Virus (/github/radhika-khandelwal/Virus/tree/master) / Twitter_Virus.ipynb (/github/radhika-khandelwal/Virus/tree/master/Twitter_Virus.ipynb)

In [1]:
```
from twitterscraper import query_tweets
import datetime as dt
import pandas as pd
```

```
INFO: {'User-Agent': 'Mozilla/5.0 (Windows NT 5.2; RW; rv:7.0a1) Gecko/20091211 SeaMonkey/9.23a1pre'}
```

In [2]:
```
!pip install twitterscraper
```

```
Requirement already satisfied: twitterscraper in /anaconda3/lib/python3.7/site-packages (1.4.0)
Requirement already satisfied: lxml in /anaconda3/lib/python3.7/site-packages (from twitterscraper) (4.3.4)
Requirement already satisfied: requests in /anaconda3/lib/python3.7/site-packages (from twitterscraper) (2.22.0)
Requirement already satisfied: billiard in /anaconda3/lib/python3.7/site-packages (from twitterscraper) (3.6.3.0)
Requirement already satisfied: bs4 in /anaconda3/lib/python3.7/site-packages (from twitterscraper) (0.0.1)
Requirement already satisfied: coala-utils~=0.5.0 in /anaconda3/lib/python3.7/site-packages (from twitterscraper)
(0.5.1)
Requirement already satisfied: idna<2.9,>=2.5 in /anaconda3/lib/python3.7/site-packages (from requests->twitterscr
aper) (2.8)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /anaconda3/lib/python3.7/site-packages (from requests->twi
tterscraper) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /anaconda3/lib/python3.7/site-packages (
from requests->twitterscraper) (1.24.2)
Requirement already satisfied: certifi>=2017.4.17 in /anaconda3/lib/python3.7/site-packages (from requests->twitte
rscraper) (2020.4.5.1)
Requirement already satisfied: beautifulsoup4 in /anaconda3/lib/python3.7/site-packages (from bs4->twitterscraper)
 (4.7.1)
Requirement already satisfied: soupsieve>=1.2 in /anaconda3/lib/python3.7/site-packages (from beautifulsoup4->bs4-
>twitterscraper) (1.8)
```

