SI 206
Homework 5: OAuth and Caching

*Note the correction made in “What to turn in”. You will be using a single github repo for the main part of the assignment & extra credits

Homework Objective:

- Know what is OAuth and how to use it
- Know how to look through API documentation and use it for making requests
- Know how to use caching

Supporting Material:

Starter code provided in github repository:
hw5_twitter.py
secret_data.py

https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user_timeline.html

Goal: Create a program to analyze the twitter timeline of a selected user to list their most frequently used words (20pts)

FILENAME: `hw5_twitter.py`

ARGUMENTS: the program takes two arguments: a twitter username and the number of tweets to analyze.

OUTPUT: the program outputs the following:

- the user name
- the number of tweets analyzed
- the five most frequent words that appear in the analyzed tweets

SAMPLE OUTPUT:

```
mwnewman$ python3 hw5_twitter.py umsi 50
USER: umsi
```

```
TWEETS ANALYZED: 50
5 MOST FREQUENT WORDS: umsi(8) being(3) is(3) improve(2) Join(2)
```

Part 1: Get user tweets (5 points)
Step 1: Fetch 25 tweets from the UMSI twitter account (https://twitter.com/umsi) using the twitter API. Pass the twitter username and number of tweets as arguments to the program
Example: `$ python3 hw5_twitter.py umsi 25`

Step 2: Now, write the json returned from the Twitter API to a file named tweet.json in a neatly formatted way (Tip: use json.dumps() with indent)
Example:

```
{
   "statuses": [
      {
         "created_at": "Thu Feb 01 04:51:26 +0000 2018",
         "id": 958925736047386624,
         "id_str": "958925736047386624",
         "text": "RT @xyz: Example tweet content",
         "truncated": false,
         "entities": {
            "hashtags": [
               {
                  "text": "GoBlue",
                  ……..
```

Part 2: Analyse tweets using NLTK to find most common words(10 points)--link nltk doc
Take a moment to look through the json file you just created to get a sense of the data you are interacting with.
Note: You will not be using tweet.json in further instructions. It's for your reference only. Feel free to comment that section of code.

Step 1: You will require the NLTK module for this section. If you don't have nltk installed, here are related instructions:http://www.nltk.org/install.html
Step 2: Gather all tweets data by accessing the text portion of the tweets. Once you have this, tokenize the words.
Step 3: Get a frequency distribution of the tokenized list.
Step 4: Ignore stop words

- (1) ignore any words that do not start with an alphabetic character [a-zA-Z],
- (2) ignore 'http', 'https', and 'RT' (these show up a lot in Twitter)

Step 4: Print the 5 most frequently used words using the frequency distribution you just created.

Part 3: Implement Caching (5 points)
Let's now add caching functionality to the above code. Caching helps reduce processing time and saves bandwidth costs (think of caching vs. fetching the same large volumes of data multiple times when the data hasn't changed)

Step 1: Feel free to use the caching code used in class. In the cache, we are associating fullURL of the request with the response we receive from twitter for that URL.

Step 2: Verify that your code is picking up data from the cache when you repeat the same request (getting 25 tweets from the umsi account). Add a line of code to the caching code block to print: "`Fetching cached data...`" when data is fetched from the cache.
Also verify that your code fetches data from twitter instead of the cache for new requests(try getting data for a different user account).

Extra Credit 1 (2 pts)

Twitter Boggle: Take two twitter accounts and analyze their tweets to find words they have in common and words that are unique to each account. Show the 5 most frequent different (unique) words for each account and the 5 most frequent common words (shared by both).

Extra Credit 2 (2 pts)

Fetch 25 tweets from the UMSI twitter account (https://twitter.com/umsi) and print the 10 most commonly occurring "Basic Verbs". Ignore stop words.

- Use NLTK for analyzing parts of speech. Use NLTK's default POS tagger and tagset (this will use the UPenn treebank tagset)
- a "verb" is anything that is tagged VB*

What to turn in:

1. Screenshot of your github repository after the last push to github(example screenshot in github instructions)

2. A link to your GitHub repository (in comments)

3. (Optional) file named hw5_ec1.py on the same github repo for Extra Credit 1.

4. (Optional) file named hw5_ec2.py on the same github repo for Extra Credit 2.

Note:

Be sure to commit everything (and **push**!) to your GitHub repo. At a minimum, your repo should include the following files, which you have modified:

- hw5_twitter.py

Do not upload secret.py (this has your API key!)

Push your code before the deadline. We'll grade the latest commit. Please don't commit after the deadline or you might receive penalty since the latest commit passes the deadline.

All code must be executable. Any code that does not run in **Python3** will be **given a score of 0.** You can receive partial credit for working programs.

Example screenshot: