DATA 133 - Introduction to Data Science I

Instructor: Renzhi Cao
Computer Science Department
Pacific Lutheran University

Announcements

- Quiz 7 (due on Thursday, 2 chances with average score!)
- You should have twitter credentials now.

Set up your app

You need to set up your app for sending tweet message:

1. Login https://dev.twitter.com/apps. Click details

2. Click Permissions, change it to Read, Write and Direct Messages.

3. Now click Keys and Tokens to regenerate the token. Now your new token should have good permission!

Review Twython

```
python -m pip install twython
```

Or conda install twython

Twython

Instantiate the client:

```
import os
CONSUMER_KEY = "TWITTER_CONSUMER_KEY"
CONSUMER_SECRET = “TWITTER_CONSUMER_SECRET"

import webbrowser
from twython import Twython

# Get a temporary client to retrieve an authentication URL
temp_client = Twython(CONSUMER_KEY, CONSUMER_SECRET)
temp_creds = temp_client.get_authentication_tokens()
url = temp_creds['auth_url']

# Now visit that URL to authorize the application and get a PIN
print(f"go visit {url} and get the PIN code and paste it below")
webbrowser.open(url)
PIN_CODE = input("please enter the PIN code: ")
```

Twython

```
# Now we use that PIN_CODE to get the actual tokens
auth_client = Twython(CONSUMER_KEY,
               CONSUMER_SECRET,
               temp_creds['oauth_token'],
               temp_creds['oauth_token_secret'])
final_step = auth_client.get_authorized_tokens(PIN_CODE)
ACCESS_TOKEN = final_step['oauth_token']
ACCESS_TOKEN_SECRET = final_step['oauth_token_secret']

# And get a new Twython instance using them.
twitter = Twython(CONSUMER_KEY,
            CONSUMER_SECRET,
            ACCESS_TOKEN,
            ACCESS_TOKEN_SECRET)
```

Twython

At this point you may want to consider saving the ACCESS_TOKEN and ACCESS_TOKEN_SECRET somewhere safe, so that next time you don’t have to go through this rigmarole.

Ready?

```
# And get a new Twython instance using them. Fill the following:
CONSUMER_KEY =
CONSUMER_SECRET =
ACCESS_TOKEN =
ACCESS_TOKEN_SECRET =
twitter = Twython(CONSUMER_KEY,
            CONSUMER_SECRET,
            ACCESS_TOKEN,
            ACCESS_TOKEN_SECRET)
```

Twython

Once we have an authenticated Twython instance, we can start performing searches:

```
# Search for tweets containing the phrase "data science"
for status in twitter.search(q='"data science"')["statuses"]:
    user = status[“user”][“screen_name”].encode(‘utf-8’)
    text = status["text"]
    print(f"{user}: {text}\n")
```

encode is not in the text book but very important

Twython

the Twitter Search API just shows you whatever handful of recent results it feels like. When you’re doing data science, more often you want a lot of tweets. This is where the Streaming API is useful. It allows you to connect to (a sample of) the great Twitter firehose. To use it, you’ll need to authenticate using your access tokens.

In order to access the Streaming API with Twython, we need to define a class that inherits from TwythonStreamer and that overrides its on_success method, and possibly its on_error method:

```
from twython import TwythonStreamer
# Appending data to a global variable is pretty poor form
# but it makes the example much simpler
tweets = []

class MyStreamer(TwythonStreamer):
    def on_success(self, data):
       """        What do we do when Twitter sends us data?        Here data
will be a Python dict representing a tweet.
       """
       # We only want to collect English-language tweets
       if data.get('lang') == 'en':
                tweets.append(data)
                print("received tweet #{“+str(len(tweets))+”}”)
       # Stop when we've collected enough
       if len(tweets) >= 100:            self.disconnect()

    def on_error(self, status_code, data):
        print(status_code, data)
        self.disconnect()
```

```
stream = MyStreamer(CONSUMER_KEY, CONSUMER_SECRET,
ACCESS_TOKEN, ACCESS_TOKEN_SECRET)

# starts consuming public statuses that contain the keyword ‘data’

stream.statuses.filter(track='data')

# if instead we wanted to start consuming a sample of *all* public statuses#
stream.statuses.sample()
```

This will run until it collects 100 tweets (or until it encounters an error) and stop, at which point you can start analyzing those tweets. For instance, you could find the most common hashtags with:

```
top_hashtags = Counter(hashtag[‘text'].lower()
                       for tweet in tweets
                       for hashtag in tweet[“entities"]["hashtags"])
print(top_hashtags.most_common(5))
```

Homework of Python

Practice the twitter API
Work on Project