In [3]:
```
begin_date = dt.date(2020,1,1)
end_date = dt.date(2020,4,23)

limit = 100
lang = "english"

tweets = query_tweets("virus", begindate=begin_date, enddate=end_date, limit=limit, lang=lang)
df = pd.DataFrame(t.__dict__ for t in tweets)
```

```
INFO: queries: ['virus since:2020-01-01 until:2020-01-06', 'virus since:2020-01-06 until:2020-01-12', 'virus since
:2020-01-12 until:2020-01-17', 'virus since:2020-01-17 until:2020-01-23', 'virus since:2020-01-23 until:2020-01-29
', 'virus since:2020-01-29 until:2020-02-03', 'virus since:2020-02-03 until:2020-02-09', 'virus since:2020-02-09 u
ntil:2020-02-15', 'virus since:2020-02-15 until:2020-02-20', 'virus since:2020-02-20 until:2020-02-26', 'virus sin
ce:2020-02-26 until:2020-03-03', 'virus since:2020-03-03 until:2020-03-08', 'virus since:2020-03-08 until:2020-03-
14', 'virus since:2020-03-14 until:2020-03-20', 'virus since:2020-03-20 until:2020-03-25', 'virus since:2020-03-25
 until:2020-03-31', 'virus since:2020-03-31 until:2020-04-06', 'virus since:2020-04-06 until:2020-04-11', 'virus s
INFO: Querying virus since:2020-01-06 until:2020-01-12
INFO: Querying virus since:2020-01-01 until:2020-01-06
INFO: Querying virus since:2020-01-12 until:2020-01-17
INFO: Querying virus since:2020-01-23 until:2020-01-29
INFO: Querying virus since:2020-01-17 until:2020-01-23
INFO: Querying virus since:2020-02-03 until:2020-02-09
INFO: Querying virus since:2020-01-29 until:2020-02-03
INFO: Querying virus since:2020-02-15 until:2020-02-20
INFO: Querying virus since:2020-02-20 until:2020-02-26
INFO: Querying virus since:2020-02-09 until:2020-02-15
INFO: Querying virus since:2020-02-26 until:2020-03-03
INFO: Querying virus since:2020-03-03 until:2020-03-08
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-01-06%20unt
il%3A2020-01-12&l=english
INFO: Querying virus since:2020-03-08 until:2020-03-14
INFO: Querying virus since:2020-03-14 until:2020-03-20
INFO: Querying virus since:2020-03-20 until:2020-03-25
INFO: Querying virus since:2020-04-06 until:2020-04-11
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-01-12%20unt
il%3A2020-01-17&l=english
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-01-17%20unt
il%3A2020-01-23&l=english
INFO: Querying virus since:2020-03-25 until:2020-03-31
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-01-23%20unt
il%3A2020-01-29&l=english
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-01-01%20unt
il%3A2020-01-06&l=english
INFO: Querying virus since:2020-03-31 until:2020-04-06
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-01-29%20unt
il%3A2020-02-03&l=english
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-02-03%20unt
il%3A2020-02-09&l=english
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-02-20%20unt
```

```
il%3A2020-02-26&l=english
INFO: Querying virus since:2020-04-17 until:2020-04-23
INFO: Querying virus since:2020-04-11 until:2020-04-17
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-02-15%20unt
il%3A2020-02-20&l=english
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-02-09%20unt
il%3A2020-02-15&l=english
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-02-26%20unt
il%3A2020-03-03&l=english
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-03-14%20unt
il%3A2020-03-20&l=english
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-03-20%20unt
il%3A2020-03-25&l=english
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-03-03%20unt
il%3A2020-03-08&l=english
INFO: Using proxy 109.111.138.239:53281
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-03-08%20unt
il%3A2020-03-14&l=english
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-03-25%20unt
il%3A2020-03-31&l=english
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-04-06%20unt
il%3A2020-04-11&l=english
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-03-31%20unt
il%3A2020-04-06&l=english
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-04-17%20unt
il%3A2020-04-23&l=english
INFO: Scraping tweets from https://twitter.com/search?f=tweets&vertical=default&q=virus%20since%3A2020-04-11%20unt
il%3A2020-04-17&l=english
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Using proxy 109.111.138.239:53281
INFO: Got 20 tweets for virus%20since%3A2020-03-08%20until%3A2020-03-14.
INFO: Got 20 tweets (20 new).
INFO: Got 19 tweets for virus%20since%3A2020-01-23%20until%3A2020-01-29.
INFO: Got 39 tweets (19 new).
INFO: Got 16 tweets for virus%20since%3A2020-03-25%20until%3A2020-03-31.
INFO: Got 20 tweets for virus%20since%3A2020-02-20%20until%3A2020-02-26.
INFO: Got 55 tweets (16 new).
INFO: Got 75 tweets (20 new).
INFO: Got 18 tweets for virus%20since%3A2020-03-14%20until%3A2020-03-20.
INFO: Got 93 tweets (18 new).
INFO: Got 20 tweets for virus%20since%3A2020-04-06%20until%3A2020-04-11.
INFO: Got 113 tweets (20 new).
INFO: Got 20 tweets for virus%20since%3A2020-04-17%20until%3A2020-04-23.
INFO: Got 133 tweets (20 new).
INFO: Got 18 tweets for virus%20since%3A2020-02-09%20until%3A2020-02-15.
INFO: Got 19 tweets for virus%20since%3A2020-01-29%20until%3A2020-02-03.
INFO: Got 152 tweets (19 new).
INFO: Got 170 tweets (18 new).
INFO: Got 17 tweets for virus%20since%3A2020-01-17%20until%3A2020-01-23.
INFO: Got 187 tweets (17 new).
INFO: Got 19 tweets for virus%20since%3A2020-03-03%20until%3A2020-03-08.
INFO: Got 206 tweets (19 new).
INFO: Got 19 tweets for virus%20since%3A2020-02-26%20until%3A2020-03-03.
INFO: Got 225 tweets (19 new).
INFO: Got 20 tweets for virus%20since%3A2020-02-03%20until%3A2020-02-09.
INFO: Got 20 tweets for virus%20since%3A2020-01-06%20until%3A2020-01-12.
INFO: Got 245 tweets (20 new).
INFO: Got 265 tweets (20 new).
INFO: Got 18 tweets for virus%20since%3A2020-01-01%20until%3A2020-01-06.
INFO: Got 283 tweets (18 new).
INFO: Got 20 tweets for virus%20since%3A2020-02-15%20until%3A2020-02-20.
INFO: Got 303 tweets (20 new).
INFO: Got 19 tweets for virus%20since%3A2020-04-11%20until%3A2020-04-17.
INFO: Got 322 tweets (19 new).
INFO: Got 20 tweets for virus%20since%3A2020-01-12%20until%3A2020-01-17.
INFO: Got 342 tweets (20 new).
INFO: Got 20 tweets for virus%20since%3A2020-03-31%20until%3A2020-04-06.
INFO: Got 20 tweets for virus%20since%3A2020-03-20%20until%3A2020-03-25.
INFO: Got 362 tweets (20 new).
```

```
INFO: Got 382 tweets (20 new).
```

