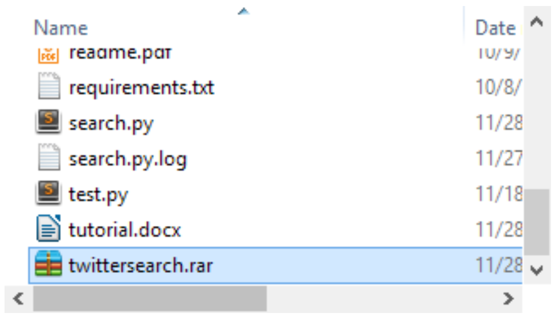
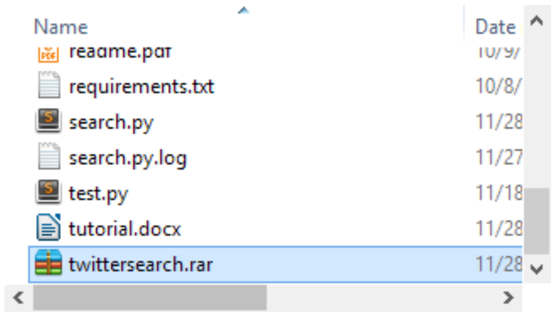
Install

1. Extract all files to a folder:



Name	Date
readme.pdf	10/9/
requirements.txt	10/8/
search.py	11/28
search.py.log	11/27
test.py	11/18
tutorial.docx	11/28
twittersearch.rar	11/28

2. Open 'must-install' directory and install essential libraries



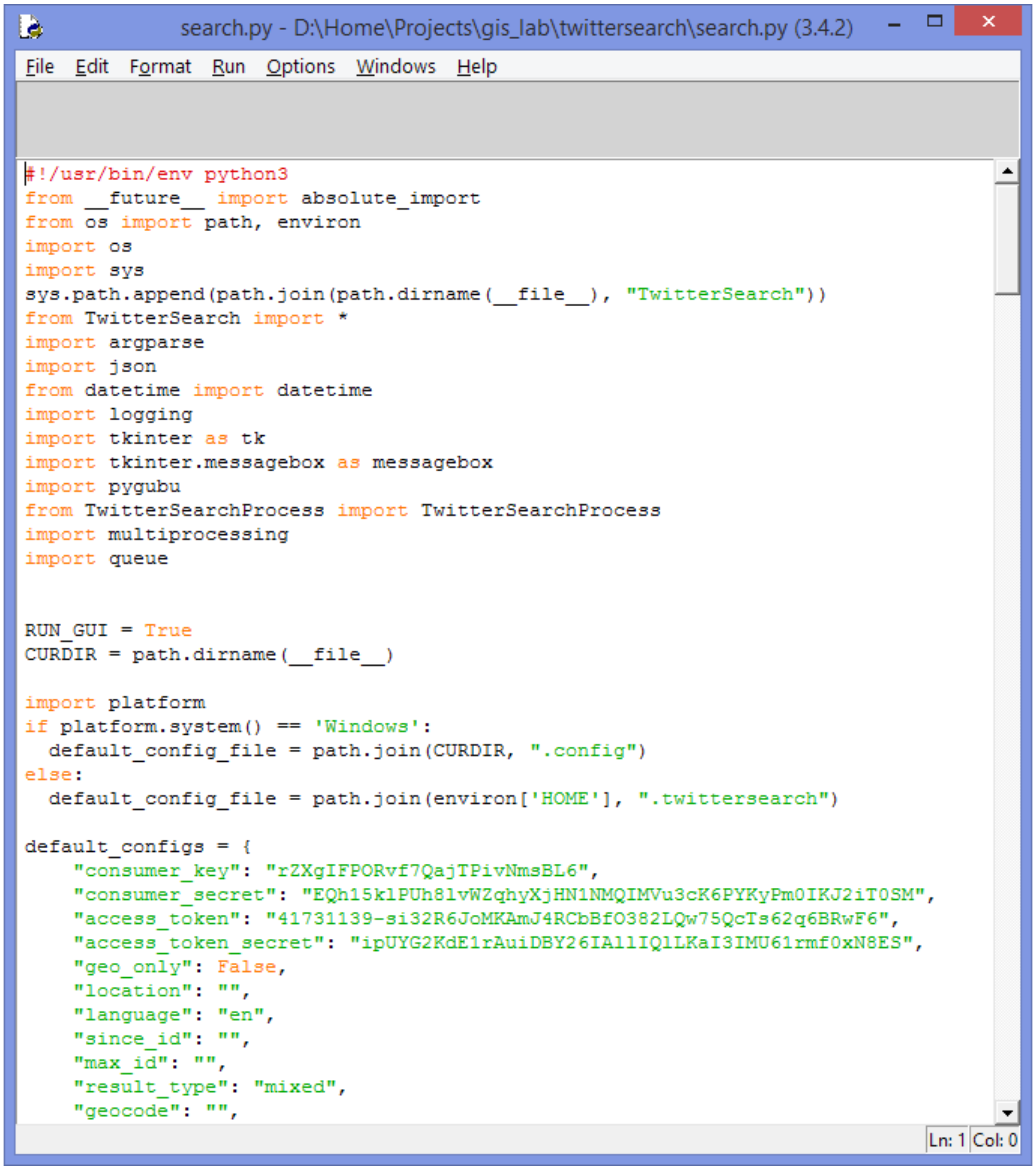
Name	Date
readme.pdf	10/9/
requirements.txt	10/8/
search.py	11/28
search.py.log	11/27
test.py	11/18
tutorial.docx	11/28
twittersearch.rar	11/28

