Twitter Stream

Analysis of Social Media Contents

Alessandro Ortis - University of Catania

To get your own consumer key, consumer secret, access token and access secret, create a Twitter application: https://apps.twitter.com/app/new (https://apps.twitter.com/app/new).

http://docs.tweepy.org/en/latest/streaming_how_to.html (http://docs.tweepy.org/en/latest/streaming_how_to.html)

1. One way of doing it:

In [1]:
```
import configparser
import wget
import os
import matplotlib.pyplot as plt

config = configparser.ConfigParser()
config.read('TwitterConfig.ini')

auth_info = config['AO_Tutorial']


CONSUMER_KEY = auth_info['consumer_key']
CONSUMER_SECRET = auth_info['consumer_secret']
ACCESS_TOKEN = auth_info['access_token']
ACCESS_TOKEN_SECRET = auth_info['access_token_secret']
#CONSUMER_KEY = 'INSERT_YOUR_OWN_HERE'
#CONSUMER_SECRET = 'INSERT_YOUR_OWN_HERE'
#ACCESS_TOKEN = 'INSERT_YOUR_OWN_HERE'
#ACCESS_TOKEN_SECRET = 'INSERT_YOUR_OWN_HERE'
```

In [4]:
```
from tweepy import (Stream, OAuthHandler) # OAuth is an open standard for access delegation, commonly used as a way for Internet
from tweepy.streaming import StreamListener
import time # For time.sleep()



class Listener(StreamListener):
    crawled_tweets = 0

    #Called when a new status arrives
    def on_status(self, status):
        #on_status() focuses on status updates.
        # do not override on_data()
        try:

            save_file = open('twitterDB.txt', 'a', encoding='utf8') # a = append
            text = status.text.replace('\n', ' ')

            save_file.write(str(time.time()) + ':: ' + text)
            save_file.write('\n')
            save_file.close()
            print(text)
            print('\n')

            media_files = []
            media = status.entities.get('media', [])
            if(len(media) > 0):
                print("media...")
                #media_files.append(media[0]['media_url'])
                media_url = media[0]['media_url']
                img_name = media[0]['media_url'].split('/')[-1]
                print(img_name)
                print("media download...")
                wget.download(media_url)
                #f = urllib3.urlopen(media_url)
                print('\n')
                a = plt.imread(img_name)
                plt.imshow(a)
                plt.show()
               # plt.imshow(img_name, dtype='float')
                #plt.show()
                print('\n')


            self.crawled_tweets +=1
            if self.crawled_tweets == 3:
                print("Disconnecting the stream listener...")
                return False
            else:
                return True

        except BaseException as err: # BaseException is the base class for all built-in exceptions. Problems that could happen ar
            print('Failed on_status, ', str(err))
            time.sleep(5)

    def on_disconnect(self, notice):
        print("Twitter sent a disconnection notice: "+notice)

    def on_error(self, status):
        print("Status: ",str(status))
        # if you get 420 as status, your app has been rate limited due too many queries
        if status == 420:
            #returning False in on_error disconnects the stream
            print("Disconnecting the stream...")
            return False
        """
        else:
            time_to_sleep = 15 * 60  + 10
            print("Sleeping for ",str(time_to_sleep), " secs...")
            time.sleep(time_to_sleep)
            print("New attempt..")
            return True
        """

auth  = OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET) # Authorizing ourselves
auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
# Create a Stream passing a Listener
twitter_stream = Stream(auth, Listener())
# Filtering tweets.
# Possible params: locations, languages, follow (people).
#The default argument for all of these is None. NB very few accounts have geolocations.
#twitter_stream.filter(track=['car'])


###
# Examples for locations (i.e., top-left and bottom-right corners of a regtangle):
```

```
# San Francisco = [-122.75,  36.8,   -121.75  ,  37.8]
# New York = [-74,40,-73,41]
# San Francisco OR New York = [-122.75,36.8,-121.75,37.8,-74,40,-73,41]
# For further details see https://developer.twitter.com/en/docs/tweets/filter-realtime/guides/basic-stream-parameters

#twitter_stream.filter(locations=[-74,40,-73,41]) # New York
#twitter_stream.filter(locations=[12.391291591, 41.8063758892, 12.5884961892, 41.9699521708]) # Rome

# A.Ortis: 317046418   BBC World: 742143
#twitter_stream.filter(follow=["317046418","742143"], is_async= True)
twitter_stream.filter(follow=["317046418"], is_async= True)
#twitter_stream.filter(track=['#covid-19'], is_async= True)
```

```
Status:  420
Disconnecting the stream...
SMA ! https://t.co/I14iNC9NnO (https://t.co/I14iNC9NnO)


media...
Fo8ECBbXsAAj3zF.jpg
media download...
Failed on_status,  HTTP Error 404: Not Found
https://t.co/YO5xHzfeuF (https://t.co/YO5xHzfeuF)


media...
Fo8EIpIXwAA6Goy.jpg
media download...
```

2. Another way of doing it:

In [4]:
```
from tweepy import (Stream, OAuthHandler)
from tweepy.streaming import StreamListener
import os

class Listener(StreamListener):

    tweet_counter = 0 # Static variable

    def login(self):
        auth = OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
        auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
        return auth

    #Called when a new status arrives
    def on_status(self, status):
        Listener.tweet_counter += 1
        print(str(Listener.tweet_counter) + '. Screen name = "%s" Tweet = "%s"'
              %(status.author.screen_name, status.text.replace('\n', ' ')))
        print('\n')
        # stop_at is a static variable
        if Listener.tweet_counter < Listener.stop_at:
            return True
        else:
            print('Max num reached = ' + str(Listener.tweet_counter))
            return False

    def getTweetsByGPS(self, stop_at_number, latitude_start, longitude_start, latitude_finish, longitude_finish):
        try:
            # Handling the number of tweets
            Listener.stop_at = stop_at_number # Create static variable
            # Handling the login (API auth)
            auth = self.login()
            # Creating a customized Stream on-the-fly
            streaming_api = Stream(auth, Listener(), timeout=100) # Socket timeout value (default: 300)
            streaming_api.filter(follow=None, locations=[latitude_start, longitude_start, latitude_finish, longitude_finish])
        except KeyboardInterrupt:
            print('Got keyboard interrupt')

    def getTweetsByHashtag(self, stop_at_number, hashtag):
        try:
            Listener.stop_at = stop_at_number
            auth = self.login()
            # Creating a customized Stream on-the-fly
            streaming_api = Stream(auth, Listener(), timeout=100)
            streaming_api.filter(track=[hashtag], is_async= True)
        except KeyboardInterrupt:
            print('Got keyboard interrupt')

    def on_error(self, status):
        print("Status: ",str(status))
        return False


# Creating a Listener (with a proper Stream instanced in)
listener = Listener()
#listener.getTweetsByHashtag(20,"Draghi")
#listener.getTweetsByHashtag(20,"Covid-19")
listener.getTweetsByHashtag(20,"Meloni")
#listener.getTweetsByHashtag(20,"Camilleri")

#listener.getTweetsByGPS(20, -84.395198, 33.746876, -84.385585, 33.841601) # Atlanta area. Tool to find coordinates for any place
#listener.getTweetsByGPS(20, 4.4024141985,35.2844271234,19.779923473,47.8221174488) # Italy
#listener.getTweetsByGPS(20, 12.391291591, 41.8063758892, 12.5884961892, 41.9699521708) #Rome
```

1. Screen name = "annabelle69250" Tweet = "RT @__Verlaine__: À Bruxelles, les yeux dans les yeux Giorgia Meloni a rappelé à Ola f Scholz que l'Italie n'avait de leçons à recevoir de p…"

2. Screen name = "nunziachinnici" Tweet = "RT @SecolodItalia1: La stilista Elisabetta Franchi accusa: “A Sanremo elogi a tante donne, nessuno che abbia citato la Meloni…” https://t.c…" (https://t.c…")

3. Screen name = "ATigreweyti" Tweet = "RT @MargMar80017297: The ET|n PM is now embarking on a vast diplomatic and economic reco nquest operation. Before arriving in Paris, Abiy we…"

4. Screen name = "AndreaBoitani" Tweet = "Andrea Boitani interpreta e discute, da differenti punti di vista, la recenteaffermaz ione di Giorgia Meloni, second… https://t.co/jtSFdFpjih" (https://t.co/jtSFdFpjih")