In [4]:
```
new_df = df
```

In [5]:
```
new_df = new_df[['username', 'tweet_id','text']]
```

In [6]:
```
new_df.head()
```

Out[6]:

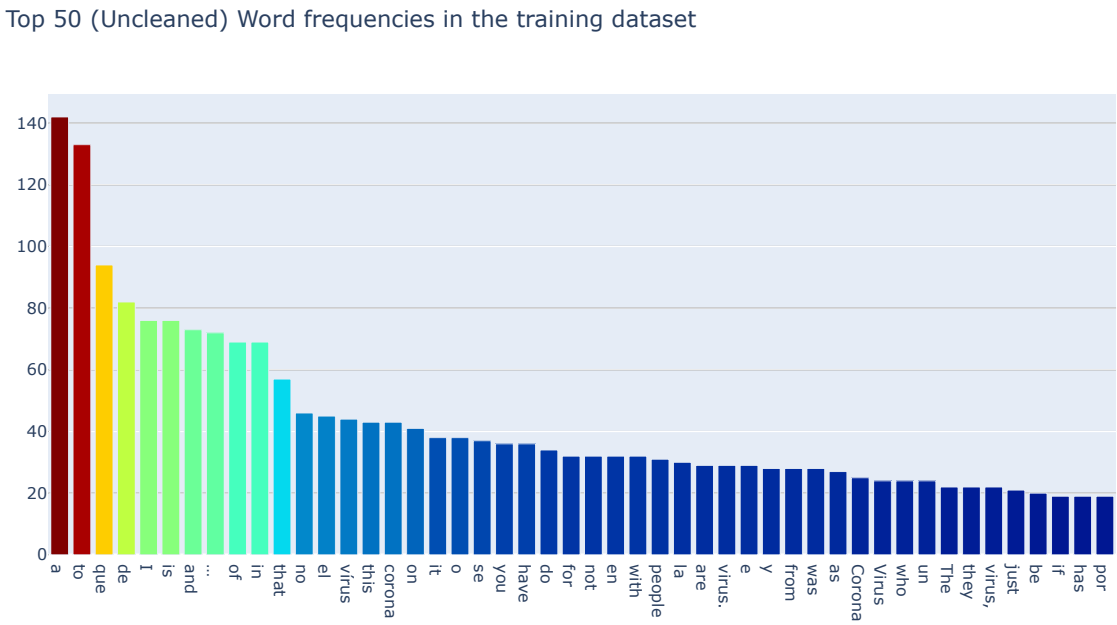
	username	tweet_id	text
0	🦠	1238615825923244032	corona virus cancelled my school yuppp
1	queiroz	1238615825348771842	É óbvio que temos que nos proteger evitar cont...
2	Julio César 🇲🇲	1238615825193373696	Ya estoy bastante cansado del virus. Cuidémono...
3	Martha Raffae	1238615824983830531	Any reduction of people you cannot verify as h...
4	Queen Kei 💋	1238615824887435266	I'm tired of hearing about the rona virus

In [7]:
```
import nltk
words = set(nltk.corpus.words.words())
new_df['text'].map(lambda sent
                   :" ".join(w for w in nltk.wordpunct_tokenize(sent) if w.lower() in words or not w.isalpha()))
```

