

IST 652 – Scripting for Data Analysis
Homework Two: Semi-structured Data
James R. Ingram II
September 8, 2018

Contents

Hashtag Intel - A Comparison of Intelligence Agency Tweets	3
Data and its Source	3
Data Questions.....	3
Question 1: Which agency uses more hashtags in their tweets?	3
Question 2: What are the 20 most frequently used hashtags by each agency?	3
Program Descriptions.....	3
twitter_client.py.....	3
twitter_get_user_timeline.py.....	3
twitter_hashtag_stats.py	4
twitter_hashtag_frequency.py	4
Output File Descriptions	4
user_timeline_CIA.jsonl	4
user_timeline_NSAGov.jsonl	5
File Output	6
twitter_hashtag_stats.py user_timeline_CIA.jsonl.....	6
twitter_hashtag_stats.py user_timeline_NSAGov.jsonl	6
twitter_hashtag_frequency.py user_timeline_CIA.jsonl	7
twitter_hashtag_frequency.py user_timeline_NSAGov.jsonl	7
Conclusion.....	7
Appendix: Python Code	9
Appendix A: twitter_client.py	9
Appendix B: twitter_get_user_timeline.py.....	10
Appendix C: twitter_hashtag_stats.py.....	11
Appendix D: twitter_hashtag_frequency.py.....	13

Hashtag Intel - A Comparison of Intelligence Agency Tweets

Data and its Source

For this project, 3200 twitter feed posts, known as “tweets”, each from the Central Intelligence Agency (CIA) and the National Security Agency (NSA) will be written to files in JSON (JavaScript Object Notation) format. Next, the “hashtag” data will be extracted for analysis and comparison. A hashtag is a word or combination of words preceded by the symbol “#” used to identify the topic of a post on social media. For both intelligence agencies’ tweets, hashtag frequency and usage will be investigated. The only data cleaning step performed was transforming hashtag text values to lowercase so that “#ThisIsAnExample” and “#thisisanexample” would both be counted as the same word during frequency analysis.

Data Questions

Question 1: Which agency uses more hashtags in their tweets?

To answer this question, summary statistics of hashtag usage will be computed for each intelligence agency. This analysis will contain counts of tweets with and without hashtags, as well as the number and percentage of hashtags used in each tweet. The results of the individual analyses will be compared with one another to determine which organization uses more hashtags and also which uses multiple hashtags more frequently.

Question 2: What are the 20 most frequently used hashtags by each agency?

Next, the twenty most frequently used hashtags for each organization will be identified. Although sentiment analysis won’t be performed, an examination of the most-used hashtag text can tell us what topics are important to each agency.

Program Descriptions

To accomplish these objectives, four Python programs have been written and executed.

twitter_client.py

Before beginning to collect data from Twitter, it was necessary to register an application with Twitter. Once registered, authentication information, including keys and access tokens, were received. The authentication details were stored in the operating system’s local environment and then called into variables within this Python program. In order to interact with the API, a Python client is required. The popular client Tweepy was chosen to make the API calls and establish authentication. The Python file “twitter_client.py” The code necessary to make the authentication calls was stored as a function that can be imported into other files, such as the “twitter_get_user_timeline.py” file described below.

twitter_get_user_timeline.py

This program is called from the command line with an argument, whose value is equal to the username of the Twitter user we want to get data from. For example, to access the timeline of the CIA, one would type the command “python twitter_get_user_timeline.py CIA” in the command line or terminal window. The program will then write the contents of the user’s tweets to a JSON Lines file with extension “.jsonl”. In a JSON Lines file, each line separated by a newline character “\n” represents a JSON document. The code uses Tweepy’s Cursor parameter to iterate through 16 pages, each containing 200 tweets, to return a maximum of 3,200 (the limit set by Twitter) of the user’s most recent tweets. Then for each entry (tweet) JSON’s built-in function json.dumps() will be used to write them to the specified

file. A tweet contains many attributes and its structure is predictable. For our analyses, hashtag data can be found within a tweet's "entities", which is a dictionary of items.

twitter_hashtag_stats.py

This program counts the total number and percentage of tweets which contain hashtags, as well as the frequency of the number of hashtags within each tweet. The file first opens the jsonl file specified by the argument following the program name. Next, it iteratively loads each line of the file (a single tweet) and assigns it to a "tweet" variable. It then calls a function to get the entities dictionary from the "tweet", and then the hashtags from the entities are put into a list. The number and frequency of hashtag values are then computed.

twitter_hashtag_frequency.py

The final file extracts data from a user's timeline and creates a list and count of the most common hashtags used. The filename from which to read the data is entered as an argument when opening the file, and the same function is used as in the previous program to extract the tweet data. The Counter function is imported from the collections module and is used to find the most common hashtag text values. These values are then displayed to the user.

