**Mining Twitter**

**Introduction**

If you have gone through the Mining Facebook tutorial, it might have become apparent that mining social media sights can be very useful for a variety of reasons (if not also very intriguing). While Facebook has been a main stay in the social media world for keeping up with friends and family, Twitter is the go to social media destination for seeing what’s trending now. Twitter data is widely used in industry in everything from analyzing social tendencies to helping companies build better applications that appeal to a broader audience.

**Purpose**

The purpose of this exercise is to once again use Python to mine the social web. This time, our focus will be solely on Twitter.

**Before We Begin…**

Please, if you all run into any trouble during the course of this exercise, do not hesitate to seek help from me. I’m here for you!!! ☺

**Procedure**

**Part 1: Becoming Friends with Twitter**

In order for us to have access to Twitter’s API, we first need to become a Twitter Developer. Doing so will grant us the ability to create a “dummy” application which will further provide us the means necessary to obtain an access token.

1. Navigate to <https://dev.twitter.com/>.
2. On the developer’s homepage, click Sing In located on the top right hand corner of the page. On the following page, login with your twitter screen name and password. Note that you have to use your Twitter screen name and not your email address tied to your account.
3. Once you are logged in, for lack of better terms, you have become an official Twitter Developer. In order to gain an access token to Twitter’s API, we need to create our “dummy” app. On the top right hand corner of the developer homepage, hover over your Twitter screen picture and click on My applications.
4. When your in My applications, click Create New App. On the following page, Fill out the required fields for your “dummy” app. For your app name and description, keep it simple and related to this course. If the app name you have selected is already taken, simply try another variation of the same name or a different name altogether. For the website, simply input a placeholder in the form of a web address. For example, <http://www.hello.com>. Make sure to agree to the terms and conditions at the bottom of the page and click Create your Twitter application.
5. If everything checks out, you should be taken to your app’s “homepage”. On this page, click on the API Keys tab. Scroll down to where you see Your access token. Under Token actions, click Create my access token. When the page refreshes, you should see some information underneath where it says Your access token. If you don’t see anything there, refresh the page until you see it.
6. Once you see your access token, scroll up to the top of the page and click Test OAuth. This will take you to a page titled OAuth Settings. On this page, you will see four fields: Consumer key, Consumer secret, Access token, and Access token secret. We will use all of these fields for when we go to create our access token in IPython. From here, lets jump into IPython and start mining Twitter.

**Part 2: Doing the Twitter OAuth Dance**

Now that we have become a Twitter developer and have gained the means necessary to create an access token to reference the Twitter API, lets go ahead and put it all to good use.

1. Open up Terminal/Command Prompt. The first thing we need to do is install the twitter package for IPython. To do so, type pip install twitter. Once the download is complete, launch IPython.
2. In IPython, start out by typing import twittrer. Next we need to set up the credentials for the Twitter access token in IPython. To do so, type the following:

consumer\_key = ‘*your\_consumer\_key\_goes\_here*’

consumer\_secret = ‘*your\_consumer\_secret\_goes\_here*’

oauth\_token = ‘*your\_access\_token\_goes\_here*’

oauth\_token\_secret = ‘*your\_access\_token\_secret\_goes\_here*’

1. Now that we have the credential set up, lets go ahead and put it all together so we can call the API. Type the following:

auth = twitter.oauth.OAuth(oauth\_token, oauth\_token\_secret, consumer\_key, consumer\_secret)

1. Now that we have that the API call authorization set up, lets connect to Twitter through IPython and set it in a variable. Type the following:

twitter\_api = twitter.Twitter(auth=auth)

1. If you were to print the twitter\_api variable, you should see something along the lines of the following:

<twitter.api.Twitter object at 0x103e41310>

**Part 3: Mining Trending Topics**

Having performed the OAuth dance with Twitter, lets see what’s trending around the world and U.S. and compare the two.

1. To get the trending topics for a given locale (or for the world for that matter), we will utilize Yahoo! GeoPlanet Where on Earth Identifiers. For our purposes, the identifiers used for the entire globe is 1 while the identifier for the United States is 23424977. That being said, we need to store these values in variables like so:

world\_id = 1

us\_id = 23424977

1. To get the trends for our two identifiers, we need to call the Twitter API for each occurrence. To do so, type the following:

world\_trends = twitter\_api.trends.place(\_id=world\_id)

us\_trends = twitter\_api.trends.place(\_id=us\_id)

1. Now that we have called the Twitter API and gotten the top trends for the world and U.S., let’s print out the results in JSON format. Type the following:

print json.dumps(world\_trends, indent = 1)

print

print json.dumps(us\_trends, indent = 1)

1. Now that we have our results printed out in a nice JSON format, wouldn’t it be nice to see if there are any common trends between the the U.S. and the globe? Well, Twitter’s API lets us do just that. To get the ball rolling, we need to set the trends we gathered above into a set. To do so, type the following:

world\_trends\_set = set([trend[‘name’] for trend in world\_trends[0][‘trends’]])

us\_trends\_set = set([trend[‘name’] for trend in us\_trends[0][‘trends’]])

1. Now that we have the sets formatted, lets see if they have anything in common and store those commonalities in a variable. Type the following:

common\_trends = world\_trends\_set.intersection(us\_trends\_set)

1. To finish, let’s print out the common trends in JSON format. Type the following:

print json.dumps(common\_trends, indent = 1)

**Wrap-Up**

Congratulations! You have just successfully mined Twitter using its API. This exercise was only a taste as to what you can search for and retrieve using the Twitter API. Have that being said; please feel free to mess around with it and find some awesome data to tell a story with.