Out[7]:
```
0                                corona virus my school
1                 mas a do corona se no de . A a . as .
2                               Ya virus . y , si a . .
3     Any reduction of people you cannot verify as n...
4                I ’ m tired of hearing about the virus
5     @ :// . be / NM63A7pYmXU real corona virus tru...
6                                            corona ta
7     someone explain to me how people are still Tak...
8     @ ! One hour to go ! on 13th ! Even the virus ...
9     O de no terminal do de Maria - . :// twitter ....
10                         1 ° de corona no de : pa la
11    China made this virus to kill their own elderl...
12    @ la corona virus :// . . / watch ? v = zG7KSh...
13    Wali Solo Virus Corona di # : :// regional . ....
14                    " ..... CORONA VIRUS " B , , 2020
15                  A si toman y las antes q el virus a
16    Too bad very sick cannot actually be tested fo...
17    @ : We don ' t need . We need our corrupt , , ...
18                                a do , e agora ta na
19          Y con dengue y corona virus , para mi moral
20    Mas de um ?(...) de pneumonia e em e .(...) , ...
21    do . da Umbrella Corp . Resident Evil sabe de ...
22    They are erecting two emergency supposedly in ...
23    They are going to admit her . She ' s on oxyge...
24    I ’ already got the virus so might as well jus...
25                       Se virus yo wi .. Sa pa non .
26                          eu , , do a . O as , e ...
27    @ I ' m a fellow # Resister . I found this , t...
28                                              virus
29               # : de do em , e :// bit . ly / 37ELfdC
                                ...
352   CA ME SOUL DE j ’ en ai de ce virus de ce conf...
353   But thank you for the here and on twitch ^^ I ...
354                                            Eu um .
355   En de 3 , solo hay de para el pueblo , de meta...
356   I just got off the horn with @ god and he said...
357                         O corona me a de 20 eu n e
358   But still 500 + it ’ s very sad . God please s...
359                             El 9 de a general virus
360           A . Um monte de , , e a galera o mortal .
361   My thought is there is no cure for Lupus which...
362                                presidente o corona
363                                  de e agora o ....
364   “ And at the end of the day , love , and it wi...
365   If you don ’ t update don ’ t you risk getting...
366    global o grave e as em . o , de , e . O nome : .
367   , o q some do :// twitter . / / status / 12425...
368                      At least this ' t the # virus #
369               corona virus , MATE O PRESIDENTE AGORA
370   di acara joget2 virus undetected , dan yang pa...
371                             Pantas orang2 , virus #
372   " dura , el sol ; , No , no da . se el calor ,...
373                de pronto si al virus me , se y se de
374                          se 62 e 65 . E no e se o ?
375   I understand . I work at a warehouse , food di...
376   Virus si o no ? :// twitter . / / status / 124...
377          10 sin x el virus : Con mi y mi de , se un
378                                um presidente , . #
379                                  as a , a do , a e
380   Tonight Live in :// FCR247 . . We are flushing...
381                               para o dele em e se .
Name: text, Length: 382, dtype: object
```

In [8]:
```
import plotly.offline as py
import plotly.graph_objs as go
py.init_notebook_mode()
```

In [9]:
```
all_words = new_df['text'].str.split(expand=True).unstack().value_counts()
data = [go.Bar(
            x = all_words.index.values[2:50],
            y = all_words.values[2:50],
            marker= dict(colorscale='Jet',
                         color = all_words.values[2:100]
                        ),
            text='Word counts'
    )]

layout = go.Layout(
    title='Top 50 (Uncleaned) Word frequencies in the training dataset'
)

fig = go.Figure(data=data, layout=layout)

py.iplot(fig, filename='basic-bar')
```



In [10]:
```
from nltk.corpus import stopwords
import string
stop = stopwords.words('english')
```

In [11]:
```
def stuff(p):

    temp = p.split()
    for i in temp:
        if i not in stop and i.isalpha() and len(i)>3:
            return i
```

In [12]:
```
# new_df['text'].apply(lambda p: i for i in p if i not in stop and i.isalpha() and len(i) > 2)
new_df['cleanwords'] = new_df['text'].apply(stuff)
new_df['cleanwords']
```

```
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-vers
us-copy
```