Output File Descriptions

user_timeline_CIA.jsonl

This file contains the twitter data collected from the CIA's feed using the "twitter_get_user_timeline.py" program above. It is in JSON Lines format where each line is a JSON document (an individual tweet). The first line is displayed below:

```
{
  "created_at": "Fri Sep 07 16:58:24 +0000 2018",
  "id": 1038109240433549313,
  "id_str": "1038109240433549313",
  "text": "@SavannahGuthrie Threading is fun & easy! Put out a tweet, reply to yourself w/ 2nd tweet, & then reply to 2nd tweet\u0026 https://t.co/IGBHibF0en",
  "truncated": true,
  "entities": {
    "hashtags": [],
    "symbols": [],
    "user_mentions": [
      {
        "screen_name": "SavannahGuthrie",
        "name": "Savannah Guthrie",
        "id": 52070270,
        "id_str": "52070270",
        "indices": [0, 16]
      }
    ],
    "urls": [
      {
        "url": "https://t.co/IGBHibF0en",
        "expanded_url": "https://twitter.com/i/web/status/1038109240433549313",
        "display_url": "twitter.com/i/web/status/1\u0026",
        "indices": [125, 148]
      }
    ]
  },
  "source": "<a href='\"http://twitter.com/\"' rel='\"nofollow\"'>Twitter Web Client</a>",
  "in_reply_to_status_id": 1038070597291524097,
  "in_reply_to_status_id_str": "1038070597291524097",
  "in_reply_to_user_id": 52070270,
  "in_reply_to_user_id_str": "52070270",
  "in_reply_to_screen_name": "SavannahGuthrie",
  "user": {
    "id": 2359926157,
    "id_str": "2359926157",
    "name": "CIA",
    "screen_name": "CIA",
    "location": "Langley, VA",
    "description": "We are the Nation's first line of defense. We accomplish what others cannot accomplish and go where others cannot go.",
    "url": "http://t.co/3PkAkcBfw9",
    "entities": {
      "url": {
        "urls": [
          {
            "url": "http://t.co/3PkAkcBfw9",
            "expanded_url": "http://cia.gov",
            "display_url": "cia.gov",
            "indices": [0, 22]
          }
        ]
      }
    },
    "protected": false,
    "followers_count": 2371336,
    "friends_count": 31,
    "listed_count": 9202,
    "created_at": "Mon Feb 24 19:08:11 +0000 2014",
    "favourites_count": 106,
    "utc_offset": null,
    "time_zone": null,
    "geo_enabled": false,
    "verified": true,
    "statuses_count": 3961,
    "lang": "en",
    "contributors_enabled": false,
    "is_translator": false,
    "is_translation_enabled": false,
    "profile_background_color": "000000",
    "profile_background_image_url": "http://abs.twimg.com/images/themes/theme14/bg.gif",
    "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme14/bg.gif",
    "profile_background_tile": false,
    "profile_image_url": "http://pbs.twimg.com/profile_images/474250448757481472/QcQ_P01E_normal.jpeg",
    "profile_image_url_https": "https://pbs.twimg.com/profile_images/474250448757481472/QcQ_P01E_normal.jpeg",
    "profile_banner_url": "https://pbs.twimg.com/profile_banners/2359926157/1401488215",
  }
}
```

```
"profile_link_color": "1B95E0", "profile_sidebar_border_color": "000000",
"profile_sidebar_fill_color": "000000", "profile_text_color": "000000",
"profile_use_background_image": false, "has_extended_profile": true, "default_profile": false,
"default_profile_image": false, "following": true, "follow_request_sent": false,
"notifications": false, "translator_type": "none"}, "geo": null, "coordinates": null, "place":
null, "contributors": null, "is_quote_status": false, "retweet_count": 45, "favorite_count":
359, "favorited": false, "retweeted": false, "possibly_sensitive": false, "lang": "en"}
```

