End-to-End Twitter Stream Processing Application

This application captures and stores all text contained in tweets while the listener is running. Two applications are available to analyze total traffic, data on specific words, or data on occurrence ranges.

Description of the Architecture

Built on an Amazon EC2 instance, Tweepy connects with the Twitter API to feed an Apache Storm architecture with data. (3) Streamparse spouts process incoming data, passing their output to (3) Streamparse bolts where tweets are filtered, split into individual words, and grouped. Output is passed to (2) count-bolts to maintain a total count for each word. Within the (2) count-bolts a postgres database is updated with the occurrence of each word, where the counts are stored after the listener is closed. Two serving layer applications written in python query this database using PsycoPG libraries to provide analytics.

Dependencies

- Amazon EC2 AMI:
  - AMI Name: UCB MIDS W205 EX2-FULL
  - AMI ID: ami-d4dd4ec3 tweepy
- PsycoPG
  - psycopg2==2.6.2
- Tweepy
- Twitter Account with an active Application running

Directory and file structure

```
w205f17_exercise2
    extweetwordcount
        src
            bolts
                parse.py
                wordcount.py
            spouts
                tweets.py
        topologies
            tweetwordcount.clj
        virtualenvs
            wordcount.txt
        config.json
        fabfile.py
        project.clj
        tasks.py
    README.md
    finalresults.py
    histogram.py
```

Usage

Capture and store tweet data:

W205~ /extweetwordcount$: sparse run

From the extweetwordcount/ directory, enter “sparse run” to start the listener and begin collecting data. Processed words and their counts will flash down the screen. After enough words have been processed, enter Ctrl+C to stop the listener.

W205~ /extweetwordcount$: Ctrl+C

Analyze tweet data using finalresults.py

From the exercise2/ directory, enter either:

finalresults.py

- Displays all captured words and their counts

finalresults.py <word>

- Displays count for provided <word>

Analyze tweet data using histogram.py

From the exercise2/ directory, enter:

histogram.py <k1>,<k2>

- Displays all captured words with counts greater than or equal to k1, and less than or equal to k2