Lab Assignment 1: Twitter Feeds

In this assignment you will read and process Twitter data. A starting file titled assignments.py is provided to you and can be found at the course's GitHub page[1], and you will fill in the missing parts. You will need a small JSON file called keys.json at the same level, that contains your Twitter access information, as described in the notes. Make sure you have done so before you proceed. Running the first 48 lines of the script in your Python 3 interpreter should tell you if it works OK or not.

You should be looking at the assignment1.py file as you read along.

The first 67 lines of the script read from Twitter and create a list called "tweets" for you. You should not change these lines, and your answers in the rest of the file will make use of the "tweets" variable. This contains all tweets that mention the words "Hanover" and "College", and it is limited to the last 8 or so days, as Twitter does not allow us to go further back in queries.

1. Write a list comprehension that will produce a list containing all the texts from the tweets. Assign the result to a variable called "texts". For each tweet, there is a field called full_text that contains the text of the tweet.
2. This is a modification of the first exercise. When the entry you are looking at is a "retweeted" tweet, then the full_text entry doesn't actually contain the full tweet, but only an abbreviated version. You can see this in some of the texts you extracted, which probably ended in ellipses. To fix this, first write a function get_full_text that is given a tweet and returns the full tweet text as follows:
   - If there is no retweeted_status key/field present, then the tweet text is indeed in the full_text list as above.
   - Otherwise, you must look into the field retweeted_status, and look at *its* full_text field.

   Next, use this function to write a list comprehension that given a list of tweets returns the full texts of the tweets.
3. After reviewing the tweet field guide[2], and in particular the entities[3] in it, create a list comprehension that produces a list that in place of each tweet contains the hashtags that were mentioned in the corresponding tweet. So your result would be a list, each of whose entries is itself a list of the hashtags that appeared in a particular tweet. Note that you should only record the text of the hashtag; by default Twitter provides more information, like the location of the tag within the tweet. Assign the result to a variable called tags_per_tweet.
4. Using the variable you created in the previous assignment, create a dictionary called hashtags. Its keys would be the different hashtags, and the value for a hashtag is the number of times that tag occurred in the tweets. Implementing this will probably require a double iteration over the list of tags_per_tweet.
5. Write code that would print the 6 most frequently occuring hashtags. Using the sorted function appropriately will get you partway there.

[1] https://github.com/skiadas/DataWranglingCourse/blob/gh-pages/assignments/assignment1.py
[2] https://dev.twitter.com/overview/api/tweets
[3] https://dev.twitter.com/overview/api/entities

6. Starting from the "tweets" list, produce a list of the tweets that have no hashtags. Use a list comprehension for this.
7. (This one is a lot more complex than the previous problems) Produce a dictionary with one key for each hashtag. The value should itself be a dictionary, with the following:

    - A key called "count" that contains the number of tweets that contained that hashtag.
    - A key called "percent" that contains the percent of tweets that contained that hashtag.
    - A key called "users" that contains a list of the handles (screen names) of the users who tweeted the tweets with that hashtag. No name should appear twice in the list for a given hashtag, even if that user had multiple tweets with that hashtag.
    - A key called "other_tags" that contains a list of all the other hashtags that appeared in the same tweet with the given hashtag. No hashtag should appear twice in a list, even if there are two tweets that both contain the same pair of hashtags.

    You should ignore tweets that are "retweets". Assign the resulting dictionary to a variable called tag_info.
8. Convert the dictionary in tag_info to a JSON string and write it to a file called tag_info.json. You could paste your result in this site[4] to make sure it worked.
9. Write code that would create a list with one entry per tweet. The entry would be a dictionary that contains the following keys:

    - text with value the string text of the tweet.
    - author with value the string handle/screen name of the tweet's author
    - date with value the string date when the tweet was created
    - hashtags with value a list of the string hashtags in the tweet
    - mentions with value a list of the string handles of the users mentioned in the tweet

    Export this dictionary as JSON to a file called simpler_tweets.json.

You should submit your completed Python file as an email attachment to me. The name of the file should include your first and last name, in addition to the assignment's number. It should contain no whitespaces.

[4] http://www.jsoneditoronline.org/