Each line is a Python dictionary, containing key value pairs corresponding to the tweet's metadata and data.

user_timeline_NSAGov.jsonl

This file type is the same as the previous example, but it contains all of the tweet data retrieved from the NSA's twitter feed. A sample line is illustrated below:

```
{"created_at": "Thu Sep 06 21:13:17 +0000 2018", "id": 1037810996268683264, "id_str":
"1037810996268683264", "text": "RT @BillingtonCyber: GEN Nakasone in has capacity as
Commander, US Cyber Command closes our #BillingtonSummit. He says we must face and ada\u2026",
"truncated": false, "entities": {"hashtags": [{"text": "BillingtonSummit", "indices": [92,
109]}], "symbols": [], "user_mentions": [{"screen_name": "BillingtonCyber", "name":
"Billington Cyber", "id": 1130167399, "id_str": "1130167399", "indices": [3, 19]}], "urls":
[]}, "source": "<a href=\"http://twitter.com\" rel=\"nofollow\">Twitter Web Client</a>",
"in_reply_to_status_id": null, "in_reply_to_status_id_str": null, "in_reply_to_user_id": null,
"in_reply_to_user_id_str": null, "in_reply_to_screen_name": null, "user": {"id": 2248872301,
"id_str": "2248872301", "name": "NSA/CSS", "screen_name": "NSAGov", "location": "Fort Meade,
MD", "description": "The official account for the National Security Agency/Central Security
Service, home to America's codemakers and codebreakers.\n\nTerms: http://t.co/4mMmeo2fF9",
"url": "http://t.co/d6ukVzNJvi", "entities": {"url": {"urls": [{"url":
"http://t.co/d6ukVzNJvi", "expanded_url": "http://www.nsa.gov", "display_url": "nsa.gov",
"indices": [0, 22]}]}}, "description": {"urls": [{"url": "http://t.co/4mMmeo2fF9",
"expanded_url": "http://1.usa.gov/1NVLm6s", "display_url": "1.usa.gov/1NVLm6s", "indices":
[135, 157]}]}}, "protected": false, "followers_count": 472042, "friends_count": 118,
"listed_count": 3900, "created_at": "Mon Dec 16 14:48:10 +0000 2013", "favourites_count": 7,
"utc_offset": null, "time_zone": null, "geo_enabled": false, "verified": true,
"statuses_count": 1962, "lang": "en", "contributors_enabled": false, "is_translator": false,
"is_translation_enabled": false, "profile_background_color": "022330",
"profile_background_image_url": "http://abs.twimg.com/images/themes/theme15/bg.png",
"profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme15/bg.png",
"profile_background_tile": true, "profile_image_url":
"http://pbs.twimg.com/profile_images/727527189525090304/OHr577N-_normal.jpg",
"profile_image_url_https": "https://pbs.twimg.com/profile_images/727527189525090304/OHr577N-
_normal.jpg", "profile_banner_url":
"https://pbs.twimg.com/profile_banners/2248872301/1462292333", "profile_link_color": "0084B4",
"profile_sidebar_border_color": "FFFFFF", "profile_sidebar_fill_color": "C0DFEC",
"profile_text_color": "333333", "profile_use_background_image": true, "has_extended_profile":
false, "default_profile": false, "default_profile_image": false, "following": true,
"follow_request_sent": false, "notifications": false, "translator_type": "none"}, "geo": null,
"coordinates": null, "place": null, "contributors": null, "retweeted_status": {"created_at":
"Thu Sep 06 20:48:13 +0000 2018", "id": 1037804688102580225, "id_str": "1037804688102580225",
"text": "GEN Nakasone in has capacity as Commander, US Cyber Command closes our
#BillingtonSummit. He says we must face and\u2026 https://t.co/HHSR2jdiGx", "truncated": true,
"entities": {"hashtags": [{"text": "BillingtonSummit", "indices": [71, 88]}], "symbols": [],
"user_mentions": [], "urls": [{"url": "https://t.co/HHSR2jdiGx", "expanded_url":
"https://twitter.com/i/web/status/1037804688102580225", "display_url":
"twitter.com/i/web/status/1\u2026", "indices": [116, 139]}]}}, "source": "<a
href=\"http://twitter.com/download/iphone\" rel=\"nofollow\">Twitter for iPhone</a>",
```

```

"in_reply_to_status_id": null, "in_reply_to_status_id_str": null, "in_reply_to_user_id": null,
"in_reply_to_user_id_str": null, "in_reply_to_screen_name": null, "user": {"id": 1130167399,
"id_str": "1130167399", "name": "Billington Cyber", "screen_name": "BillingtonCyber",
"location": "Washington, DC", "description": "Billington Cybersecurity produces world class
conferences featuring leaders from industry, government and the military. #Billingtonsummit
#Billingtoncyber", "url": "https://t.co/6tGF537gW7", "entities": {"url": {"urls": [{"url":
"https://t.co/6tGF537gW7", "expanded_url": "http://www.billingtoncybersecurity.com/",
"display_url": "billingtoncybersecurity.com", "indices": [0, 23]}}}, "description": {"urls":
[]}}, "protected": false, "followers_count": 950, "friends_count": 306, "listed_count": 26,
"created_at": "Tue Jan 29 05:46:26 +0000 2013", "favourites_count": 180, "utc_offset": null,
"time_zone": null, "geo_enabled": false, "verified": false, "statuses_count": 869, "lang":
"en", "contributors_enabled": false, "is_translator": false, "is_translation_enabled": false,
"profile_background_color": "000000", "profile_background_image_url":
"http://abs.twimg.com/images/themes/theme1/bg.png", "profile_background_image_url_https":
"https://abs.twimg.com/images/themes/theme1/bg.png", "profile_background_tile": false,
"profile_image_url":
"http://pbs.twimg.com/profile_images/904741376981684224/6FuIcOPs_normal.jpg",
"profile_image_url_https":
"https://pbs.twimg.com/profile_images/904741376981684224/6FuIcOPs_normal.jpg",
"profile_banner_url": "https://pbs.twimg.com/profile_banners/1130167399/1536240531",
"profile_link_color": "1B95E0", "profile_sidebar_border_color": "000000",
"profile_sidebar_fill_color": "000000", "profile_text_color": "000000",
"profile_use_background_image": false, "has_extended_profile": false, "default_profile":
false, "default_profile_image": false, "following": false, "follow_request_sent": false,
"notifications": false, "translator_type": "none"}, "geo": null, "coordinates": null, "place":
null, "contributors": null, "is_quote_status": false, "retweet_count": 22, "favorite_count":
38, "favorited": false, "retweeted": false, "possibly_sensitive": false, "lang": "en"},
"is_quote_status": false, "retweet_count": 22, "favorite_count": 0, "favorited": false,
"retweeted": false, "lang": "en"}