5. Screen name = "cinzia_pietro" Tweet = "Letta e’ felice e orgoglioso per il risultato…."

6. Screen name = "gallese_giorgio" Tweet = "RT @elio_vito: Ma c’è qualcuno, un giornalista, un commentatore che dica come la de stra, il governo Meloni, abbia perso, in termini assolut…"

7. Screen name = "BennaGianni" Tweet = "RT @emifittipaldi: Meloni, se uno scrittore la critica o un giornalista fa un’inchiesta che non le garba, li manda subito a processo. Se d…"

8. Screen name = "AdolpheBenitoG" Tweet = "RT @__Verlaine__: À Bruxelles, les yeux dans les yeux Giorgia Meloni a rappelé à Ola f Scholz que l'Italie n'avait de leçons à recevoir de p…"

9. Screen name = "Helu_Tdf21" Tweet = "RT @knkoiii: The ET|n PM is now embarking on a vast diplomatic and economic reconquest op eration. Before arriving in Paris, Abiy went to Ro…"

10. Screen name = "PerrierLincoln" Tweet = "RT @__Verlaine__: À Bruxelles, les yeux dans les yeux Giorgia Meloni a rappelé à Ol af Scholz que l'Italie n'avait de leçons à recevoir de p…"

11. Screen name = "mesiihiluf" Tweet = "RT @Abadit__hayelom: 🌸The ET|n PM is now embarking on a vast diplomatic and economic r econquest operation. Before arriving in Paris, Abiy w…"

12. Screen name = "canadafreepress" Tweet = "RT @canadafreepress: Italy’s Georgia Meloni Rips Woke Extremists to Shreds – Are Y ou Watching America? https://t.co/7zKkZ8BMIn" (https://t.co/7zKkZ8BMIn")

13. Screen name = "BennaGianni" Tweet = "RT @LucillaMasini: Nel 2017 il Governo Gentiloni si era costituito parte civile nel pr ocesso Ruby ter, ma la Meloni ha revocato l'atto. Dio…"

14. Screen name = "AlexanderLandSp" Tweet = "RT @Orraf5: La #RepublicaDelleBanane "Processo Ruby ter, Palazzo Chigi non sarà pi ù parte civile. La Meloni ha dato mandato di revocare la…"

15. Screen name = "bettyvegas3" Tweet = "RT @Liwi_00: The ET|n PM is now embarking on a vast diplomatic and economic reconquest operation. Before arriving in Paris, Abiy went to Ro…"

16. Screen name = "mesiihiluf" Tweet = "RT @haddis_simret: TheET|n PM is now embarking on a vast diplomatic &economic reconq uest operation.Before arriving in Paris,Abiy went to Ro…"

17. Screen name = "Stefano173456" Tweet = "RT @DavideR46325615: Caso Ruby Ter, la Premier Giorgia Meloni RITIRA la costituzione di parte civile contro Silvio Berlusconi. Un preparati…"

18. Screen name = "Antonio98573743" Tweet = "@Adriana04966084 @Stefano173456 @berlusconi @GiorgiaMeloni @matteosalvinimi @melon i Spazzatura https://t.co/NuojxQIPZ3" (https://t.co/NuojxQIPZ3")

19. Screen name = "ProcopioDiC" Tweet = "RT @elio_vito: Ma c’è qualcuno, un giornalista, un commentatore che dica come la destr a, il governo Meloni, abbia perso, in termini assolut…"

20. Screen name = "VITOsucalabrisi" Tweet = "RT @__Verlaine__: À Bruxelles, les yeux dans les yeux Giorgia Meloni a rappelé à O laf Scholz que l'Italie n'avait de leçons à recevoir de p…"

