girlgirl',
'jobretentionscheme',
'DilDilPakistan',
'cyclosporine',
'InsécuritéAlimentaire',
'Corona',
'Soundcloud',
'Democracy',
'OUTNOW',
'Friends4Ever',
'ToryBritain',
'BillNye',
'GranfondoWhistler',
'EffectiveMultilateralism',
'VOTELayconNonStop',
'ZimbabweanLivesMatter',
'fooddonations',
'ceramicsstudio',
'NayibBukele',
'manifencours',
'kingsheadpub',
```

```
'StuckwithUVideo',
'arts',
'LOA',
'TrumpCrimeSyndicate',
'Cummgate',
'Live',
'breastreconstruction',
'IDWeek2020',
'WuhanCoronavirus',
'Navigli',
'MayorsQuestions',
'ADT',
'davidovsburnaboy',
'neighbourhoodbarbershop',
'mustfollow',
'whore',
'protection',
'SEC',
'PRESIDENTTRUMP',
'day56oflockdown',
'CovidScam',
'Reach',
'socceraid',
'JaCovid19',
'Buffett',
'ReliefForAll',
'cellphone',
'MortalityRate',
'art',
'CorruptJoe',
'oldbayseasoning',
'BlacklivesMaters',
'languesofficielles',
'トロント',
'Patient',
'artenquarantaine',
'Ireland',
'HS2',
'laval',
'bisexual',
'Queen',
'LetsTryUBI',
'BigScottishBookClub',
'funnymemes',
'FundingOpportunity',
```

```
'murderhornets',
'themusictechguyuk',
'SNACovid19Awareness',
'LondonAfterPeople',
'walk',
'teacherhorizons',
'caronavirusoutbreak',
'NotALeaderJustALiberal',
'Humanitarian',
'DerekSloan',
'DoggingTales',
'keeplaughing',
'brackenfellhigh',
'race',
'YULIN',
'covertspy',
'Vaccines2020',
'NengiXShoesByFlora',
'トランスフォーマー',
'SAVIEZVOUS',
'7-Jul',
'FrancoisLegault',
'Gofundme',
'ABhealth',
'screwedbytheRCN',
'DisiplinWajibMasker',
'Cardiff',
'Marks',
'5D',
'aircargo',
'Costco',
'trends',
'Light',
'LloydsPeople',
'Maroc',
'tiktokteens',
'Medicines',
'Gloucestershire',
'Covidmarshall',
'Footwear',
'saultnews',
'LOwLAtency',
'Aerosol',
'COVIDAlertApp',
'FREE',
```

```
'jaddm',
'WHO',
'eidmubarak2020',
'covidcomedy',
'CECRA',
'ChinWag',
'sundayfeels',
'HEOR',
'lockdowner',
'catchingupwithfriends',
'MediaFreedom',
'NoWhereIsSafe',
'TwitterPoll',
'TheReidOut',
'ToryGenocide',
'dougford',
'NEWS',
'wtpBlue',
'InsightIncolor',
'lifestyles',
'Desantis',
'Biohackers',
'atlas',
'farleft',
'BroncosCountry',
'sweet',
'upcycler',
'StLouisCardinals',
'WeareTogether',
'JacobReesMogg',
'stress',
'BLAME',
'TourDeThanks',
'InternationalDayOfLivingTogetherInPeace',
'Liban',
'working_conditions',
'Collaboration',
'sexadvice',
'customise',
'Vienna',
'Brazil',
'kingcollege',
'abc730',
'Caritastic',
'covidsigns',
```

```
'modernslavery',
'WWEEliteSquad',
'ShineOn',
'oursupport',
'UNO',
'豆原一成',
'CCANationalConference',
'BreastScreen',
'AskCBCNews',
'封じ込め',
'WFPCaribbean',
'WiFi',
'Salud',
'GoodMorning',
'vicfabe',
'CRISPR',
'booforboris',
'covidmaternity',
'minutesilence',
'Claygate',
'acil',
'HappyWednesday',
'goingabroad',
'新型コロナウイルス',
'Tassie',
'mhpss',
'UniversalCredit',
'financialadvisor',
'NoJudticeNoPeace',
'Covid19Georgia',
'StayHome',
'LeCapitaineIvre',
'university',
'PeerReview',
'fitness',
'STD',
'MediaBriefingCOVID19SA',
'OneVoice1',
'SleepyJoeBiden',
'N95',
'webcoic',
'raredigitalart',
'LQvMS',
'extremeweather',
'demand',
```