Choose the correct version, note that the bit width must match your Python version. That is, if you have Python 3.4 64-bit installed, you have to install the lxml-3.4.1.win-amd64-py3.4.exe .

Python 2 is not supported due to the poor unicode support in Python 2 libraries. Please install Python 3.x.

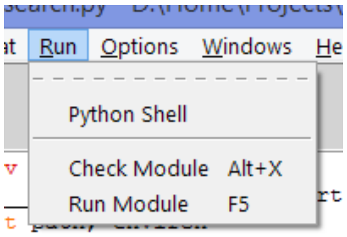
Usage

1. Load the 'search.py' to IDLE:

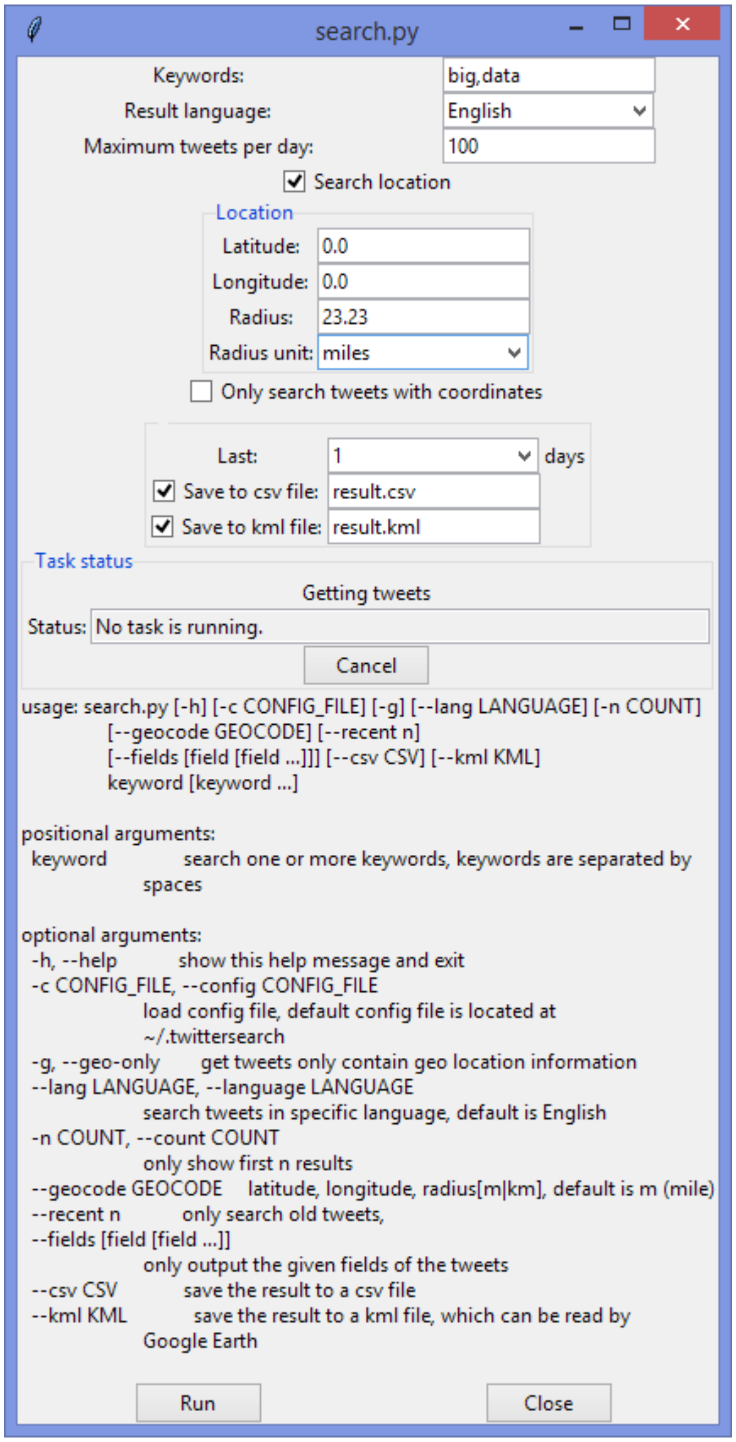

search.py - D:\Home\Projects\gis_lab\twittersearch\search.py (3.4.2)
File Edit Format Run Options Windows Help
Ln: 1 Col: 0

```python
#!/usr/bin/env python3
from __future__ import absolute_import
from os import path, environ
import os
import sys
sys.path.append(path.join(path.dirname(__file__), "TwitterSearch"))
from TwitterSearch import *
import argparse
import json
from datetime import datetime
import logging
import tkinter as tk
import tkinter.messagebox as messagebox
import pygubu
from TwitterSearchProcess import TwitterSearchProcess
import multiprocessing
import queue


RUN_GUI = True
CURDIR = path.dirname(__file__)

import platform
if platform.system() == 'Windows':
  default_config_file = path.join(CURDIR, ".config")
else:
  default_config_file = path.join(environ['HOME'], ".twittersearch")

default_configs = {
    "consumer_key": "rZXgIFPORvf7QajTPivNmsBL6",
    "consumer_secret": "EQh15klPUh8lvWZqhyXjHN1NMQIMVu3cK6PYKyPm0IKJ2iT0SM",
    "access_token": "41731139-si32R6JoMKAmJ4RCbBf0382LQw75QcTs62q6BRwF6",
    "access_token_secret": "ipUYG2KdE1rAuiDBY26IAl1IQlLKaI3IMU61rmf0xN8ES",
    "geo_only": False,
    "location": "",
    "language": "en",
    "since_id": "",
    "max_id": "",
    "result_type": "mixed",
    "geocode": "",
```

2. Click menu 'Run' -> 'Run Module' or press 'F5' on the keyboard to launch the script