Max num reached = 20

3. Cleaning tweets:

In [5]:
```
import re
from string import punctuation
from nltk.corpus import stopwords
from nltk.tokenize import TweetTokenizer

punctuation += '´´’…“”——»«' # string.punctuation misses these.

cache_english_stopwords = stopwords.words('english') # Could speed up code by making this a set

def tweet_clean(tweet):
    print('Original tweet:', tweet, '\n')
    # Remove HTML special entities (e.g. &amp;)
    tweet_no_special_entities = re.sub(r'\&\w*;', '', tweet)
    print('No special entitites:', tweet_no_special_entities, '\n')
    # Remove tickers (Clickable stock market symbols that work like hashtags and start with dollar signs instead)
    tweet_no_tickers = re.sub(r'\$\w*', '', tweet_no_special_entities) # Substitute. $ needs to be escaped because it means somet
    print('No tickers:', tweet_no_tickers, '\n')
    # Remove hyperlinks
    tweet_no_hyperlinks = re.sub(r'https?:\/\/.*\/\w*', '', tweet_no_tickers)
    print('No hyperlinks:', tweet_no_hyperlinks, '\n')
    # Remove hashtags
    tweet_no_hashtags = re.sub(r'#\w*', '', tweet_no_hyperlinks)
    print('No hashtags:', tweet_no_hashtags, '\n')
    # Remove Punctuation and split 's, 't, 've with a space for filter
    tweet_no_punctuation = re.sub(r'[' + punctuation.replace('@', '') + ']+', ' ', tweet_no_hashtags)
    print('No punctuation:', tweet_no_punctuation, '\n')
    # Remove words with 2 or fewer letters (Also takes care of RT)
    tweet_no_small_words = re.sub(r'\b\w{1,2}\b', '', tweet_no_punctuation) # \b represents a word boundary
    print('No small words:', tweet_no_small_words, '\n')
    # Remove whitespace (including new line characters)
    tweet_no_whitespace = re.sub(r'\s\s+', ' ', tweet_no_small_words)
    tweet_no_whitespace = tweet_no_whitespace.lstrip(' ') # Remove single space left on the left
    print('No whitespace:', tweet_no_whitespace, '\n')
    # Remove characters beyond Basic Multilingual Plane (BMP) of Unicode:
    tweet_no_emojis = ''.join(c for c in tweet_no_whitespace if c <= '\uFFFF') # Apart from emojis (plane 1), this also removes h
    print('No emojis:', tweet_no_emojis, '\n')
    # Tokenize: Change to lowercase, reduce length and remove handles
    tknzr = TweetTokenizer(preserve_case=False, reduce_len=True, strip_handles=True) # reduce_len changes, for example, waaaaaayy
    tw_list = tknzr.tokenize(tweet_no_emojis)
    print('Tweet tokenize:', tw_list, '\n')
    # Remove stopwords
    list_no_stopwords = [i for i in tw_list if i not in cache_english_stopwords]
    print('No stop words:', list_no_stopwords, '\n')
    # Final filtered tweet
    tweet_filtered =' '.join(list_no_stopwords) # ''.join() would join without spaces between words.
    print('Final tweet:', tweet_filtered)

s = '    RT @Amila #Test\nTom\'s newly listed Co. &amp; Mary\'s unlisted     Group to supply tech for nlTK.\nh.. $TSLA $AAPL http
tweet_clean(s)
```

```
Original tweet:     RT @Amila #Test
Tom's newly listed Co. &amp; Mary's unlisted     Group to supply tech for nlTK.
h.. $TSLA $AAPL https:// (https://) t.co/x34afsfQsh

No special entitites:     RT @Amila #Test
Tom's newly listed Co.  Mary's unlisted     Group to supply tech for nlTK.
h.. $TSLA $AAPL https:// (https://) t.co/x34afsfQsh

No tickers:     RT @Amila #Test
Tom's newly listed Co.  Mary's unlisted     Group to supply tech for nlTK.
h..   https:// (https://) t.co/x34afsfQsh

No hyperlinks:     RT @Amila #Test
Tom's newly listed Co.  Mary's unlisted     Group to supply tech for nlTK.
h..

No hashtags:     RT @Amila
Tom's newly listed Co.  Mary's unlisted     Group to supply tech for nlTK.
h..
```

Exercise

Create your own Tweets listener based on some location (e.g., right here) or hashtag (e.g., #SocialMediaAnalysisUNICT), start listening while publishing some related content on Twitter. The objective is to check if your listener is able to catch your own tweets.

Exercise

Explore the Tweepy API on not yet seen methods (e.g., get specific user information). For example, extract the tweets published by some famous person.
https://tweepy.readthedocs.io/en/latest/api.html (https://tweepy.readthedocs.io/en/latest/api.html)

In [ ]: