Getting computing into the classroom: playing with Twitter

Erik Spence

SciNet HPC Consortium

11 December 2014

Big Data

Social data is among the most interesting to play with. There are many examples, Twitter, Facebook, LinkedIn, *etc.*

The plan for today:

- An introduction to Python's dictionary datatype.
- Introduction to the Twitter API, getting set up with the Twitter feed, and playing with the Twitter API.
- Introduction to geocode, converting addresses into coordinates.
- Creating Google maps.
- Combining them all.

Dictionaries

Dictionaries are a python data type which associates keys to values.

Create a dictionary:

```
>>> a = dict(one = 1, two = 2, three = 3)
>>>
>>> b = {'one': 1, 'two': 2, 'three': 3}
>>>
>>> c = {}
>>> c['one'] = 1; c['two'] = 2; c['three'] = 3
>>>
>>> c
{'one': 1, 'three': 3, 'two': 2}
>>>
```

All of the above commands are equivalent.

Dictionaries, continued

Dictionaries have some handy functions built in.

```
>>> a.keys()
['three', 'two', 'one']
>>> 'one' in b
True
>>> 'four' in b
False
>>> a.setdefault('four',0)
>>> a
{'four': 0, 'one': 1, 'three': 3, 'two': 2}
>>> c['one']
1
```

Dictionaries are most useful when you need a dynamic datatype that can grow as needed.

Twitter

Accessing Twitter data is alot of fun. However, there are some issues with getting access:

- In the past one could just directly download the data, no strings attached.
- As of 2012 things aren't quite so easy. Accessing the data now requires an OAuth access token, which requires a registered Twitter application to generate.
- Registering an application requires a Twitter account. Don't have an account? No problem. It takes 30 seconds to set one up.
- Don't have a registered application? No problem. It takes 2 minutes to set one up.
- Once you have the application, we generate the tokens, and away we go!

Getting a Twitter account

Here are the steps for getting a Twitter account.

- Go to `www.twitter.com`.
- Sign up for an account, choose a user id.
- On the page listing people to follow, I clicked on the 'x's, so I didn't have to follow anyone.
- Skip uploading the photo.
- Lower right corner: skip the 'Find people you know' step.
- Check your email, click on the 'Confirm now' button.

Once we have a Twitter account, we need to register a Twitter App.

Setting up a Twitter App

To get the OAuth tokens, which we need to access Twitter data from our machines, we need to register a Twitter App.

- Go to `http://apps.twitter.com`. You may need to log in if you weren't already logged in.
- Click on 'Create New App'.
- Give your App a name, description, filler website (I used http://www.unnecessaryquotes.com).
  - Your App name has to be unique. There are alot out there. It may take a few tries to find a unique one.
  - Don't forget the "http://" in front of your URL.
- Click on "Yes I agree", and create your App.

Congratulations, you've set up your App!

Setting up a Twitter authentication

To access the data from your machine, you need authentication tokens.

- On your App's webpage, click on the "Keys and Access Tokens" tab.
- The first two keys you need are the "Consumer Key" and "Consumer Secret". Copy-and-paste them into your favourite text editor.
- Go to the bottom of the page, and click on "Generate Access Tokens", or something like that.
- Copy-and-paste the "Access Token" and "Access Token Secret" keys into your favourite text editor, so you don't lose them.
- You're done!

Congratulations, you're all set to access Twitter data!

Setting up authentication in Python

Before we test your authentication credentials, let's put the credentials in a block of code, so we can forget about them.

We will use the twitter Python module to access the Twitter API.

```
# twitter_setup.py
import twitter

# Our access tokens.  These have been
# shortened, so they won't work.
# Please use your own.
C_KEY = "1cuA1c6wb7v55d0h584KNF74E"
C_SECRET = "RFtT8crWVsYjFsqMCsOKLz"
A_TOKEN = "2919154010-uPOASA8P3CIy"
A_SECRET = "4DoJZp1aq2Z5UVzioa0QVG"

# Instantiate the OAuthHandler.
auth = twitter.oauth.OAuth(A_TOKEN,
  A_SECRET, C_KEY, C_SECRET)

# Instantiate the twitter object.
api = twitter.Twitter(auth = auth)
```

Setting up Python

Be sure to perform the following steps:

- Create a directory to hold your code.
- Move the Python IDLE to the directory you're using.

These steps will make running code from the command line easier.

```
>>>
>>> import os
>>>
>>> os.chdir("C:\\Users\\Erik\\Desktop\\mycode")
>>>
```

This directory path will be different if we're running on a Linux image, but os.chdir should still work.

Testing your Twitter access using Python

```python
# twitter_test.py
from twitter_setup import api

# The authentication has already
# been done in twitter_setup.py.
tweeter = api.users.lookup(
  screen_name = "CBCNews")[0]

print("Tweeter screen name:",
  tweeter['screen_name'])
print("Tweeter followers count:",
  tweeter["followers_count"])
print("Tweeter description:",
  tweeter["description"])

tweets = api.statuses.user_timeline(
  count = 3, id = tweeter['id'])

for tweet in tweets:
  print("\n Tweet:", tweet['text'])
```

```
>>> execfile("twitter_test.py")
Tweeter screen name: CBCNews
Tweeter follower count: 720717
Tweeter description: Canadian breaking
news and analysis from CBCNews.ca, TV
and radio.

Tweet: RT @CBCAlerts: #NASA's #Orion
spacecraft splashes down in Pacific
Ocean after successful test flight.

Tweet: RT @cbcradio: "We fought back
to keep the idea of liberation alive."
#NelsonMandela http://t.co/DlyNNYjeS5
@cbcideas http://t.co/tBKDjQ02a0

Tweet: RT @Astro_Jeremy: Awesome to
see #NASA_Orion safely under parachute!
http://t.co/fyJvBhS9nl
>>>
```

Understanding the Twitter API

To fully utilize the API, we need to understand a bit of Twitter terminology.

- *tweet*: comment, up to 140 characters long. Tweets also contain two additional pieces of metadata:
  - *entities*: users, hashtags, URLs and media that are part of the tweet.
  - *places*: locations in the real world, attached to the tweet. May be where the tweet was authored, or a reference to a place described in the tweet.
- *timelines*: chronologically sorted collections of tweets.
  - *home timeline*: the view you see when you log in, looking at all the tweets from the users that you are following.
  - *user timeline*: the collection of tweets only from a certain user.

Our CBCNews tweets were from the CBCNews user timeline.

Things we can study with Twitter information

The amount of Twitter data is impressive. Some questions that might be asked of students:

- What topics are trending in your city? Country?
- What is the friendship distribution of your closest friends?
- What are the most-frequently-used words in the tweets using a given trending hashtag?
- What is the average number of unique words per tweet? What's the distribution?
- What is the number of hashtags per tweet? @mentions? Links?

Other ideas?

What's trending in a given location?

Using the API, we can see what things are trending in specific locations.

```
>>> execfile("twitter_trending.py")
#hayesvideo
Christmas
#dec6
#PlayStationExperience
#WORDSMUSICVIDEO
#WWCDraw2015
Uncharted 4
Santa
Toronto
Xbox
>>>
```

```python
# twitter_trending.py
from twitter_setup import api

# "WOE": Yahoo!'s Where On Earth id.
CAN_WOE_ID = 23424775

# Get the trends.
can_trends = api.trends.place(
  _id = CAN_WOE_ID)[0]

for trend in can_trends:
  print trend["name"]
```

Try some other WOEID values:

- whole world = 1
- Ontario = 2344922
- Toronto = 4118

Some notes about trending

Some things to bear in mind when searching for trends:

- Trends are only updated every 5 minutes. So it doesn't make sense to ask for such results to be updated more frequently than that.
- Twitter imposes 'rate limits' on how many requests an application can make to any give API resource within a given time window. For example, the trending request limits applications to 15 requests per 15-minute window.
- Yahoo!'s Where On Earth IDs are the de facto standard for tying the web to geographic locations.
- To look up other WOE IDs, go to `http://woeid.rosselliot.co.nz`.

Searching for tweets

The day I did this slide, "Leafs" was trending in Canada. How do I go about searching for tweets that use "Leafs"?

```
>>> from twitter_setup import api
>>> results = api.search.tweets(q = 'Leafs', count = 3)
>>> for tweet in results['statuses']:
  ...   print
  ...   print tweet['user']['name'] + ', ' + tweet['user']['screen_name']
  ...   print tweet['text']
  ...
Rhys Jessop, Thats_Offside
RT NigelCadbury: Leafs are beating Miller like that time Mandela fought Tyson

Callum-Mullac, PippinTooksPint
Toronto maple leafs ftw

Mike Tarnes, TheRealMTarnes
Anchor: How'd U get tickets 2 the Leafs game?  Conner McDavid: just sitting
around at home so I called Bobby Orr to get me some.  #MustBeNice
```

Twitter returns a tonne of info

A lot of information came with those tweets. Take a look:

```
>>> for i in results:
  ...  print i
  ...
search_metadata
statuses
```

```
>>> len(results['statuses'])
3
```

```
>>> for i in results['statuses'][0]:
  ...  print i
  ...
contributors
truncated
text
in_reply_to_status_id
id
favorite_count
source
retweeted
coordinates
entities
in_reply_to_screen_name
in_reply_to_user_id
retweet_count
id_str
favorited
user
geo
in_reply_to_user_id_str
possibly_sensitive
lang
created_at
in_reply_to_status_id_str
place
metadata
```

```
>>>
```

...and don't forget the user info

```
>>> for i in results['statuses'][0]['user']:
  ...   print i
  ...
follow_request_sent
profile_use_background_image
profile_text_color
default_profile_image
id
profile_background_image_url_https
verified
profile_location
profile_image_url_https
profile_sidebar_fill_color
entities
followers_count
profile_sidebar_border_color
id_str
profile_background_color
listed_count
is_translation_enabled
```

```
utc_offset
statuses_count
description
friends_count
location
profile_link_color
profile_image_url
following
geo_enabled
profile_banner_url
profile_background_image_url
name
lang
profile_background_tile
favourites_count
screen_name
notifications
url
created_at
contributors_enabled
time_zone
```

Finding Twitter friends

Twitter allows people to befriend one another. How do we determine one's friendship network?

```
>>> from twitter_setup import api
>>> q = api.friends.ids(screen_name = 'CBCNews')
>>>
>>> len(q['ids'])
2107
>>>
>>> q['ids'][0]
126434587
>>> myids = q['ids'][0:100]
>>>
>>> subquery = api.users.lookup(user_id = myids)
>>> subquery[0]['screen_name']
'patranya'
>>> subquery[0]['name']
'Patranya Bhoolsuwan'
```

Using geocoder with Python

The pygeocoder module accesses the Google Geocoding API.

- The API allows you to convert from an address to lattitude and longitude.
- The opposite operation can also be performed.
- Can also be used to filter/correct addresses (assuming Google is correct).
- The lattitude and longitude are needed if you're going to put the address on a map.

```
>>> from pygeocoder import Geocoder
>>> address = Geocoder.geocode(
  "15 smith red hook NY")
>>> address.valid_address
True
>>> address.country
'United States'
>>> address.administrative_area_level_1
'New York'
>>> address.administrative_area_level_2
'Dutchess County'
>>> address.postal_code
'12571'
>>> address.formatted_address
'15 Smith Street, Red Hook, NY 12571,
  USA'
>>> address.coordinates
(41.992597000000004,
  -73.885348999999991)
```

Using geocoder with Python, continued

But you don't need to specify the whole address to get useful information:

- The API allows you to convert just about anything into a lattitude and longitude.
- However, if you put in something which is not a proper address, the 'valid_address' field will be false.

```
>>> from pygeocoder import Geocoder
>>> address = Geocoder.geocode(
  "McGill University")
>>> address.valid_address
False
>>> address.country
'Canada'
>>> address.formatted_address
'McGill University, 845 Rue Sherbrooke
Ouest, Montr\xe9al, QC H3A 0G4, Canada'
>>> address =
  Geocoder.geocode('Toronto')'
>>> address.formatted_address
'Toronto, ON, Canada'
>>> address.coordinates
(43.653225999999997,
  -79.383184299999996)
```

The Google Geocoding API

Some notes about the Google Geocoding API:

- Like the Twitter API, there is a rate limit on how often queries can be made to the Geocoding API: 2500 requests per 24 hour period, 5 requests per second.
- You may run into the second limit later this class. As such you may need to use the time.sleep command to slow down your code.

```
import time
time.sleep(0.2) # Sleep for 0.2 seconds.
```

- In case you accidently put invalid characters into the geocoder argument, it's best to protect the attempt, in case the attempt fails:

```
try:
  address = Geocoder.geocode(some_address)
except:
  pass          # If you're in a loop, you may want to use 'continue'.
```

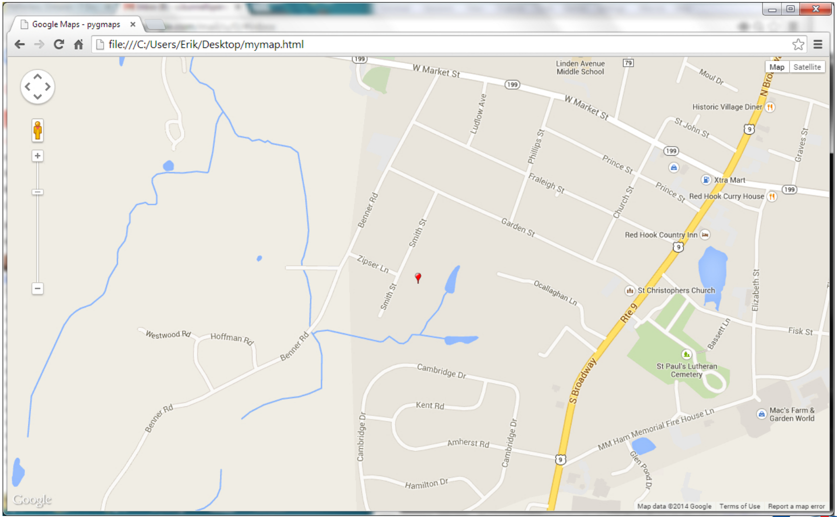
Using Google Maps with Python

The pygmaps module is a wrapper for Google Maps API.

- The module allows you to generate an HTML file which shows your GPS data on a Google Map.
- pygmaps.maps initializes the map, the first two arguments are the center of the map, the third argument is the 'zoom' factor.
- pygmaps uses HTML colour codes.
- mymap.draw creates the map.

```
>>> import pygmaps
>>>
>>> coord = address.coordinates
>>>
>>> mymap = pygmaps.maps(coord[0],
  coord[1], 16)
>>>
>>> mymap.addpoint(coord[0],
  coord[1], '#0000FF')
>>>
>>> mymap.draw('mymap.html')
>>>
>>> import webbrowser as wb
>>>
>>> wb.open_new_tab('mymap.html')
```

And we're on the map!



Questions for students

Being able to present analysis results graphically can allow further insight into the results than is sometimes immediately apparent. Some questions that might be asked of students:

- What is the geographic distribution of people that are tweeting with a given hashtag?
- What is the geographic location of the tweets in the country?
- What is the geographic distribution of languages used in Canadian tweets?
- Possible project, find all tweets with the word 'measles' in them. Geographic distribution?

Other ideas?

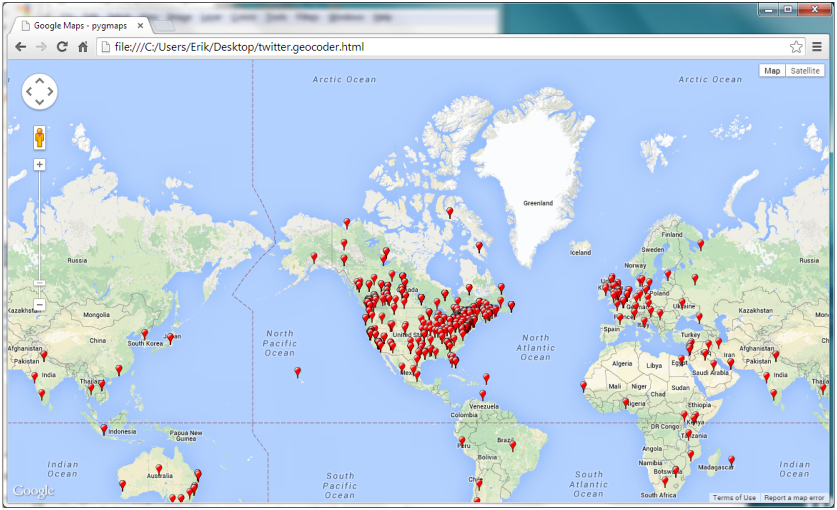
Assignment

Your assignment for the remainder of the class: create a Google map on which is plotted the geographic locations of CBCNews' Twitter friends.

Some notes about the assignment:

- Note that Twitter will only accept a maximum of 100 user lookups per query. You should bundle your queries in batches of 100.
- You should use the user's 'location' key as their location.
- Note that the user's locations are generally cities. The geocoder's 'valid_address' key will be false; use the coordinates.
- Centre your map on (56.95, -98.31), zoom factor of 2.
- CBCNews has about 2100 friends. You only get 2500 geocoding queries per day. Be sure to test your code carefully before you do your production run.
- Put a time.sleep command in you query loop so that you don't exceed 5 queries per second.

CBCNews' friends



Graphing CBCNews' friends

```
# CBCNews_friends.py
from twitter_setup import api
from pygeocoder import Geocoder
import pygmaps, time

q = api.friends.ids(screen_name = 'CBCNews')
num = len(q['ids'])
mymap = pygmaps.maps(56.95, -98.31, 5)

for n in range(0,num,100):
  ids = q['ids'][n:n+100]
  subquery = api.users.lookup(user_id = ids)
  for user in subquery:
    try:
      address = Geocoder.geocode(user['location'])
    except:
      continue
    mymap.addpoint(address.coordinates[0], address.coordinates[1])
    time.sleep(0.2)
mymap.draw('twitter.geocoder.html')
```