Now the main GUI should show up, it may take a few seconds to load at the first time.



Keywords: big,data
Result language: English
Maximum tweets per day: 100
[✓] Search location

Location
Latitude: 0.0
Longitude: 0.0
Radius: 23.23
Radius unit: miles

[ ] Only search tweets with coordinates

Last: 1 days
[✓] Save to csv file: result.csv
[✓] Save to kml file: result.kml

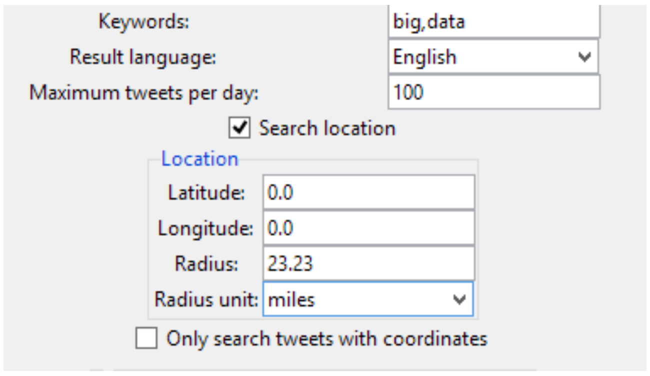
Task status
Getting tweets
Status: No task is running.

```
usage: search.py [-h] [-c CONFIG_FILE] [-g] [--lang LANGUAGE] [-n COUNT]
                 [--geocode GEOCODE] [--recent n]
                 [--fields [field [field ...]]] [--csv CSV] [--kml KML]
                 keyword [keyword ...]

positional arguments:
  keyword               search one or more keywords, keywords are separated by
                        spaces

optional arguments:
  -h, --help            show this help message and exit
  -c CONFIG_FILE, --config CONFIG_FILE
                        load config file, default config file is located at
                        ~/.twittersearch
  -g, --geo-only        get tweets only contain geo location information
  --lang LANGUAGE, --language LANGUAGE
                        search tweets in specific language, default is English
  -n COUNT, --count COUNT
                        only show first n results
  --geocode GEOCODE     latitude, longitude, radius[m|km], default is m (mile)
  --recent n            only search old tweets,
  --fields [field [field ...]]
                        only output the given fields of the tweets
  --csv CSV             save the result to a csv file
  --kml KML             save the result to a kml file, which can be read by
                        Google Earth
```

3. Config search settings

   There are tree main parts of the GUI:

   1. First is the configuration fields:



Keywords: big,data
Result language: English
Maximum tweets per day: 100
✓ Search location
Location
Latitude: 0.0
Longitude: 0.0
Radius: 23.23
Radius unit: miles
☐ Only search tweets with coordinates

Keywords is a comma separated list of words to search.

Language is the language of tweets will be in. Twitter will try to search the tweets in that language.

Maimum tweets per day is the maximum number of tweets will be retrieved and stored in the file of one single day, the script will move on to search the tweets sent next day when it reaches the count limit.

Search location when checked, the 'Location' section will become active, and you can set the location information there, twitter will search tweets sent from that location.

Latitude the latitude coordinate of the location, must be a floating point number.

Longitude the longitude coordinate of the location, must be a floating point number.

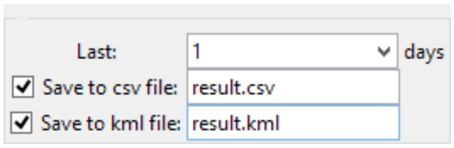
Radius the radius of the search area, must be an **integer**.

Radius unit the unit of the radius, can be "miles" or "kilometers".

GEO Only option will make the script only return tweets that contain geo location information.

Search old tweets will make the script only search tweets sent in the past('recent' mode), up to 7 days. If it's unchecked, the script will run in 'mixed' mode, that it, it mixes the old tweets with new tweets.

2. The second part is the option panel for 'recent mode', it's only available when 'Search old tweets' was checked.



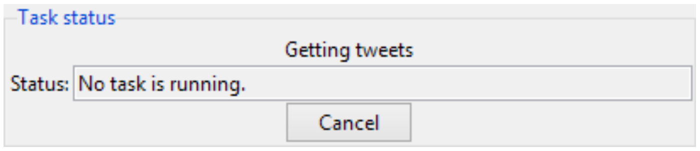
Last: 1 days
✓ Save to csv file: result.csv
✓ Save to kml file: result.kml

Last n days will tell the script only search tweets that were sent in last n days. You can select up to 7 days.

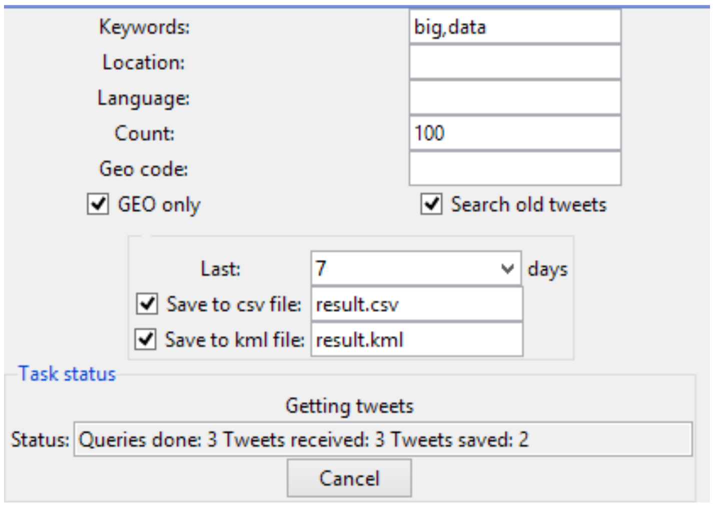
Save to csv file checkbox will make the script save the result in a csv file, the text field after that is the file name.

Save to kml file checkbox will make the script save the result in a KML file, the file contains a list of place markers, you can open the kml file in Google Earth and see each tweet's location.

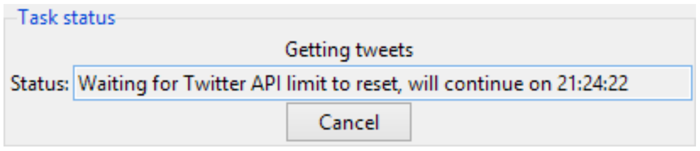
3. The third part is 'Task Status' panel, it will show the status when the search task is running.



