Twitter Crawler

Rashid Goshtasbi and Brendan Cheng

SID: 861056442 and 861055750

Hello there! T-Crawler is an open source project for Linux to download tweets over twitter that has geolocation enabled.

Collaboration Details

Description of contributions of each team member

Rashid Goshtasbi

- Figuring out an API to use
- Researching instructions to download additional libraries to incorporate with API and product feature
- Researching on implementation on API
- Creating Twitter account and signing up for Twitter Developer Access for special keys required by API
- Figuring our necessary functions to use to run library functions
- **Coding**: Configuring twitter API streaming functions:

  ```
  l = listener()
  auth = OAuthHandler(ckey, csecret)
  auth.set_access_token(atoken, asecret)

  while True:
      try:
      twitterStream = Stream(auth, l)
      twitterStream.filter(locations=[-180,-90,180,90], languages=["en"])
  except:
      continue
  ```

- **Coding**: Contributing to converting streaming tweets to JSON format to parse:

```
with open("tweets"+str(numFile)+".txt", 'a') as output:
        if(os.path.getsize("tweets"+str(numFile)+".txt") < 10000000):

            #LOADS ALL OF THE TWEET DATA INTO VARIABLE "TWEET"
            tweet = json.loads(data)
```

- **Coding**: Contributed to parsing tweets for url links:

```
#IF STATEMENTS LOOKS FOR TWEETS THAT HAVE TITLE
#IF IT DOESNT HAVE A URL, IT DOESNT PARSE IT
if (text_tweet.find('http') != -1):
    url_text = text_tweet.find(str1)
    link_text = text_tweet[url_text:url_text+23]
    #print link_text

    content = urllib2.urlopen(link_text).read()
    soup = BeautifulSoup(content, "html.parser")
    tweet[u'linktitle']= soup.title.string
else:
    pass
```

- **Coding**: Contributed to retrieving title of url link in tweet's text

- Implementing of "**try:**" and "**except:**" protocols:

```
...
try:
...
except:
...
```

- Figuring out how to output each JSON string onto a NEWLINE in our output file

- Figured out how to output each file into any computer using:

```
#path for files to save
path = os.getcwd()
```

Brendan Cheng

- Researching instructions to download additional libraries to incorporate with API and product feature
- Researching on implementation on API

- **Coding**: Configuring twitter API streaming functions:

```
l = listener()
auth = OAuthHandler(ckey, csecret)
auth.set_access_token(atoken, asecret)

while True:
    try:
    twitterStream = Stream(auth, l)
    twitterStream.filter(locations=[-180,-90,180,90], languages=["en"])
except:
    continue
```

- **Coding**: Contributing to converting streaming tweets to JSON format to parse:

```
with open("tweets"+str(numFile)+".txt", 'a') as output:
        if(os.path.getsize("tweets"+str(numFile)+".txt") < 10000000):

            #LOADS ALL OF THE TWEET DATA INTO VARIABLE "TWEET"
            tweet = json.loads(data)
```

- **Coding**: Contributed to parsing tweets for url links:

```
#IF STATEMENTS LOOKS FOR TWEETS THAT HAVE TITLE
#IF IT DOESNT HAVE A URL, IT DOESNT PARSE IT
if (text_tweet.find('http') != -1):
    url_text = text_tweet.find(str1)
    link_text = text_tweet[url_text:url_text+23]
    #print link_text

    content = urllib2.urlopen(link_text).read()
    soup = BeautifulSoup(content, "html.parser")
    tweet[u'linktitle']= soup.title.string
else:
    pass
```

- **Coding**: Contributed to retrieving title of url link in tweet's text
- Report on limitations, data structures
- Figuring out how to output each JSON string onto a NEWLINE in our output file

Overview of System

Two functions: main(), listener()

- Main function:
    - Handles Twitter permissions
    - Utilizes listener() to gather and store data

- Listener function:
    - Gathers data, Tweets, from Twitter and loads into a JSON object
    - Parses Tweets into different fields of data from JSON object
    - Stores links and titles from parsed data into output file

- Libraries:
    - time
    - os
    - sys
    - random
    - requests
    - re
    - urllib
    - urlib2
    - HTMLParser
    - Stream
    - OAuthHandler
    - StreamListener
    - BeautifulSoup
    - Additional Libraries:
    - tweepy
    - tweepy.streaming
    - bs4
    - Data Structures:
    - json
    - string

Limitations

- Need to install additional libraries to use program
- If a Tweet has two links, T-Crawler can only crawl the first link
- Only gathers data from Tweets in English
- Gathers data from Tweets from all location, not specific

- Exit through a keyboard interruption from the terminal may take several tries due to parent and child processes running

Instructions

May vary depending on Operating System, examples below were used on Linux OSX

NOTE: Program will download files to the current directory you have cloned the Github to or where you have the file in

1. Install PIP: In terminal, type in: **sudo easy_install pip**
2. Install tweepy API: In terminal, type: **sudo pip install tweepy**
3. Install BeautifulSoup: In terminal, type in: **sudo pip install beautifulsoup4**
4. How to obtain T-Crawler using GitHub or if you have tcrawler.py file already:
    1. If you already have tcrawler.py program, skip ahead to step 5
    2. clone program from https://github.com/rgosh001/tweets.git by typing **git clone https://github.com/rgosh001/tweets.git**
    3. Once clone is complete, using the command line in your terminal, type **cd tweets** to enter resposity.

5. While in the directory with tcrawler.py in it, to run T-Crawler, type into the command line **python tStream.py** NOTE: Program will also print out tweets downloaded into terminal window and filenames associate by "**tweets#.txt**" will start appearing with downloaded tweets
6. The downloaded files will max at 10MB each and will continue downloading 10MB files till the total size of 5GB is reached (download time may vary on internet connection as tested prior to release)

PHOTOS SHOWING SYSTEM IN ACTIOB BELOW