Out[12]:
```
0           corona
1            óbvio
2            estoy
3        reduction
4            tired
5             real
6           corona
7             okay
8             hour
9            vírus
10            caso
11           China
12          cumbia
13            Wali
14          CORONA
15           toman
16            sick
17            need
18          Passou
19           salto
20         falamos
21        Logotipo
22            They
23            They
24         already
25           virus
26            Quem
27          fellow
28           corno
29     notificação
          ...
352           SOUL
353          thank
354        tolerar
355      Venezuela
356           Guys
357         corona
358          still
359       comencen
360          gente
361        thought
362           Esse
363      bolsonaro
364           love
365         update
366     Comunidade
367           isso
368          least
369         corona
370       Kebayang
371         Pantas
372      calcinada
373         pronto
374           deve
375     understand
376          Virus
377          salir
378     presidente
379        escolas
380        Tonight
381        deveria
Name: cleanwords, Length: 382, dtype: object
```

In [13]:
```
# Storing the first text element as a string
first_text = new_df.cleanwords
print(first_text)
print("="*90)
#print(first_text.split(" "))
```

```
0           corona
1            óbvio
2            estoy
3        reduction
4            tired
5             real
6           corona
7             okay
8             hour
9            vírus
10            caso
11           China
12          cumbia
13            Wali
14          CORONA
15           toman
16            sick
17            need
18          Passou
19           salto
20         falamos
21        Logotipo
22            They
23            They
24         already
25           virus
26            Quem
27          fellow
28           corno
29     notificação
           ...
352           SOUL
353          thank
354        tolerar
355      Venezuela
356           Guys
357         corona
358          still
359       comencen
360          gente
361        thought
362           Esse
363      bolsonaro
364           love
365         update
366     Comunidade
367           isso
368          least
369         corona
370       Kebayang
371         Pantas
372      calcinada
373         pronto
374           deve
375     understand
376          Virus
377          salir
378     presidente
379        escolas
380        Tonight
381        deveria
Name: cleanwords, Length: 382, dtype: object
==========================================================================================
```

In [14]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/radhika/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[14]: True

In [15]:
```
stopwords = nltk.corpus.stopwords.words('english')
len(stopwords)
```

Out[15]: 179

In [16]:
```
stemmer = nltk.stem.PorterStemmer()
```

In [17]:
```
print("The stemmed form of running is: {}".format(stemmer.stem("running")))
print("The stemmed form of runs is: {}".format(stemmer.stem("runs")))
print("The stemmed form of run is: {}".format(stemmer.stem("run")))
```

```
The stemmed form of running is: run
The stemmed form of runs is: run
The stemmed form of run is: run
```

In [18]:
```
print("The stemmed form of leaves is: {}".format(stemmer.stem("leaves")))
```

```
The stemmed form of leaves is: leav
```

In [19]:
```
from nltk.stem import WordNetLemmatizer
lemm = WordNetLemmatizer()
print("The lemmatized form of leaves is: {}".format(lemm.lemmatize("leaves")))
```

```
The lemmatized form of leaves is: leaf
```

In [20]:
```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /Users/radhika/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[20]:
```
True
```

In [21]:
```
def print_top_words(model, feature_names, n_top_words):
    for index, topic in enumerate(model.components_):
        message = "\nTopic #{}:".format(index)
        message += " ".join([feature_names[i] for i in topic.argsort()[:-n_top_words - 1 :-1]])
        print(message)
        print("="*70)
```

In [22]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

In [23]:
```
lemm = WordNetLemmatizer()
class LemmaCountVectorizer(CountVectorizer):
    def build_analyzer(self):
        analyzer = super(LemmaCountVectorizer, self).build_analyzer()
        return lambda doc: (lemm.lemmatize(w) for w in analyzer(doc))
```

In [24]:
```
new_df['cleanwords'].loc[new_df['cleanwords'].map(lambda p: type(p) is float)]
```

Out[24]:
```
Series([], Name: cleanwords, dtype: object)
```

In [25]:
```
new_df.cleanwords = new_df.cleanwords.loc[new_df.cleanwords.map(lambda p:p is not None)]
```