```
'Europa',
'AmigaDateCuenta',
'CDL',
'Rebellion',
'COVIDmyths',
'doctoral',
'Goats',
'ndg',
'TrumpSuperSpreader',
'GaryMacKayDay',
'Palestinian',
'darahuang',
'SwiftCurrent',
'PeerRevWeek20',
'mytopfans',
'Annedemic',
'amauk',
'InternationalDayofPersonsWithDisabilities',
'spike',
'Fatidabad',
'Scientists4Assange',
'salah',
'TeamCanada',
'britishairways',
'boozeban',
'Cuba',
'halloween',
'MenstruationDayParade',
'frightening',
'pamdemic',
'death',
'oldfriends',
'Laboratories',
'SpreadTheWord',
'OUResearch',
'SendSmileZ',
'kits',
'Megaproject',
'commonsense',
'KeepYourChildHome',
'Howrah',
'StopLoiSecuriteGlobale',
'SpiceGirls',
'WorldChocolateDay',
'FletcherDean',
```

```
'FipAcademicLive',
'ConfinementJour53',
'StateofPossible',
'mother',
'MayoClinic',
'tiktok',
'CoronavirusItalia',
'filmmaking',
'RSMClub',
'insurtech',
'bojo',
'tory',
'GlobalDayofSolidarity',
'Translink',
'NHSWorkersSayNO',
'AskListenDo',
'CDC',
'Germany',
'WeGotThisWA',
'Cannabidiol',
'dayofintegrity',
'ASM',
'KyivNotKiev',
'wearenotgoingaway',
'climatechange',
'WilliamShakespeare',
'Southfields',
'Inuit',
'hydroxychroloquine',
'reliefprint',
'deficits',
'FreddieMercury',
'ParksLife',
'OToole',
'Cristobal',
'KempKILLS',
'DARBAR_Ni_Deli',
'TimeTogether',
'geographyfromhome',
'RevelsConnects',
'SpanishFlu',
'QuarantineRadio',
'covid19testkits',
'EMFs',
'familyfirms',
```

```
'拡散希望',
'PS752',
'WeAreSorryNengi',
'Archive',
'services',
'DurhamRiding',
'histoire',
'ImpeachBarrNOW',
'PhysEd',
'pstrilla',
'parenthood',
'commute',
'proinfection',
'9News',
'Colcorona',
'fitfam',
'OCD',
'80s',
'SkeemSaam',
'rtept',
'flamengo',
'Venezuela',
'Coronavirusindonesia',
'EarthDay50',
'IranUncovered',
'emotional',
'Minęła20',
'Taliban',
'BPD',
'Honduras',
'whitehousevirus',
'safety',
'smallbusinesstips',
'excellent',
'software',
'mRNA',
'FRIENDS',
'cooking',
'sw',
'YouKnowHowToMakePeopleSmile',
'resisters',
'westandwiththesector',
'sincronizzazione',
'reason',
'Nurse',
```

```
'eosinophilia',
'schoolsreopening',
'gouvcan',
'coronavirusbriefing',
'Seniors',
'letter',
'parathyroid',
'ClintonFoundation',
'Winter',
'lockdownextended',
'bigboobs',
'OVERDOSEÔBAILE',
'director',
'greetingcards',
'popularity',
'covidopening',
'Conley',
'trinitybellwoodspark',
'APAC',
'phd',
'ConstitutionalRights',
'camwhore',
'onlineshopping',
'whiterockpier',
'covidinquirynow',
'COVID19outbreak',
'1stGUNDAM',
'UFHneedshelp',
'CivilSociety',
'technocracy',
'www',
'totalharms',
'communitystrong',
'ChurchEnd',
'PRIMERDESINGLTD',
'1ofTheMillion',
'onlyfansbabe',
'here',
'china',
'vaping',
'ABgov',
'CanadaRecovery',
'dollar',
'verses',
'RandPaul',
```

```
'StayTheFuckHome',
'WeLoveNHS',
'puzzlelover',
'Whistleblowers',
'Branding',
'provincialparks',
'JonSnow',
'ArtistoftheSummer',
'BEER',
'stgeorgesday',
'NewsMax',
'PoliticalLogics',
'quarantinehealth',
'KamalaHarrisForVP',
'scales',
'wesingusinguk',
'CoronaVirusHoax',
'traitor',
'CompassionateRelease4Reality',
'state',
'ingresomínimo',
'ubistimulus',
'redbubbleartist',
'sick',
'NetflixCAJun20',
'Drilly4Mayor',
'frontlineheroes',
'everythingisfine',
'sackcunmings',
'CRISPS',
'BTS',
'financialdisputes',
'disarmament',
'SleepyEyes',
'ColonialHeights',
'tuesdayvibe',
'WednesdayThoughts',
'VWFC',
'supportindieartists',
'Reynosa',
'Burnaboy',
'beer',
'LordLamont',
'Bojo',
'Muscat',
```

```
'coal',
'COVIDVaccine',
'TablighiJamaat',
'TuboMask',
'medicosporlaverdad',
'bloggerstribe',
'GeneralFlynn',
'parrot',
'NottingHillCarnival',
'FutureFund',
'Facemask',
'PhoKingBon',
'COVID19Vic',
'jobkeeper',
'FasesDesescalada',
'Americans',
'MJAInSight',
'HospitalLiveBBC',
'QatarAirways',
'Qingdao',
'Grammy',
'SackBoris',
'liquidità',
'KeepCalm',
'ChildrensAid',
'wearing',
'UNPeacekeepersDay',
'cachorros',
'UMQ',
'LekkiMassacre',
'UltrasOnly',
'residence',
'WearAMaskSaveALife',
'JohnChayka',
'borishasfailedtheuk',
'JADecides2020',
'TheFailedPresident',
'johnnydepp',
'socio',
'scotland',
'PTSD',
'blacklivesmatter',
'الحجر_المنزلي',
'covid19science',
'inhaler',
```

```
'covidrecovery',
'londonprotests',
'Sweatpants',
'pingpong',
'ExcludeUsFromBan',
'DavidIcke',
'2hrsNonStopJamz',
'interfaith',
'stayactive',
'AirCanada',
'AffordableHousing',
'urbaninnovation',
'angry',
'TEMAS',
'cjrs',
'PattersonLakes',
'postcards',
'Ahmedabad',
'disinfectant',
'cryptocurrency',
'Telangana',
'VaccinesWork',
'Gas',
'SARSCOV2',
'myKHSC',
'Diplomacy',
'excited',
'cndimm',
'stream',
'BorisHasFailedBritain',
'TimeToMove',
'LiberalHypocrisy',
'ygkyouth',
'ThursdayMotivationJun20',
'AlertaCOVID19',
'ReporteDiario',
'shame',
'USMCA',
'friends',
'layoffs',
'Vehicles',
'ML',
'50swomen',
'MikeHucabee',
'ChildLabor',
```

```
'ThanksBrett',
'wexit',
'Immunosuppressed',
'leadership',
'HealthIT',
'Asexual',
'RecallKenney',
'vids',
'CPolíticaAL',
'TDL10',
'FrontLine',
'influenza',
'perks',
'Trikafta',
'USAelection2020',
'flu',
'PlayinthePandemic',
'PayItForward',
'토론토맛집',
'EXHIBITION',
'UpbeatLockdown',
'YCDAB',
'dominicummimgs',
'MOTIVATIONMONDAY',
'Medien',
'MetrolandFund',
'Sick',
'newjob',
'Health',
'CountryMusic',
'sihatmilikku',
'VoteRedToSaveAmerica',
'BackToWorkSafely',
'MondayMotivaton',
'negazionisti',
'tellandbegoodnews',
'snapchat',
'Disease',
'ArjunKapoor',
'Iveson',
'TranscendProduction',
'theprayer',
'Bilbao',
'LGBTQI',
'rhino',
```