```

File Output

twitter_hashtag_stats.py user_timeline_CIA.jsonl

Use of hashtags:

1162 tweets without hashtags (36.36%)
 2034 (elite) tweets with at least one hashtag (63.64%)

Number of hashtags used:

1421 tweets with 1 hashtags (44.46% total, 69.86% elite)
 125 tweets with 3 hashtags (3.91% total, 6.15% elite)
 447 tweets with 2 hashtags (13.99% total, 21.98% elite)
 36 tweets with 4 hashtags (1.13% total, 1.77% elite)
 4 tweets with 5 hashtags (0.13% total, 0.20% elite)
 1 tweets with 7 hashtags (0.03% total, 0.05% elite)

twitter_hashtag_stats.py user_timeline_NSAGov.jsonl

Use of hashtags:

513 tweets without hashtags (26.15%)
 1449 (elite) tweets with at least one hashtag (73.85%)

Number of hashtags used:

706 tweets with 1 hashtags (35.98% total, 48.72% elite)
 44 tweets with 4 hashtags (2.24% total, 3.04% elite)
 9 tweets with 5 hashtags (0.46% total, 0.62% elite)
 501 tweets with 2 hashtags (25.54% total, 34.58% elite)
 186 tweets with 3 hashtags (9.48% total, 12.84% elite)
 3 tweets with 6 hashtags (0.15% total, 0.21% elite)

twitter_hashtag_frequency.py user_timeline_CIA.jsonl

```
oss: 331
museum: 253
inmemoriam: 240
oss75: 238
wwii: 178
intelcon: 86
histint: 62
donovan: 57
pdb: 55
ciak9: 32
oxcart: 29
openingceremony: 27
k9: 27
u2: 27
cartography: 23
internships: 22
fisher: 22
cat: 21
ublr raid: 21
otd: 20
```

