Collecting Raw Tweets – User Guide

Setup Instructions

Note: This project requires Python 3.4 or later. On the school server, this is python3.

1. Navigate to /tweetanalyzer.

2. If you have not already, initialize and activate a virtual environment using Python 3's venv module.

   A. To avoid interfering with any package installations you may already have for your system-wide Python interpreter (and to resolve some permissions errors we encountered when attempting to install some libraries on the school server), we recommend installing our
   B. project's dependencies using a Python virtual environment. To do so, simply execute the following command in /tweetanalyzer:

   ```
   $ python3 -m venv venv
   ```

   (Note: If you are not using the school server, your Python 3 distribution may be under a different name like "python" or "python34".)

   This command will create a copy of your Python 3 distribution in a subdirectory called "venv". Once activated, installations will be made to this copy instead of your main distribution.

   C. Activate the virtual environment using the following command:

   Linux:

   ```
   $ source venv/bin/activate
   ```

   Windows:

   ```
   $ venv\Scripts\activate
   ```

   You will know activation is successful because the terminal prompt will now begin with the text "(venv)".

   ```
   (venv) $
   ```

   The virtual environment will remain activated until you close the console or enter the command "deactivate." If you accidentally deactivate the virtual environment, simply navigate back to /tweetanalyzer (if needed) and repeat step 3B.

3. Navigate to getTweets_pkg and install dependencies in the virtual environment.

```
(venv) $ cd getTweets_pkg
(venv) $ pip install -r requirements.txt
```

NOTE: We have noticed that the numpy and pandas libraries take several minutes to install on the school server. The installation may appear to hang, but it is still working! Installation seems to take much less time on our own computers, however.

4. When installation is complete, you may run the following scripts:

   A. ./getTweets_pkg/main.py

   ```
   (venv) $ python3 main.py
   ```

   The README for our getTweets program provides instructions for how to use the command line interface to retrieve Tweets based on a targeted search (e.g. keyword, hashtag, user). It is available on our project's GitHub page at: https://github.com/CS467-Lich/tweetanalyzer/blob/master/getTweets_pkg/README.md.

   We collect the following data from each Tweet:
   - "user": The username making the Tweet.
   - "date": The date the Tweet was made in a format such as "Sun Feb 03 23:12:08 +0000 2019".
   - "text": The body of the Tweet, including user mentions ("@[user]") or hashtags. Retweets begin with "RT @[originalAuthor]:".
   - "source": The application used to make the tweet (e.g. Twitter's web client or Twitter for iPhone).
   - "coordinates": The geographic coordinates for the Tweet. In practice, we have found that this is typically withheld.
   - "language": The ISO language code for the Tweet's language. We filter for English tweets only.
   - "hashtags": An array of objects with the attributes "text" (the hashtag stripped of the leading "#") and "indices" (an array of two integers in which the first integer is the index in the tweet text where the tweet begins and the second integer is the index where it ends).'

   B. ./getTweets_pkg/joinJSON.py

   joinJSON.py is a helper script that combines the data from all files in the directory specified at the beginning of the script (MY_DATA_FOLDER) into one and saves the combined data to the specified subfolder (MY_DATA_SUBFOLDER) within MY_DATA_FOLDER and filename (MY_FINAL_DATA). The data is output in both .json and .csv formats. We used this script to combine the results from multiple searches or random samplings into a single data file (i.e. one file per category).

   For example, given the following…

```
14  MY_DATA_FOLDER = "joinJSON/"
15  MY_DATA_SUBFOLDER = "mySubfolder"
16  MY_FINAL_DATA = 'CombinedData'
```

… the script would load all files from the directory /joinJSON and output the combined data to /joinJSON/mySubfolder/CombinedData.json and /joinJSON/mySubfolder/CombinedData.csv.

Please be aware that this script requires all input data files in /joinJSON to be in .json format. To experiment with joinJSON.py, copy several .json files created using /getTweets_pkg/main.py into the /joinJSON directory and execute the script:

```
$ python3 joinJSON.py
```