```
//anaconda3/lib/python3.7/site-packages/pandas/core/generic.py:5096: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-vers
us-copy
```

In [26]:
```
cleanwordss = new_df.cleanwords.loc[new_df.cleanwords.map(lambda p:type(p) != float)]
```

In [27]:
```
cleanwordss.map(lambda p: type(p)).value_counts()
```

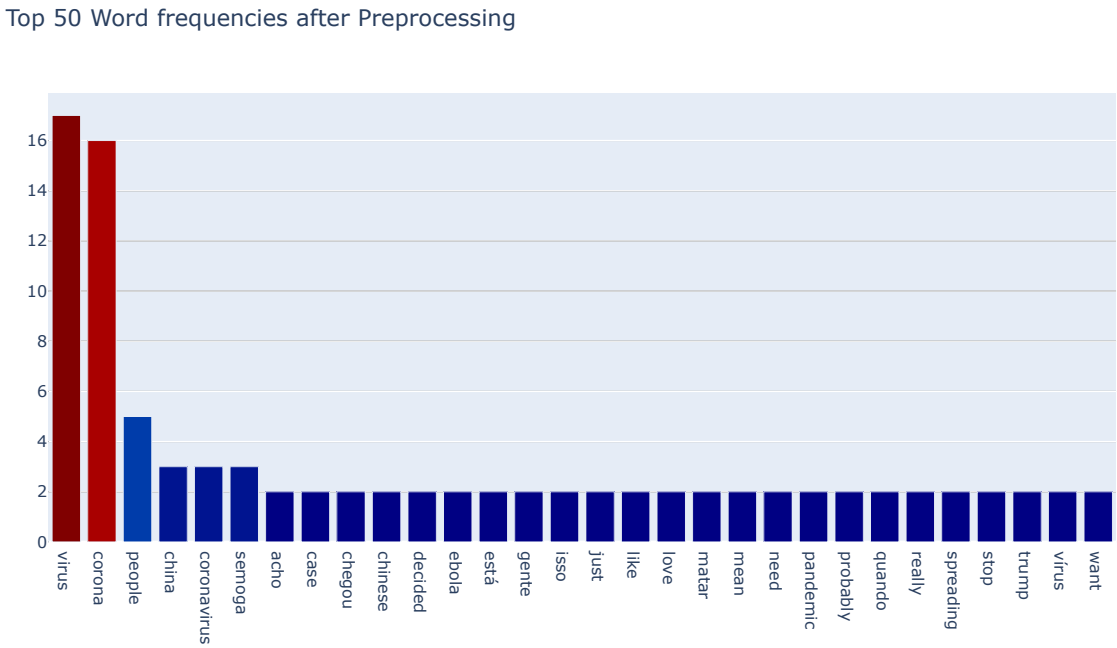
Out[27]:
```
<class 'str'>    377
Name: cleanwords, dtype: int64
```

In [28]:
```
new_df['cleanwords'] = new_df['cleanwords'].dropna()
```

```
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-vers
us-copy
```

In [29]:
```
text = list(cleanwordss)
# Calling our overwritten Count vectorizer
tf_vectorizer = LemmaCountVectorizer(max_df=0.95,
                                     min_df=2,
                                     stop_words='english',
                                     decode_error='ignore')
tf = tf_vectorizer.fit_transform(text)
```

In [30]:
```
import numpy as np
feature_names = tf_vectorizer.get_feature_names()
count_vec = np.asarray(tf.sum(axis=0)).ravel()
zipped = list(zip(feature_names, count_vec))
x, y = (list(x) for x in zip(*sorted(zipped, key=lambda x: x[1], reverse=True)))
# Now I want to extract out on the top 15 and bottom 15 words
Y = np.concatenate([y[0:15], y[-16:-1]])
X = np.concatenate([x[0:15], x[-16:-1]])