twitter_hashtag_frequency.py user_timeline_NSAGov.jsonl

```
nsa: 658
tbt: 120
news: 115
gencyber: 108
cryptochallenge: 93
iadgov: 88
cybersecurity: 64
cyber: 55
cyberaware: 52
factfriday: 39
ncm: 33
nsas: 30
puzzleperiodical: 29
stem: 18
icymi: 15
ncsam: 14
womeninstem: 13
ncx2018: 11
otd: 10
puzzle: 10
```

Conclusion

Both the Central Intelligence Agency and the National Security Agency collect intelligence using various methods; however, the CIA focuses on HUMINT or human intelligence, whereas the NSA focuses on SIGINT or signals intelligence. Due to the nature of its methods of intelligence collection, the NSA is known to be more involved with technology. Perhaps this is why they make greater use of hashtags in their tweets than does the CIA (~74% and ~64%, respectively). For tweets that include hashtags, the NSA also has a greater percentage of multi-hashtag tweets.

Looking at the contents of the most commonly used hashtags, we see that the CIA has a number of terms related to history of the organization (oss, museum, wwii, histint), while the NSA has more posts

related to technology (cryptochallenge, cybersecurity, cyber, cyberware, stem, womeninstem). Information about the frequency and contents of these two organizations' tweets can provide information on the value and emphasis placed on technology.

While this project focused only on preliminary analysis of hashtag use, future work could look at specific mentions of technology-related terms within the hashtags and the text of the tweets, themselves.

Appendix: Python Code

Appendix A: twitter_client.py

```
'''
Name of file: twitter_client.py
Date created/updated: September 8, 2018
Created by: James Ingram
Code adapted from: Bonzanini, M. (2016). Mastering Social Media Mining with Python.
Birmingham, UK: Packt Publishing Ltd.
Purpose of program: create a function to setup and authenticate a twitter API client and
return a twitter.API object
'''

# import functions, libraries, and dependencies
import os
import sys
from tweepy import API
from tweepy import OAuthHandler

# Setup twitter authentication; return tweepy.OAuthHandler object
def get_twitter_auth():
    # assigns twitter API authentication data stored in local environment to variables
    try:
        consumer_key = os.environ['TWITTER_CONSUMER_KEY']
        consumer_secret = os.environ['TWITTER_CONSUMER_SECRET']
        access_token = os.environ['TWITTER_ACCESS_TOKEN']
        access_secret = os.environ['TWITTER_ACCESS_SECRET']
    # returns key error if unable to load authentication data
    except:
        sys.stderr.write("TWITTER_* environment variables not set\n")
        sys.exit(1)
    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_secret)
    return auth

# Setup twitter API client; return tweepy.API object
def get_twitter_client():
    auth = get_twitter_auth()
    client = API(auth)
    return client
```

Appendix B: twitter_get_user_timeline.py

```
'''
Name of file: twitter_get_user_timeline.py
Date created/updated: September 8, 2018
Created by: James Ingram
Code adapted from: Bonzanini, M. (2016). Mastering Social Media Mining with Python.
Birmingham, UK: Packt Publishing Ltd.
Purpose of program: uses twitter client from twitter_client.py to pull tweets from a user's
timeline and store it as a jsonl file
'''

# import functions, libraries, and dependencies
import sys
import json
from tweepy import Cursor
from twitter_client import get_twitter_client

if __name__ == '__main__':
    # file argument = user's twitter handle
    user = sys.argv[1]
    client = get_twitter_client()
    # uses argument in filename
    fname = "user_timeline_{}.jsonl".format(user)
    # writes twitter data to json lines file (each entry separated by new line)
    with open(fname, 'w') as f:
        # iterates through sixteen pages of 200 tweets each to retrieve a maximum of 3,200
        # tweets
        for page in Cursor(client.user_timeline, screen_name=user, count=200).pages(16):
            for status in page:
                f.write(json.dumps(status._json)+"\n")
```