```
'decorcomplete',
'momlife',
'marketupdate',
'cruel',
'ransomwareattacks',
'WeAreSyria',
'madeinChina',
'wrdsb',
'letsfightcoronavirus',
'VirtualRamadan',
'Covid19Out',
'artawesome',
'Phlebotomy',
'shows',
'lnh',
'BBCPM',
'tobeornottobe',
'machinelearning',
'discountcode',
'MedEd',
'Riverside',
'avemaria',
'TrustBarometer',
'shadows',
'coronavirusafrica',
'Lovelondon',
'divineguidance',
'Netflix',
'FullLengthArticle',
'Euronomani',
'BAPSLuton',
'italia',
'sjw',
'greenspace',
'WithRohingya',
'Forecasting',
'Boards',
'covidcon',
'dnb',
'hair',
'beautiful',
'Eton',
'covid19leadership',
'SurvivalOfTheFittest',
'NewsCorp',
```

```
'MáximaDisciplina',
'ImranKhan',
'MasksOff',
'Breastfeeding',
'smurfs',
'JTTT',
'IWS',
'keepingbreakfastgoing',
'ChiefMedicalOfficer',
'Staycation2020',
'IndigenousServices',
'LastWord',
'Kansas',
'athomepedicure',
'keepgymsopen',
'Pretty',
'quebec',
'CoronavirusRDC',
'WhitePrivilege',
'obesity',
'ClubBeatzAtHome',
'GPAB',
'eu',
'Covidaus',
'UKSports',
'MacEngMotivation',
'IMFP2020',
'StuckwithU',
'BarnardCastle',
'SLAMinutes',
'iboshow',
'rockandroll',
'cockatoo',
'Netherlands',
'coventrycreates',
'Sevenoaks',
'vanmorrison',
'CAOT',
'USpolitics',
'شي_إن_رمضان',
'ravulizumab',
'Books',
'cndpoli',
'Universities',
'livewebinar',
```

```
'mobilephone',
'thoughtoftheday',
'Adivasis',
'TogetheratHome',
'RMFansEnCasa',
'WeAreWatching',
'Window',
'LATENIGHT',
'Yegcc',
'covid19prevention',
'trochu',
'from',
'personalliberties',
'Guide',
'pharmasupplychain',
'TRANSFORM',
'AboriginalLaw',
'AffiliateMarketing',
'CoadyGrad',
'CovidVic',
'reintroduce',
'BidenCabinet',
'orilliayouthcentre',
'whooping',
'littératie',
'CONservative',
'Nepal',
'ProtectOurStudents',
'Angrynomics',
'School',
'Honey',
'chocolate',
'Orkney',
'Cranbourne',
'Brunch',
'EmilyMurphyGSA',
'ArtsFunding',
'CovidTesting',
'Wembley',
'info',
'tlmep',
'TheDoseCBC',
'CDNBusinesses',
'無限大',
'UBACares',
```

```
'AsianMurderHornets',
'OW',
'اعدام',
'AIEducation',
'ShutUpChallenge',
'UKSPINE',
'planned',
'Belgique',
'SkyNews',
'barnardcastle',
'dataprotection',
'donateyourdevice',
'Niger',
'agroecology',
'climatestrike',
'umf',
'whatdoyoumeme',
'ridelots',
'बंगाली',
'DanishMaskStudy',
'FurFreeBritain',
'SmartFoodPlanner',
'allergy',
'narratives',
'MCCQE2',
'FillYourHeartWithIreland',
'ToryCorruption',
'Attack',
'Twilio',
'politicalcartoon',
'Renewables',
'Tudor',
'Aboriginal',
'VHC',
'تعز',
'fyp',
'BN',
'brexittradetalks',
'AgCan',
'Assad_Genocide',
'GladysMustGo',
'vacunacovid',
'Cafés',
'humancapital',
'CONTE',
```

```
 'SaturdayNight',
 'scammers',
 'hairdressers',
 'VEALive',
 'hurricane',
 'Covid19AB',
 'flight',
 'ToryTouretteSyndrome',
 'billion',
 'autopsy',
 ...}
```

In [69]:
```
# Count vectorize the hastags extracted from us tweets

unique_hastags_str = ' '.join([str(elem) for elem in unique_hastags])
unique_hashtags_us_str = ' '.join([str(elem) for elem in unique_hashtags_us])

documents = [unique_hastags_str, unique_hashtags_us_str]

# Create the Document Term Matrix
count_vectorizer = CountVectorizer(stop_words='english')
count_vectorizer = CountVectorizer()
sparse_matrix = count_vectorizer.fit_transform(documents)

# Convert Sparse Matrix to Pandas Dataframe if you want to see the word frequencies.
doc_term_matrix = sparse_matrix.todense()
cv_us = pd.DataFrame(doc_term_matrix,
                     columns=count_vectorizer.get_feature_names_out(),
                     index = ['all_hashtags', 'us_hashtags']).iloc[[1]]
cv_us
```

Out[69]:

	02jul20	0613fm_0509	06strong	10	1000families	1000islands	1000lives	1000names	1000opportunities	1000poe
us_hashtags	0	1	1	0	0	0	0	0	0	

1 rows × 54939 columns

In [70]:
```
# Count vectorize the hastags extracted from uk tweets

unique_hastags_str = ' '.join([str(elem) for elem in unique_hastags])
unique_hashtags_uk_str = ' '.join([str(elem) for elem in unique_hashtags_uk])

documents = [unique_hastags_str, unique_hashtags_uk_str]
```

```
# Create the Document Term Matrix
count_vectorizer = CountVectorizer(stop_words='english')
count_vectorizer = CountVectorizer()
sparse_matrix = count_vectorizer.fit_transform(documents)

# Convert Sparse Matrix to Pandas Dataframe if you want to see the word frequencies.
doc_term_matrix = sparse_matrix.todense()
cv_uk = pd.DataFrame(doc_term_matrix,
                     columns=count_vectorizer.get_feature_names_out(),
                     index = ['all_hashtags', 'uk_hashtags']).iloc[[1]]
cv_uk
```

Out[70]:

	02jul20	0613fm_0509	06strong	10	1000families	1000islands	1000lives	1000names	1000opportunities	1000poe
uk_hashtags	0	0	0	0	1	0	1	1	1	

1 rows × 54939 columns

In [71]:

```
# Count vectorize the hastags extracted from australia tweets

unique_hastags_str = ' '.join([str(elem) for elem in unique_hastags])
unique_hashtags_australia_str = ' '.join([str(elem) for elem in unique_hashtags_australia])

documents = [unique_hastags_str, unique_hashtags_australia_str]

# Create the Document Term Matrix
count_vectorizer = CountVectorizer(stop_words='english')
count_vectorizer = CountVectorizer()
sparse_matrix = count_vectorizer.fit_transform(documents)

# Convert Sparse Matrix to Pandas Dataframe if you want to see the word frequencies.
doc_term_matrix = sparse_matrix.todense()
cv_australia = pd.DataFrame(doc_term_matrix,
                     columns=count_vectorizer.get_feature_names_out(),
                     index = ['all_hashtags', 'australia_hashtags']).iloc[[1]]
cv_australia
```

Out[71]:

	02jul20	0613fm_0509	06strong	10	1000families	1000islands	1000lives	1000names	1000opportunities	10(
australia_hashtags	0	0	0	0	0	0	0	0	0	

1 rows × 54939 columns

In [72]:
```
# Count vectorize the hastags extracted from ireland tweets

unique_hastags_str = ' '.join([str(elem) for elem in unique_hastags])
unique_hashtags_ireland_str = ' '.join([str(elem) for elem in unique_hashtags_ireland])

documents = [unique_hastags_str, unique_hashtags_ireland_str]