# Plotting the Plot.ly plot for the Top 50 word frequencies
data = [go.Bar(
            x = x[0:50],
            y = y[0:50],
            marker= dict(colorscale='Jet',
                         color = y[0:50]
                        ),
            text='Word counts')]

layout = go.Layout(
    title='Top 50 Word frequencies after Preprocessing'
)

fig = go.Figure(data=data, layout=layout)

py.iplot(fig, filename='basic-bar')
```



In [31]:
```
from sklearn.decomposition import LatentDirichletAllocation
lda = LatentDirichletAllocation(n_components=11, max_iter=5,
                                learning_method = 'online',
                                learning_offset = 50.,
                                random_state = 0)
```

In [32]:
```
lda.fit(tf)
```

Out[32]:
```
LatentDirichletAllocation(batch_size=128, doc_topic_prior=None,
                          evaluate_every=-1, learning_decay=0.7,
                          learning_method='online', learning_offset=50.0,
                          max_doc_update_iter=100, max_iter=5,
                          mean_change_tol=0.001, n_components=11, n_jobs=None,
                          perp_tol=0.1, random_state=0, topic_word_prior=None,
                          total_samples=1000000.0, verbose=0)
```

In [33]:
```
n_top_words = 40
print("\nTopics in LDA model: ")
tf_feature_names = tf_vectorizer.get_feature_names()
print_top_words(lda, tf_feature_names, n_top_words)
```

```
Topics in LDA model:

Topic #0:love chegou need case acho mean gente probably quando coronavirus like ebola corona want está decided chi
nese isso virus stop spreading pandemic vírus china really semoga matar trump people just
=====================================================================

Topic #1:pandemic matar stop virus decided quando ebola semoga mean case gente really está spreading like acho peo
ple just china probably vírus chegou isso want chinese love trump need coronavirus corona
=====================================================================

Topic #2:corona acho china coronavirus want isso semoga está quando case matar people chinese like mean ebola pand
emic really spreading chegou gente virus vírus love stop just probably need decided trump
=====================================================================

Topic #3:probably spreading just corona matar quando chinese case decided stop pandemic vírus need trump want peop
le semoga gente really está virus ebola chegou coronavirus love acho mean isso china like
=====================================================================

Topic #4:decided really chinese corona probably quando acho matar stop spreading need virus mean like case está eb
ola want pandemic people china coronavirus isso just semoga vírus chegou love gente trump
=====================================================================

Topic #5:isso like está mean corona chinese really want stop trump people coronavirus quando case pandemic china d
ecided vírus virus chegou probably matar love semoga gente just need spreading acho ebola
=====================================================================

Topic #6:virus china quando love está vírus gente like want chinese case chegou people decided probably acho pande
mic matar trump stop spreading mean ebola really corona coronavirus isso need semoga just
=====================================================================

Topic #7:trump vírus mean want just stop virus need chinese really coronavirus está quando like case semoga gente
acho corona chegou love pandemic people china spreading decided matar isso ebola probably
=====================================================================

Topic #8:semoga coronavirus vírus acho just ebola case mean need decided corona stop china está matar like chegou
trump gente pandemic really probably chinese virus love want quando isso people spreading
=====================================================================

Topic #9:people gente chinese ebola case want semoga está love pandemic chegou acho corona really trump decided qu
ando coronavirus isso like matar probably virus stop just need mean spreading vírus china
=====================================================================