Appendix C: twitter_hashtag_stats.py

```

'''
Name of file: twitter_hashtag_stats.py
Date created/updated: September 8, 2018
Created by: James Ingram
Code adapted from: Bonzanini, M. (2016). Mastering Social Media Mining with Python.
Birmingham, UK: Packt Publishing Ltd.
Purpose of program: counts the number of hashtags used in a user's tweets and calculates the
frequencies of the different hashtag counts
'''

# import function, libraries, and dependencies
import sys
from collections import defaultdict
import json

def get_hashtags(tweet):
    # entities is a dictionary of URLs, hashtags, and mentions in a tweet
    entities = tweet.get('entities', {})
    # gets a list of the hashtags from the entities dictionary
    hashtags = entities.get('hashtags', [])
    # returns the text of the hashtags
    return [tag['text'].lower() for tag in hashtags]

def usage():
    print("Usage:")
    print("python {} <filename.jsonl>".format(sys.argv[0]))

if __name__ == '__main__':
    if len(sys.argv) != 2:
        usage()
        sys.exit(1)
    fname = sys.argv[1]
    # open jsonl file specified by command line argument
    with open(fname, 'r') as f:
        hashtag_count = defaultdict(int)
        for line in f:
            # read in each line of file and assign to tweet
            tweet = json.loads(line)
            # for each tweet, retrieve the hashtags
            hashtags_in_tweet = get_hashtags(tweet)
            # count the number of hashtags in each tweet
            num_of_hashtags = len(hashtags_in_tweet)
            # add to the count of each hashtag count
            hashtag_count[num_of_hashtags] += 1
        # calculate number of tweets containing hashtags
        tweets_with_hashtags = sum([count for num_of_tags, count in hashtag_count.items() if
num_of_tags > 0])
        # calculate number of tweets without hashtags
        tweets_no_hashtags = hashtag_count[0]
        # calculate total number of tweets
        tweets_total = tweets_no_hashtags + tweets_with_hashtags
        # calculate percentage of tweets with hashtags
        tweets_with_hashtags_percent = "%.2f" % (tweets_with_hashtags / tweets_total * 100)
        # calculate percentage of tweets without hashtags
        tweets_no_hashtags_percent = "%.2f" % (tweets_no_hashtags / tweets_total * 100)
        # assign values to variables
        notags = "{} tweets without hashtags ({}%)".format(tweets_no_hashtags,
tweets_no_hashtags_percent)
        withtags = "{} (elite) tweets with at least one hashtag
({}%)".format(tweets_with_hashtags, tweets_with_hashtags_percent)

```

```
# display results
print("Use of hashtags:")
print(notags)
print(withtags)
print("\nNumber of hashtags used:")
# display frequencies and percentages of various hashtag counts
for tag_count, tweet_count in hashtag_count.items():
    if tag_count > 0:
        percent_total = "%.2f" % (tweet_count / tweets_total * 100)
        percent_elite = "%.2f" % (tweet_count / tweets_with_hashtags * 100)
        print("{} tweets with {} hashtags ({}% total, {}% elite)".format(tweet_count,
tag_count, percent_total, percent_elite))
```

Appendix D: twitter_hashtag_frequency.py

```
'''
Name of file: twitter_hashtag_frequency.py
Date created/updated: September 8, 2018
Created by: James Ingram
Code adapted from: Bonzanini, M. (2016). Mastering Social Media Mining with Python.
Birmingham, UK: Packt Publishing Ltd.
Purpose of program: extract the hashtags from a user's timeline and create a list and count of
the most common ones
'''

# import functions, libraries, and dependencies
import sys
from collections import Counter
import json

#
def get_hashtags(tweet):
    # entities is a dictionary of URLs, hashtags, and mentions in a tweet
    entities = tweet.get('entities', {})
    # gets a list of the hashtags from the entities dictionary
    hashtags = entities.get('hashtags', [])
    # returns the text of the hashtags
    return [tag['text'].lower() for tag in hashtags]

if __name__ == '__main__':
    # argument = name of json file from which to extract json data
    fname = sys.argv[1]
    # read in json lines file
    with open(fname, 'r') as f:
        hashtags = Counter()
        for line in f:
            tweet = json.loads(line)
            hashtags_in_tweet = get_hashtags(tweet)
            hashtags.update(hashtags_in_tweet)
        # display 20 most common hashtags and their frequencies
        for tag, count in hashtags.most_common(20):
            print("{}: {}".format(tag, count))
```