# Create the Document Term Matrix
count_vectorizer = CountVectorizer(stop_words='english')
count_vectorizer = CountVectorizer()
sparse_matrix = count_vectorizer.fit_transform(documents)

# Convert Sparse Matrix to Pandas Dataframe if you want to see the word frequencies.
doc_term_matrix = sparse_matrix.todense()
cv_ireland = pd.DataFrame(doc_term_matrix,
                      columns=count_vectorizer.get_feature_names_out(),
                      index = ['all_hashtags', 'ireland_hashtags']).iloc[[1]]
cv_ireland
```

Out[72]:

	02jul20	0613fm_0509	06strong	10	1000families	1000islands	1000lives	1000names	1000opportunities	1000
ireland_hashtags	0	0	0	0	0	0	0	0	0	

1 rows × 54939 columns

In [73]:
```
# Count vectorize the hastags extracted from new_zealand tweets

unique_hastags_str = ' '.join([str(elem) for elem in unique_hastags])
unique_hashtags_new_zealand_str = ' '.join([str(elem) for elem in unique_hashtags_new_zealand])

documents = [unique_hastags_str, unique_hashtags_new_zealand_str]

# Create the Document Term Matrix
count_vectorizer = CountVectorizer(stop_words='english')
count_vectorizer = CountVectorizer()
```

```
sparse_matrix = count_vectorizer.fit_transform(documents)

# Convert Sparse Matrix to Pandas Dataframe if you want to see the word frequencies.
doc_term_matrix = sparse_matrix.todense()
cv_new_zealand = pd.DataFrame(doc_term_matrix,
                      columns=count_vectorizer.get_feature_names_out(),
                      index = ['all_hashtags', 'new_zealand_hashtags']).iloc[[1]]
cv_new_zealand
```

Out[73]:

	02jul20	0613fm_0509	06strong	10	1000families	1000islands	1000lives	1000names	1000opportunities
new_zealand_hashtags	0	0	0	0	0	0	0	0	0

1 rows × 54939 columns

In [74]:

```
# Count vectorize the hastags extracted from canada tweets

unique_hastags_str = ' '.join([str(elem) for elem in unique_hastags])
unique_hashtags_canada_str = ' '.join([str(elem) for elem in unique_hashtags_canada])

documents = [unique_hastags_str, unique_hashtags_canada_str]

# Create the Document Term Matrix
count_vectorizer = CountVectorizer(stop_words='english')
count_vectorizer = CountVectorizer()
sparse_matrix = count_vectorizer.fit_transform(documents)

# Convert Sparse Matrix to Pandas Dataframe if you want to see the word frequencies.
doc_term_matrix = sparse_matrix.todense()
cv_canada = pd.DataFrame(doc_term_matrix,
                      columns=count_vectorizer.get_feature_names_out(),
                      index = ['all_hashtags', 'canada_hashtags']).iloc[[1]]
cv_canada
```

Out[74]:

	02jul20	0613fm_0509	06strong	10	1000families	1000islands	1000lives	1000names	1000opportunities	100(
canada_hashtags	1	0	0	2	0	1	0	0	0	

1 rows × 54939 columns

In [75]:

```
# Cosine similarity between hashtags of different country pairs
```

```
print("US & UK =", sklearn.metrics.pairwise.cosine_similarity(cv_us, cv_uk))
print("US & Australia =", sklearn.metrics.pairwise.cosine_similarity(cv_us, cv_australia))
print("US & Ireland =", sklearn.metrics.pairwise.cosine_similarity(cv_us, cv_ireland))
print("US & New Zealand =", sklearn.metrics.pairwise.cosine_similarity(cv_us, cv_new_zealand))
print("US & Canada =", sklearn.metrics.pairwise.cosine_similarity(cv_us, cv_canada))