Task status
Getting tweets
Status: No task is running.

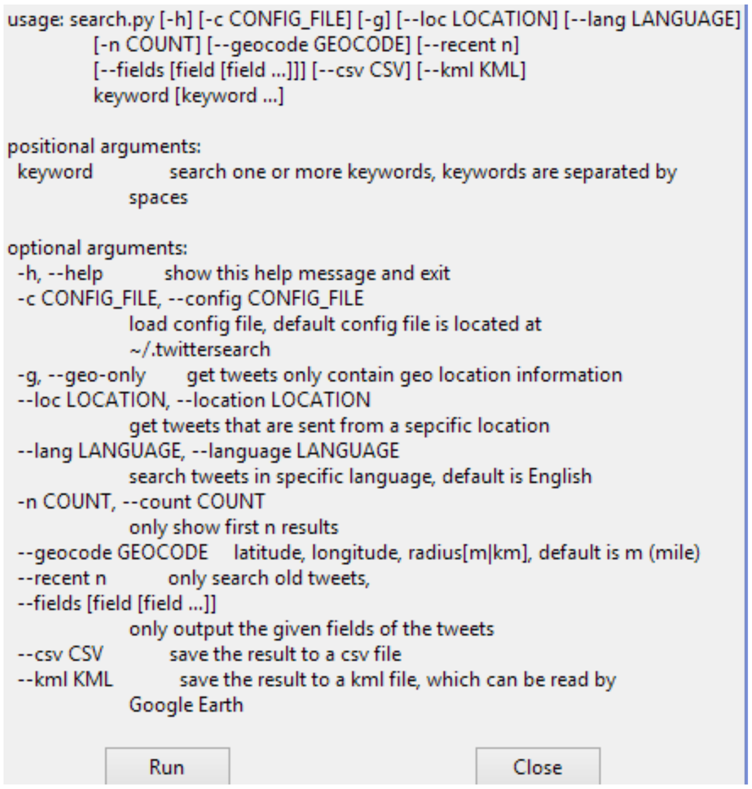

Keywords: big,data
Location:
Language:
Count: 100
Geo code:
☑ GEO only ☑ Search old tweets
Last: 7 days
☑ Save to csv file: result.csv
☑ Save to kml file: result.kml
Task status
Getting tweets
Status: Queries done: 3 Tweets received: 3 Tweets saved: 2

When the API limit is reached, the status will change:

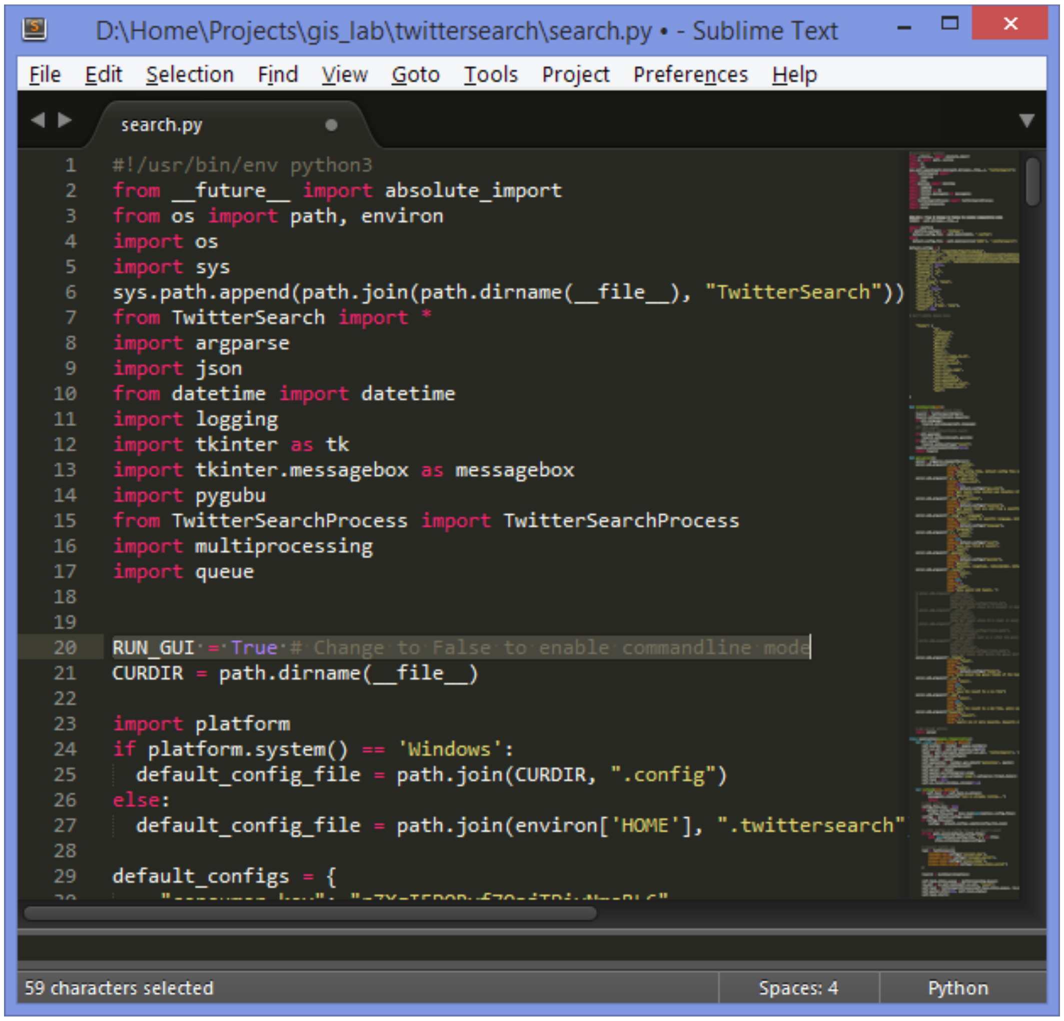

Task status
Getting tweets
Status: Waiting for Twitter API limit to reset, will continue on 21:24:22

Notice: The default API limit window is 15 minutes, and the default request limit of the search API is 180 per window, that means the script can only make up to 180 queries in 15 minutes, each query can hold upto 100 results(tweets), to get more tweets the script has to wait until next window, so there are roughly 18000 tweets retrieved in every 15 minutes, but note that not all of them will be saved to csv/kml if 'Geo Only' was checked.

4. The last part is the description and control buttons

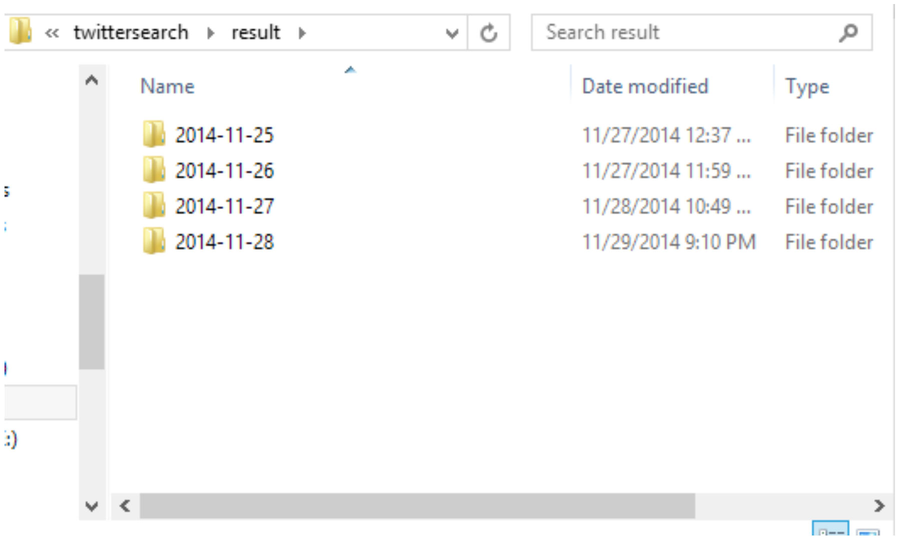

```
usage: search.py [-h] [-c CONFIG_FILE] [-g] [--loc LOCATION] [--lang LANGUAGE]
                 [-n COUNT] [--geocode GEOCODE] [--recent n]
                 [--fields [field [field ...]]] [--csv CSV] [--kml KML]
                 keyword [keyword ...]

positional arguments:
  keyword               search one or more keywords, keywords are separated by
                        spaces

optional arguments:
  -h, --help            show this help message and exit
  -c CONFIG_FILE, --config CONFIG_FILE
                        load config file, default config file is located at
                        ~/.twittersearch
  -g, --geo-only        get tweets only contain geo location information
  --loc LOCATION, --location LOCATION
                        get tweets that are sent from a sepcific location
  --lang LANGUAGE, --language LANGUAGE
                        search tweets in specific language, default is English
  -n COUNT, --count COUNT
                        only show first n results
  --geocode GEOCODE     latitude, longitude, radius[m|km], default is m (mile)
  --recent n            only search old tweets,
  --fields [field [field ...]]
                        only output the given fields of the tweets
  --csv CSV             save the result to a csv file
  --kml KML             save the result to a kml file, which can be read by
                        Google Earth
```

The description and arguments are for commandline use. In other words, `search.py` can run in commandline mode, to make it run in commandline, simply change the `RUN_GUI` option in the script to `False`.


D:\Home\Projects\gis_lab\twittersearch\search.py • - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
search.py
59 characters selected
Spaces: 4
Python

```python
#!/usr/bin/env python3
from __future__ import absolute_import
from os import path, environ
import os
import sys
sys.path.append(path.join(path.dirname(__file__), "TwitterSearch"))
from TwitterSearch import *
import argparse
import json
from datetime import datetime
import logging
import tkinter as tk
import tkinter.messagebox as messagebox
import pygubu
from TwitterSearchProcess import TwitterSearchProcess
import multiprocessing
import queue


RUN_GUI = True # Change to False to enable commandline mode
CURDIR = path.dirname(__file__)

import platform
if platform.system() == 'Windows':
    default_config_file = path.join(CURDIR, ".config")
else:
    default_config_file = path.join(environ['HOME'], ".twittersearch"

default_configs = {
```

When you click the 'Run' button, the search will begin, and the result will be saved to the 'result' folder under current folder:

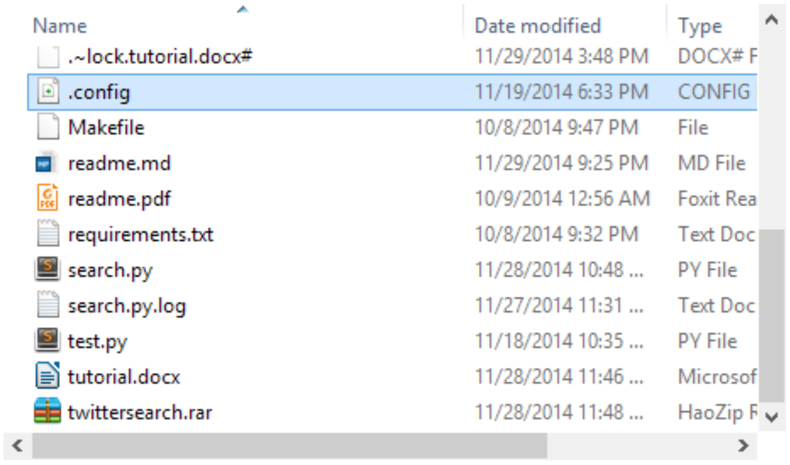


Name	Date modified	Type
2014-11-25	11/27/2014 12:37 ...	File folder
2014-11-26	11/27/2014 11:59 ...	File folder
2014-11-27	11/28/2014 10:49 ...	File folder
2014-11-28	11/29/2014 9:10 PM	File folder

