Twitter Analysis DB

opencircuits.com/Twitter_Analysis_DB

Contents

Goal

This document just started ... will probably take a week or so look at history tab to see what is going on

Twitter Analysis DB is a Python open source, program and an accompanying database (and/or database creation tool) for 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.

This program is designed to be:

- Load a body of tweets into a database (currently sqlite)
- Break the tweets down into a concordance.
- Have an accompanying db of English words (I need to find out how to provide within copyright guidelines)
- 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 more features below.?

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
Who should use this program 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, I do not plan to fix this.
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 soruce, 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 comming at GitHub, see [Code at GitHub] (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 --|
                |
                |-- tbd---------- --| -> 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
                                    |
                                    | -- 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
```

Put them in your system making "....whatever" anything convenient for your Python (that is move the files to where you keep your Python source).

Note that there may be a certain amount of left over, dead code, in the directory 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.

Some parts of the program think that mySql is available. It should run fine without it (there will be a message or two in the console), untill you try to use the database then it is not so gracious. Straight ahead use as a terminal does not use the database.

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.

Command Line Arguments

If you run with the command line parameters=paramaters_b then after the regular parameters file runs, then the system looks for parameters_b.py and uses that to override values that you might want to tweak (or completely redo. There are two examples in the directory follow the pattern in them and you should be fine).

This can be especially useful if you want to run two copies connected to different ports and possibly running in different modes. In this case it is also nice to change (its in parameters) the **icon** and **color** for each instance of the program. You can write or use the little bat file to start them (although this leaves a dos console hanging around) Command line arguments can also be placed in shortcuts. In either case they may take some tweaking to run in/from your file locations.

The above mostly applies to Windows, but the program run fine with Linux (including the Raspberry Pi) and I suppose the Mac. Of course the .bat file and shortcuts will not work, but similar facilities exist in the other OSs.

Running It

When you run it it should open a windows a lot like the picture Smart Terminal 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 = "smart_terminal.py_log"). The most likely errors will point to missing Python modules like pyserial. You should install with pip (or conda if using Spyder). Let me know how it goes.

Normally the terminal does not open the com port until you press the <Open> button. The parameters are displayed in the GUI if you do not like them (for example when you press the <open> button the port open status changes to "open failed", not the desired "open") you can shutdown, edit parameters.py and restart. There is a simpler way.

> First configure parameters.py to know the name of a text editor on your system. 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. In a flash the program should restart with the new parameters.

..... 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: Python Smart Terminal 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)

These are left over from another project, I may build similar pages for this or not