print("UK & Australia =", sklearn.metrics.pairwise.cosine_similarity(cv_uk, cv_australia))
print("UK & Ireland =", sklearn.metrics.pairwise.cosine_similarity(cv_uk, cv_ireland))
print("UK & New Zealand =", sklearn.metrics.pairwise.cosine_similarity(cv_uk, cv_new_zealand))
print("UK & Canada =", sklearn.metrics.pairwise.cosine_similarity(cv_uk, cv_canada))

print("Australia & Ireland =", sklearn.metrics.pairwise.cosine_similarity(cv_australia, cv_ireland))
print("Australia & New Zealand =", sklearn.metrics.pairwise.cosine_similarity(cv_australia, cv_new_zealand))
print("Australia & Canada =", sklearn.metrics.pairwise.cosine_similarity(cv_australia, cv_canada))

print("Ireland & New Zealand =", sklearn.metrics.pairwise.cosine_similarity(cv_ireland, cv_new_zealand))
print("Ireland & Canada =", sklearn.metrics.pairwise.cosine_similarity(cv_ireland, cv_canada))

print("New Zealand & Canada =", sklearn.metrics.pairwise.cosine_similarity(cv_new_zealand, cv_canada))
```

```
US & UK = [[0.39604837]]
US & Australia = [[0.3914101]]
US & Ireland = [[0.33044264]]
US & New Zealand = [[0.3955574]]
US & Canada = [[0.3706462]]
UK & Australia = [[0.39789377]]
UK & Ireland = [[0.38926298]]
UK & New Zealand = [[0.40640613]]
UK & Canada = [[0.438726]]
Australia & Ireland = [[0.3465078]]
Australia & New Zealand = [[0.40683859]]
Australia & Canada = [[0.40633968]]
Ireland & New Zealand = [[0.36270065]]
Ireland & Canada = [[0.36642889]]
New Zealand & Canada = [[0.41508126]]
```

In [76]:
```
data = {'US': [1, 0.39604837, 0.3914101, 0.33044264, 0.3955574, 0.3706462],
        'UK': [0.39604837, 1, 0.39789377, 0.38926298, 0.40640613, 0.438726],
        'Australia': [0.3914101, 0.39789377, 1, 0.3465078, 0.40683859, 0.40633968],
        'Ireland': [0.33044264, 0.38926298, 0.3465078, 1, 0.36270065, 0.36642889],
        'New Zealand': [0.3955574, 0.40640613, 0.40683859, 0.36270065, 1, 0.41508126],
        'Canada': [0.3706462, 0.438726, 0.40633968, 0.36642889, 0.41508126, 1]}

# Creates pandas DataFrame.
```

```
df = pd.DataFrame(data, index=['US',
                               'UK',
                              'Australia',
                              'Ireland',
                              'New Zealand',
                              'Canada'])

# print the data
df
```

Out[76]:

	US	UK	Australia	Ireland	New Zealand	Canada
US	1.000000	0.396048	0.391410	0.330443	0.395557	0.370646
UK	0.396048	1.000000	0.397894	0.389263	0.406406	0.438726
Australia	0.391410	0.397894	1.000000	0.346508	0.406839	0.406340
Ireland	0.330443	0.389263	0.346508	1.000000	0.362701	0.366429
New Zealand	0.395557	0.406406	0.406839	0.362701	1.000000	0.415081
Canada	0.370646	0.438726	0.406340	0.366429	0.415081	1.000000

In [77]:

```
df.style.background_gradient(cmap ='viridis')\
        .set_properties(**{'font-size': '13px'})
```

Out[77]:

	US	UK	Australia	Ireland	New Zealand	Canada
US	1.000000	0.396048	0.391410	0.330443	0.395557	0.370646
UK	0.396048	1.000000	0.397894	0.389263	0.406406	0.438726
Australia	0.391410	0.397894	1.000000	0.346508	0.406839	0.406340
Ireland	0.330443	0.389263	0.346508	1.000000	0.362701	0.366429
New Zealand	0.395557	0.406406	0.406839	0.362701	1.000000	0.415081
Canada	0.370646	0.438726	0.406340	0.366429	0.415081	1.000000

In [ ]: