Parsing Real-time Tweets

Ramanujam Parthasarathy
Monetate

@ramanujam
http://github.com/ramanujam

Why?

- Freely available public intelligence
- News reaches twitter super fast and spreads
- 250 million tweets per day (Oct ‘11)
- Track keywords, sentiment analysis, data mining
- Because you can!

Twitter Streaming API

https://stream.twitter.com/1/statuses/filter.json

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Error 401 Unauthorized</title>
</head>
<body>
<h2>HTTP ERROR: 401</h2>
<p>Problem accessing '/1/statuses/filter.json'. Reason:
<pre>    Unauthorized</pre>
```

Twitter Streaming API

```
curl https://stream.twitter.com/1/
statuses/filter.json?
track=SEARCH_PARAMETER -
uUSERNAME:PASSWORD
```

Twitter Python Libraries

- Tweepy
- Python Oauth2
- Python Twitter
- Twython
- Twitty Twister
- Tweetstream

Use a library that has support for Oauth

Get your auth keys

pugtalk

Details | Settings | OAuth tool | @Anywhere domains | Reset keys | Delete

foofoofoofoofoo

http://ramanuj.me

Organization

Information about the organization or company associated with your application. This information is optional.

Organization	None
Organization website	None

OAuth settings

Your application's OAuth settings. Keep the "Consumer secret" a secret. This key should never be human-readable in your application.

Access level	Read-only About the application permission model
Consumer key	`U10WJutrcjEoparnrKdC6w`
Consumer secret	`e5CIbG7rTYAAGE3u8lFmUH8KcoJcfGyGs7PLJtCVfQc`

Basic keyword/hashtag tracking

Set the keywords to track

```
streamer = tweepy.Stream(auth=auth, listener=StreamListener(), timeout=3000000000 )
setTerms = ['happy', 'delighted', 'merry', 'cheerful']
```

Print the status

```
class StreamListener(tweepy.StreamListener):
    status_wrapper = TextWrapper(width=60, initial_indent='    ', subsequent_indent='    ')

    def on_status(self, status):
        try:
            print '\n %s  %s  via %s\n' % (status.author.screen_name, status.created_at, status.source)
            print status.text
```

File: tracker.py

Save to MongoDB

```
connection = pymongo.Connection("localhost", 27017)
db = connection.election

words = ["ron paul", "obama", "romney"]
with tweetstream.FilterStream(username, password, track=words) as stream:
    for tweet in stream:
        db.tweets.save(tweet)
```

Pymongo (http://api.mongodb.org/python/2.2/index.html) is a recommended way to work with MongoDB from Python

Plot some data

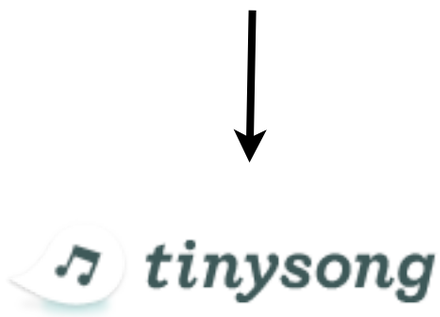
Pie chart of the devices used by people who tweet about Obama and Romney

```
import collections
import webbrowser
from pygooglechart import PieChart3D

def draw_obama_chart():
    with open('obama.txt', 'r') as f:
        data_o = collections.Counter([line.split(',')[1].strip() for line in f.readlines()]).most_common()

    o_data = [app[1] for app in data_o][0:10]
    o_labels = [app[0] for app in data_o][0:10]

    obamaChart = PieChart3D(400, 100)
    obamaChart.add_data(o_data)
    obamaChart.set_pie_labels(o_labels)
    webbrowser.open_new_tab(obamaChart.get_url())
```

File: plotapps.py

Play some music

#pugmusic



http:\/\/tinysong.com\/6OAB

Play some music

```
def on_status(self, status):
    try:
        tweetText = status.text
        truncatedText = tweetText.rsplit('#pugmusic')[0]
        tinysongurl = 'http://tinysong.com/b/'+truncatedText+'?format=json&key='+tinykey
        print tinysongurl
        tinyresponse = requests.get(tinysongurl).text
        jsonResponse = json.loads(tinyresponse)
        songurl = jsonResponse["Url"]
        print songurl
        webbrowser.open_new_tab(songurl)
```

File: playmusic.py

Enqueue more songs

- Grooveshark javascript api
- **addSongsByID**
  - window.Grooveshark.addSongsByID(13766524)

Notify

Twilio API to send a text or make a call based on an event trigger (eg: > 10 statuses in a 10 second window)

setTerms = ['$AAPL', 'down', '530']
setTerms = [‘aws’, ‘down’]
setTerms = [‘phillies’, ‘win’]

```
    def on_status(self, status):
        try:
            message = client.sms.messages.create(to="+17077262685", from_="+14155992671",
                                                 body=status.text)

            call = client.calls.create(to="17077262685", from_="+14155992671",
                                       url="http://foo.com/call.xml")
```

File: sendtext.py

Thank You!

http://github.com/ramanujam/pugtalk