Topic #10:like acho ebola gente want semoga quando chinese people mean case really probably china chegou isso coro
navirus decided está virus just pandemic matar spreading stop vírus corona love need trump
=====================================================================
```

In [34]:
```
first_topic = lda.components_[0]
second_topic = lda.components_[1]
third_topic = lda.components_[2]
fourth_topic = lda.components_[3]
```

In [35]:
```
first_topic.shape
```

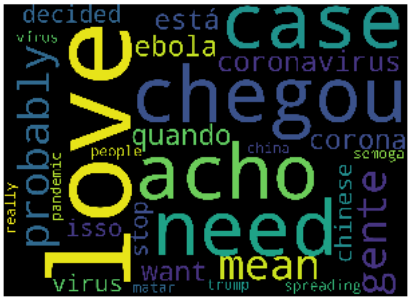
Out[35]:
```
(30,)
```

In [36]:
```
first_topic_words = [tf_feature_names[i] for i in first_topic.argsort()[:-50 - 1 :-1]]
second_topic_words = [tf_feature_names[i] for i in second_topic.argsort()[:-50 - 1 :-1]]
third_topic_words = [tf_feature_names[i] for i in third_topic.argsort()[:-50 - 1 :-1]]
fourth_topic_words = [tf_feature_names[i] for i in fourth_topic.argsort()[:-50 - 1 :-1]]
```

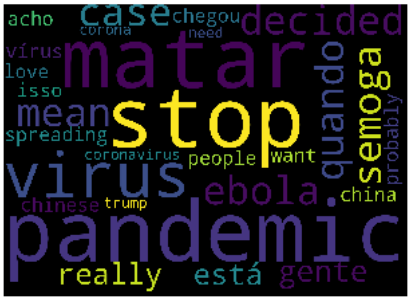
In [37]:
```
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in /anaconda3/lib/python3.7/site-packages (1.6.0)
Requirement already satisfied: numpy>=1.6.1 in /anaconda3/lib/python3.7/site-packages (from wordcloud) (1.18.2)
Requirement already satisfied: matplotlib in /anaconda3/lib/python3.7/site-packages (from wordcloud) (3.1.0)
Requirement already satisfied: pillow in /anaconda3/lib/python3.7/site-packages (from wordcloud) (6.1.0)
Requirement already satisfied: cycler>=0.10 in /anaconda3/lib/python3.7/site-packages (from matplotlib->wordcloud)
 (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /anaconda3/lib/python3.7/site-packages (from matplotlib->wordc
loud) (1.1.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /anaconda3/lib/python3.7/site-packages
(from matplotlib->wordcloud) (2.4.0)
Requirement already satisfied: python-dateutil>=2.1 in /anaconda3/lib/python3.7/site-packages (from matplotlib->wo
rdcloud) (2.8.0)
Requirement already satisfied: six in /anaconda3/lib/python3.7/site-packages (from cycler>=0.10->matplotlib->wordc
loud) (1.12.0)
Requirement already satisfied: setuptools in /anaconda3/lib/python3.7/site-packages (from kiwisolver>=1.0.1->matpl
otlib->wordcloud) (41.0.1)
```

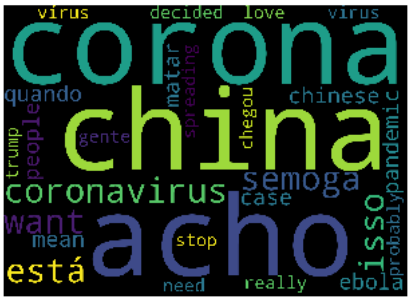
In [40]:
```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt

firstcloud = WordCloud(
                           stopwords=STOPWORDS,
                           background_color='black',
                           width=2500,
                           height=1800
                          ).generate(" ".join(first_topic_words))
plt.imshow(firstcloud)
plt.axis('off')
plt.show()
```



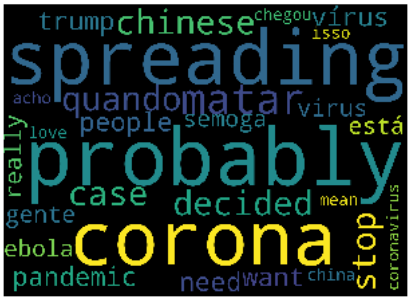
In [39]:
```
cloud = WordCloud(
                           stopwords=STOPWORDS,
                           background_color='black',
                           width=2500,
                           height=1800
                          ).generate(" ".join(second_topic_words))
plt.imshow(cloud)
plt.axis('off')
plt.show()
```



In [41]:
```
cloud = WordCloud(
                           stopwords=STOPWORDS,
                           background_color='black',
                           width=2500,
                           height=1800
                          ).generate(" ".join(third_topic_words))
plt.imshow(cloud)
plt.axis('off')
plt.show()
```



In [42]:
```
cloud = WordCloud(
                          stopwords=STOPWORDS,
                          background_color='black',
                          width=2500,
                          height=1800
                         ).generate(" ".join(fourth_topic_words))
plt.imshow(cloud)
plt.axis('off')
plt.show()
```



Let's take a look at the bigram

In [43]:
```
bigrams = nltk.bigrams(cleanwordss)
```

In [44]:
```
from collections import Counter

counter = Counter(bigrams)
print(counter.most_common(10))
```

```
[(('Corona', 'virus'), 2), (('corona', 'óbvio'), 1), (('óbvio', 'estoy'), 1), (('estoy', 'reduction'), 1), (('redu
ction', 'tired'), 1), (('tired', 'real'), 1), (('real', 'corona'), 1), (('corona', 'okay'), 1), (('okay', 'hour'),
 1), (('hour', 'vírus'), 1)]
```

In [ ]: