IAE 101 – Introduction to Digital Intelligence
Fall, 2024
Programming Project 04 – Creating a Twitterbot

Assigned: Monday, November 25th, 2022
Due: **Friday, December 13th, 2024, at 11:59 PM**

Description:

Twitter (now X) is a social media platform through which users can post short messages that everyone calls "tweets", but which the developers of Twitter used to insist were called "statuses". Initially, the length of tweets was restricted to 140 characters, but the limit was eventually increased to 280 characters. The current limits are unclear and depend upon whether you are paying for elevated privileges for your account.

Users can engage with each other and gain access to each other's status updates by following each other. When you log into Twitter you are presented with your home timeline, which is a display of all the tweets sent by you and the other users that you follow, typically ordered from most recent to least recent (there are actually two separate timelines now; one is labelled "For You" and contains an algorithmically selected mix of tweets the account you follow and tweets the system thinks you will be interested in; the other, called "Following", is composed entirely of your tweets and those of the accounts you follow, in mostly chronological order).

Twitter engagement and interaction takes many forms. Some users post frequently while others post seldom or never. Some users post the everyday minutiae of their lives. Others engage in extended political discussions. Politicians, journalists, academics, celebrities, and professionals of many other types make use of Twitter to pursue their various interests. The medium has shown remarkable versatility given the strict limits on the length of status updates. (Twitter as a medium for engagement has thinned-out and become a bit shallower over the last couple of years.)

Since early on in its existence, Twitter has also been home to automated participants, Twitterbots –programs that read and post status updates to Twitter. These algorithmic users vary quite a bit in their overall sophistication. Dr. Graham discussed several examples during her lecture last Wednesday. I follow at least two movie related bots—one that posts the script of the 1995 movie *Hackers* one line at a time, and another that posts lines from Kurt Russel's character Jack Burton from the movie *Big Trouble in Little China* (both are now, sadly, defunct). However, their behavior can be much more interesting and complex than this.

Here are some lists of popular and well-regarded Twitterbots.

- https://blog.mozilla.org/internetcitizen/2018/01/19/10-twitter-bots-actually-make-internet-better-place/
- http://nymag.com/intelligencer/2015/11/12-weirdest-funniest-smartest-twitter-bots.html
- https://discover.bot/bot-talk/popular-twitter-bots-2019/
- https://www.poynter.org/tech-tools/2017/7-of-the-best-twitter-bots-in-journalism/

Twitterbots are a kind of public performance, and they are often an interactive form of public performance in which the bot engages with the status updates posted by other users in interesting, or significant, or amusing, or beautiful ways.

For this assignment you will build a Twitterbot using Python and the simple_twit library (which is built on top of the Tweepy library) for accessing Twitter's Application Programming Interface. I have simplified the interface to make it easier for you all to use and to restrict the operations you can take. Your real task is to think about an interesting way to have your bot perform on Twitter and implement that in mytwtterbot.py.

Directions:

1. Create a new account on Twitter (this step is required).
   -Please set your username according to the following scheme:
   - `IAE101_F24`
   - `Followed by a numerical identifier contained in the spreadsheet linked below`

   -This will ensure that all students will have a unique account name. You can set your display name to whatever you like (within reason), though your name would be the best choice.

   -Please use your stonybrook.edu email address when you register the account.

   -After creating your account, find some accounts to follow according to your interests.

   -You must also follow our accounts:
   - IAE101_ckane
   - IAE101_egraham

   -In the spreadsheet confirm that you have created your account and followed the instructors.
      -You will not receive the credentials necessary to interact with twitter until you have done so.

2. Install the Tweepy library using pip.
   -Mac: `python3 -m pip install tweepy --user`

   -Windows: `py -3 -m pip install tweepy --user`

   -Please try to install Tweepy right away so that any problems can be resolved as early as possible in this assignment.

3. Download the files simple_twit.py and mytwitterbot.py from Brightspace.

-simple_twit.py is the interface I have written to simplify and restrict the way you can use the Tweepy library to access Twitter.

-mytwitterbot.py is the file in which you will implement your Twitterbot. It is already prepared for you to start making calls to Twitter's API.

-Make sure that simple_twit.py and mytwitterbot.py are always in the same folder on your computer.

4. Download and read ***very carefully*** the description of the simple_twit.py documentation. This will describe how you can engage with Twitter and other Twitter user's when building your bot.

5. Add your assigned developer credentials to mytwitterbot.py.
-Each set of credentials is composed of a API Key (aka Consumer Key and an API Key Secret (aka Consumer Secret).
-Assign each one to the respective all-caps variable near the top of mytwitterbot.py.
-Do not share these credentials with anyone outside the class, use only your assigned credentials, do not use them for anything other than this project.
-If we find you are misusing our Twitter developer credentials, we will fail you.

6. Spend some time looking through the Tweepy documentation to see how Twitter's API is accessed from Python.

https://tweepy.readthedocs.io/en/latest/index.html

-You may see some additional functionality not available through simple_twit.py that you would like to use. Discuss it with me and I may add it.

7. Spend some time thinking about what you might like to do with your Twitterbot. How would you like it to interact with Twitter and its Users? What information from Twitter will you need in order to make those interactions work? Sit down and write out a design for your Twitterbot.
-If you have an idea, but are not sure how to make it work, come discuss it with us, and we may be able to suggest a way to do it, or offer a similar alternative that can be done instead.

8. Implement your Twitterbot, according to your design, in the mytwitterbot.py file.

Design and Implementation constraints:

You are only permitted to use the functions defined in the simple_twit library for interacting with Twitter. You are not permitted to use the methods defined in the Tweepy package directly.

We can define an "Action Unit" for a twitterbot as the combination of two actions—a request for information, a post to Twitter whose content is based on the result of the information request.

The "request for information" can be any of the following:

- Computing a value in your program (e.g., the current time of day)
- Accessing a value stored in a list (e.g., a list of famous quotes or song lyrics represented as strings in your program)
    - The access can be randomized.
- Retrieving information over the internet (e.g., looking up weather or other information through some public API)
- Searching through posts or users on Twitter.

The "post to Twitter" can be any of the following:

- A simple post with text.
- Text (or emojis) used to create simple images.
- A post with media (images, gifs – let's avoid any video or audio files).
- A post with text and media.
- A quote tweet (not implemented in simple_twit yet)
- A reply to another tweet with text and /or media
    - This option should be avoided; having bots interact directly with other users without getting that user's explicit permission first is prohibited by Twitter's rules on automated behavior.

Your bots should implement exactly 1 action unit. If your action unit contains a search of Twitter itself, keep the number of results requested for that search small.

Your action unit should at most be executed once every 6 hours.

If you want to loosen these constraints, you should apply for your own developer credentials at: https://developer.x.com

9. Your bot must adhere to Twitter's policies about acceptable behavior on their platform, especially their policies regarding automated behavior. Here is a summary:

Spam, bots, and automation
The use of the X API and developer products to create spam, or engage in any form of platform manipulation, is prohibited. You should review the X Rules on **platform manipulation and spam**, and ensure that your service does not, and does not enable people to, violate our policies.
Services that perform write actions, including posting Posts, following accounts, or sending Direct Messages, must follow the Automation Rules. In particular, you should:

- Always get explicit consent before sending people **automated replies or Direct Messages**
- Immediately respect requests to opt-out of being contacted by you
- Never perform bulk, aggressive, or spammy actions, including **bulk following**

- Never post identical or substantially similar content across **multiple accounts**

If you're operating an API-based bot account you must clearly indicate what the account is and who is responsible for it. You should never mislead or confuse people about whether your account is or is not a bot. A good way to do this is by including a statement that the account is a bot in the profile bio.

A fuller description can be found here:

https://help.twitter.com/en/rules-and-policies/twitter-automation

Submission Instructions:

1. Due Friday, December 13th, 2021 – Completed Twitterbot

   -You must turn in a description of your design for your Twitterbot. It should not be more than 1 page in length.

   -Your description must be typewritten and turned in on Brightspace as a PDF.

   -You must complete the following 4 exercises in you mytwitterbot.py file

   1. Retrieve and print out the 10 most recent tweets from your bot's home timeline.
   2. Retrieve and print out the 10 most recent tweets from another user.
   3. Post a single tweet to your bot's timeline.
   4. Post a single media tweet to your bot's timeline.

   -Complete the implementation of your twitterbot in the twitterbot() function inside mytwitterbot.py.

   -Submit to Brightspace:

   1. A PDF containing the description of your twitterbot.
   2. mytwitterbot.py file containing your implementation of the four exercises and your twitterbot.