Twitter Analysis DB

opencircuits.com/Twitter_Analysis_DB

Contents

- 1 Goal
    - 1.1 Status
- 2 What/Why
- 3 Installation
    - 3.1 My Environment/Your Environment
    - 3.2 Download
    - 3.3 Run
- 4 Configure to Run
    - 4.1 Basic
    - 4.2 Running It
- 5 Notes on the Code
- 6 Design
- 7 Customizing/Extending
- 8 Additional Info

Goal

This document could use improvement, but the software largely works (see Status below) and I am probably done unless there is expressed interest in knowing more, or I add major new features. Look at history tab to see what is going on in the document. If you would rather look at at the application than read about it see Twitter Analysis DB GUI which you should check out at some point in any case.

Twitter Analysis DB is a Python open source, program and an accompanying database, running in a Graphical User Interface tool (and/or database creation tool) for the analysis of a body of tweets. Currently the program is in early alpha and its design goals are evolving at least as fast as the code is being written.

The point:

- Quickly select and view tweets (typically for a single person based on a number of criteria).
- Display concordance and simple linguistic analysis for a body of tweets.
- Provide a database of tweet and concordance data for analysis using tools external to the app.

This program has some supporting functionality:

- Load a body of tweets into a database (currently sqlite)
- Break the tweets down into a concordance.
- Have an accompanying db of English words (right now sourced from Kaggle)
- Provide a variety of queries against the database for possible enlightenment.

I will try to documented well enough so people can relatively easily extend and adapt the program. Or as alternative they can use other tools with the database like SQLiteStudio. It should be fairly easy to download and use even for those without a desire to dive into the code but, I assume some knowledge of Python, and a Python Environment to run it in. In Python 3.6 or so.

See the graphical user interface here (with screen shot): **Twitter Analysis DB GUI**.

This application is also part of a family of applications see the category below: Python Projects.

This is an article started by Russ Hensel, see **"http://www.opencircuits.com/index.php?title=Russ_hensel#About My Articles"** About My Articles for a bit of info.

Code will be at GitHub, see **[Code at GitHub]** See the GUI here at **Twitter Analysis DB GUI**

Status

- It works....typically produces output consistent with user input.
- Overall structure seems sound and extensible.
- Should be relatively easy to add additional queries, joins, columns, select criteria, without massive coding effort.

- But.... it is full of opportunities for enhancement. Right now my interests have shifted so I may not do much further work. Possibilits for improvement:
    - Clean up tweet in the database build stage. Pretty good not but still some odd "words" get through.
    - User interface is evolving but still not as user friendly as I would like.
    - Selects == (also know as Reports or Queries...) are more demos of what is possible than what is truly useful and informative, several are experiments in the technology of the application.
    - Sqllite still doing ok at 4 years of trump tweets and 300k of words.
    - No database optimizations yet.... I run on ram drive for speed. DB is about 40 MBytes with 4 years of trump tweets
    - Report formatting is basic, but workable. Nicest overall format for human readability is probably "html", best to pass to other applications is probably "csv", most responsive in time is "msg" -- sent to message area, often sub second response.
    - Not sure what area of work is most useful, have been driven lately by programming challenges need to focus for a bit on improving usefulness.
    - Still printing some unnecessary junk used in debugging, remove most... if output is needed sent to py_log, but whole logging parts of the application could use a careful review (not happening soon).

What technical knowledge should users have (and How):

Who	What	How
Person with little programming experience, no interest in Python. Looking for download and install.	Probably should use another program.	Not well suited to use this, but I may try to build an exe at some point.
Programming in some text based language, willing to install or has Python installed.	Run the basic program. Customize in fairly minor ways.	Edit the parameter file to configure the terminal.
Modest Python experience	Modify all over the place, save data to database	Program should be well documented in source, with some supplement in this wiki, or ask the author.

What/Why

I am curious about what has been said on twitter, when, and what vocabulary has been used.

So what are the features?

- Free open source
- Runs across OSs Linux (inc **Raspberry Pi**), Mac or Windows
- Python
- Multiple tables, preloaded with sample data.

- Database Interface (sqlite, could be modified for other relational databases)
- Parameter file for wide range of modifications of program behavior.
- Uses standard Python logging class.
- For more on the features see ???

Limits:

What would you like to see in the program or documentation? Email me.

Installation

This program is intended for those who at least occasionally develop in Python. I expect that they already run some things in Python and will just add this as an additional project. There is no install program you just download the files, place where you keep your projects and run. A bit more later in this section.

My Environment/Your Environment

The program has a better chance of running if your environment is not too much different from mine. The most important is that it is Python **3.6**

Before you begin to install you should know a bit about the environment that I have used to build, test and run the program. If your environment differs too much you may have trouble getting it to run.

I run Python mostly using the install that comes with Anaconda Spyder and often use the IDE it installs. This is not necessary, it is just an nice install that downloads a lot of stuff that technical folks find useful. I use conda and pip to add to this install and do not know offhand all that is in it. You can look at the include statements to get some idea of what you might need to add. Or you can just keep running it and add the packages it complains about.

Download

Code coming at GitHub, see [GitHub Repository] (it is Python and you can run directly from the source) Email me if you have issues (use this link User:Russ_hensel). You will get a zip file, unzip it and you should get:

```
.... whatever --|
                |
                |-- twitter_analysis_db --- all code required to run the application ( not sure if smart_terminal or python_smart_terminal or nothing is top level name, just put it in some well named place )
                                        |    some logs from my running of the code may or may not be present these will be deleted as they creep in, when you run the program you will
                                        |    get your own log files ... all typically named xxx.py_log
                                        |
                                        | --> input    ---  input files used to build the database.
                                        | --> output   ---  files produced by the database selects.
                                        | --> images   ---  image files, mostly screen shots, icons... or what ever, not important for the code.
                                        | --> wiki_etc ---  various files documenting program, including at least some of the material from this wiki
                                        |                   also some sample output files
                                        | --> help     ---  help files, documentation for various selects
```

Put them in your system making "....whatever" anything convenient for your Python installation (that is move the files to where you keep your Python source, not your installed module location).

Note that there may be a certain amount of left over, dead code, in the directories I am cleaning out bit by bit, someday it may be nice and neat. For now if you want to tinker look at the design info below first.

Run

Run it until it stops complaining about dependencies (in the console), after that (and perhaps even before) the GUI should come up. You are installed.

I have run the program on both Windows 10 and Rasperian on a RPi. It should work in most OS's. Let me know about issues.

Configure to Run

Basic

Basic configuration of parameters like database file name, ... is all done in a file called parameters.py. It seemed easier to simply use a Python text file instead of some other format like an ini file. Pretty much all the file does is set instance variables in itself. It is used by the program controller (tweet_app.TweetApp) to create an instance of Parameters and then the values can be used. Save the original (parameters.py, maybe I will include a backup maybe not) in case you mess it up too much.

I have made yet another pass to clean up and comment the code in parameters. The paramters file is its own primary documentation, so read it if you want to change parameters. Let me know if you have issues. You should understand some values are being phased out but may still have some implementation and some may be coming in and have little or no implementation. The comments should let you identify these situations.

Parameters starts out with some "meta" parameters. These are defined early in the creation of the objects and may effect other values. In any case you can always define a value twice, the last one always wins. The most important meta parameter is mode, you should not change it from self.mode = "working on this" unless you understand the implications or do not mind going on a ride.

=

Running It

When you run it it should open a windows a lot like the picture Twitter Analysis DB GUI. Errors may show up in your Python console or the log file (look in parameters.py for the name of the log file, typically self.pylogging_fn = "tweet_app.py_log"). The most likely errors will point to missing Python modules like pyperclip. You should install with pip (or conda if using Spyder). Let me know how it goes.

For info on general use of parmeters.py see Configuration Files For Python

> An early setting to configure in parameters.py to set the name of a text editor on your system. This is important for working with text files including the .txt output files and the parameters.py file. For mine this is one of:

```
self.ex_editor   =  r"leafpad"    # linux and pi

self.ex_editor   =  r"D:\apps\Notepad++\notepad++.exe" # for windows.
```

It is set up to auto switch between the two os to make copying the whole program back and forth between the windows and linux a bit easier.

Now when you run it the button <Edit Parms> should let you edit the parameters.py file. Edit it and save.

Hit the <Restart> button and in a flash (more or less) the program should restart with the new parameters, starting is fast because previously imported material does not need to be re-imported.

..... more here soon ?.....

Notes on the Code

This code is now an early work in progress. Until I loose interest in it it will probably improve. However, adding features is more part of the life of a programmer than polishing old features; making something better that seems good enough is not always the priority it should be. If you do not like it, mostly keep it to yourself unless it is accompanied by an offer to improve it. I do not need ideas, I need time. That said if you think you have a helpful comment contact me, my page will tell you how: User:Russ_hensel

Design

Description of the design, and a bit to help you figure out what the files do (for .py files also see the top of file).

see: Twitter Analysis Technical

Customizing/Extending

Simple customization may be done simply by changing the parameter file, for other stuff you will need to wade into the code. I try to name, factor, and comment well, but it is a work in progress.

Additional Info

Click on the category smart Twitter Analysis DB (and perhaps the others as well)