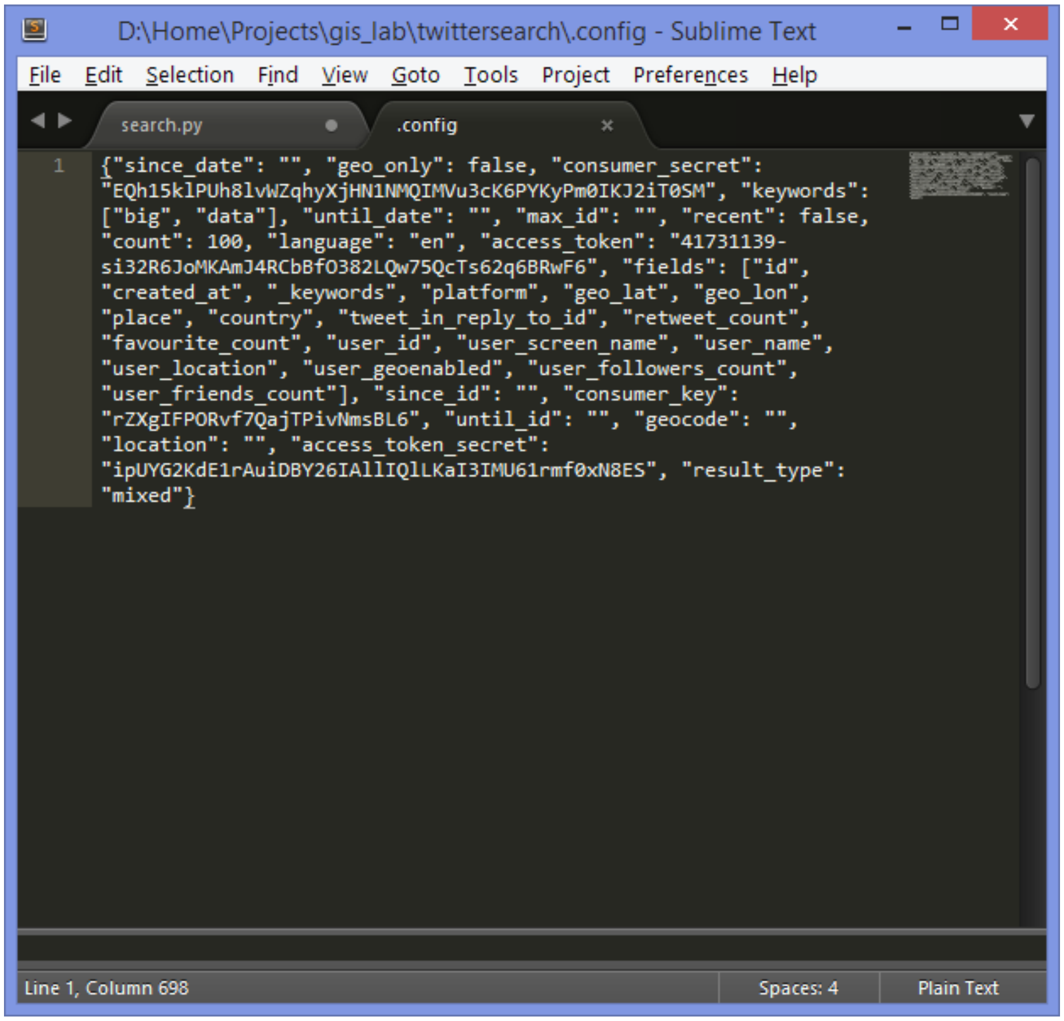
Each subfolder in 'result' folder contains the tweets of that day.

Configs

The script will create a config file under the same folder of the script. To be able to use Twitter's API, you will have to register an app in Twitter's developer portal, and obtain your `consumer_key`, `consumer_secret`, `access_token` and `access_token_secret` information.



Name	Date modified	Type
.~lock.tutorial.docx#	11/29/2014 3:48 PM	DOCX# F
.config	11/19/2014 6:33 PM	CONFIG
Makefile	10/8/2014 9:47 PM	File
readme.md	11/29/2014 9:25 PM	MD File
readme.pdf	10/9/2014 12:56 AM	Foxit Rea
requirements.txt	10/8/2014 9:32 PM	Text Doc
search.py	11/28/2014 10:48 ...	PY File
search.py.log	11/27/2014 11:31 ...	Text Doc
test.py	11/18/2014 10:35 ...	PY File
tutorial.docx	11/28/2014 11:46 ...	Microsof
twittersearch.rar	11/28/2014 11:48 ...	HaoZip F


D:\Home\Projects\gis_lab\twittersearch\.config - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
search.py
.config
Line 1, Column 698
Spaces: 4
Plain Text

```
{"since_date": "", "geo_only": false, "consumer_secret":
"EQh15klPUh8lvWZqhyXjHN1NMQIMVu3cK6PYKyPm0IKJ2iT0SM", "keywords":
["big", "data"], "until_date": "", "max_id": "", "recent": false,
"count": 100, "language": "en", "access_token": "41731139-
si32R6JoMKAmJ4RCbBfO382LQw75QcTs62q6BRwF6", "fields": ["id",
"created_at", "_keywords", "platform", "geo_lat", "geo_lon",
"place", "country", "tweet_in_reply_to_id", "retweet_count",
"favourite_count", "user_id", "user_screen_name", "user_name",
"user_location", "user_geoenabled", "user_followers_count",
"user_friends_count"], "since_id": "", "consumer_key":
"rZXgIFPORvf7QajTPivNmsBL6", "until_id": "", "geocode": "",
"location": "", "access_token_secret":
"ipUYG2KdE1rAuiDBY26IAllIQlLKaI3IMU61rmf0xN8ES", "result_type":
"mixed"}
```

Usually you don't need to change these settings.