Storytelling with Streaming Data

HW2-Presidential Election Tweets Rate

YU WANG UNI: yw2783 - February 23, 2016

Purpose

This project consumes streaming data from Twitter API and parse tweets for presidential election. In other words, we parse tweets whose content is about the election and identify different parties: Hillary Clinton or Donald Trump and intend to tell who is more popular and hot on the twitter topic. An event under the scenario would be a new tweet about the topic which contains keyword 'Hillary ' or 'Trump'. We will record the time when it generated in database and calculating the moving average rate of two category. By comparing the rate, we would know the current more popular candidate and the system would send an alert once one rate has surpassed the other.

By observation, tweets about Hillary is usually generated faster than Trump. So assume we are CIOs who run Hillary's election and would be really interested when Trump's popularity on twitter has surpassed Hillary. Therefore the alert would be triggered, represented by both image and message on the webpage. (Please refer to data visualization section)

In summary:

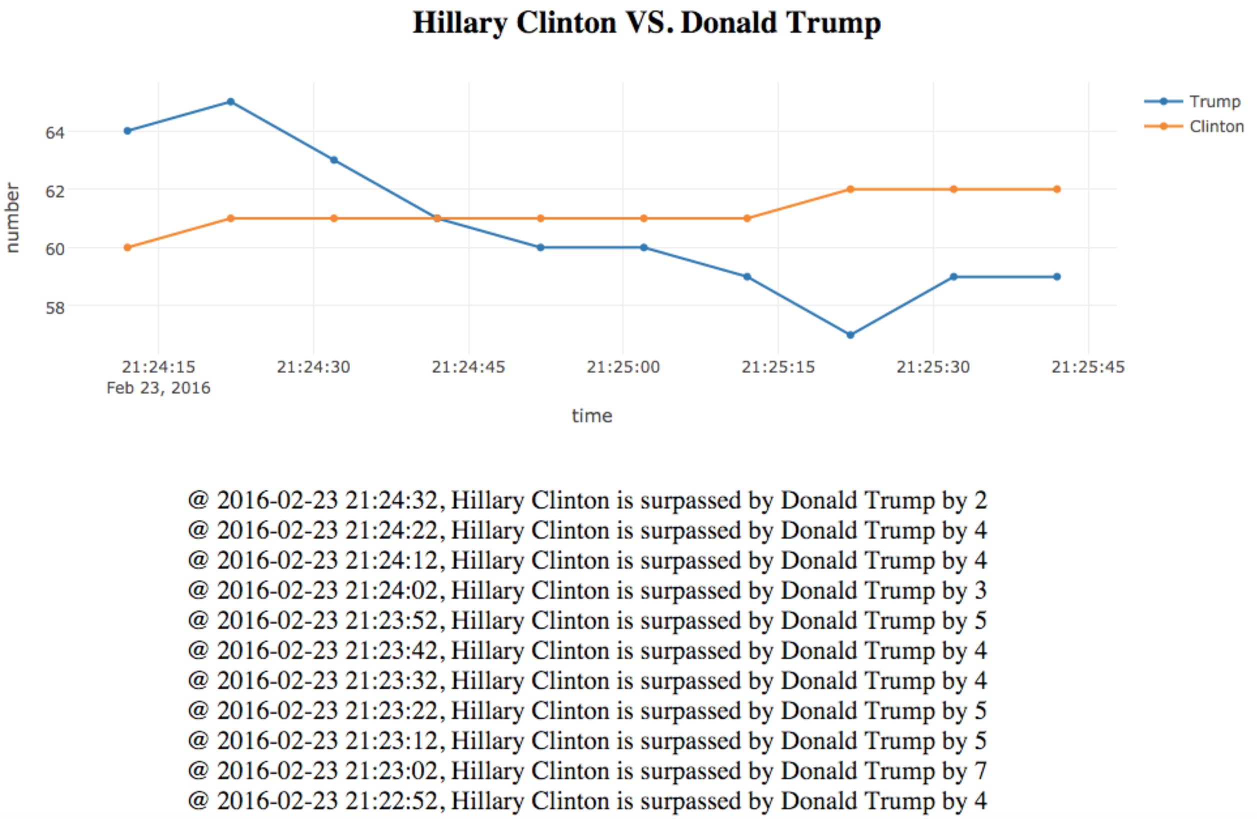
- Rate we are tracking — Tweets generating rate for two party of presidential election. Each rate represent which side is currently more popular and 'hot' on twitter topic
- Alerting system — The system will send an alert on the webpage if the rates relationship has changed during the time. i.e. If ' Trump ' is currently more popular than 'Hillary', a notice would be generated on the page illustrating that now people's attention has switched to Trump.

This document will demonstrate how the rates will be presented (with demo graph), how the streaming data is processed in each step, what is the structure of source code, and how to run the application. The following sections will elaborate the information respectively.

Data Visualization

The project will present current hot candidate on tweets.

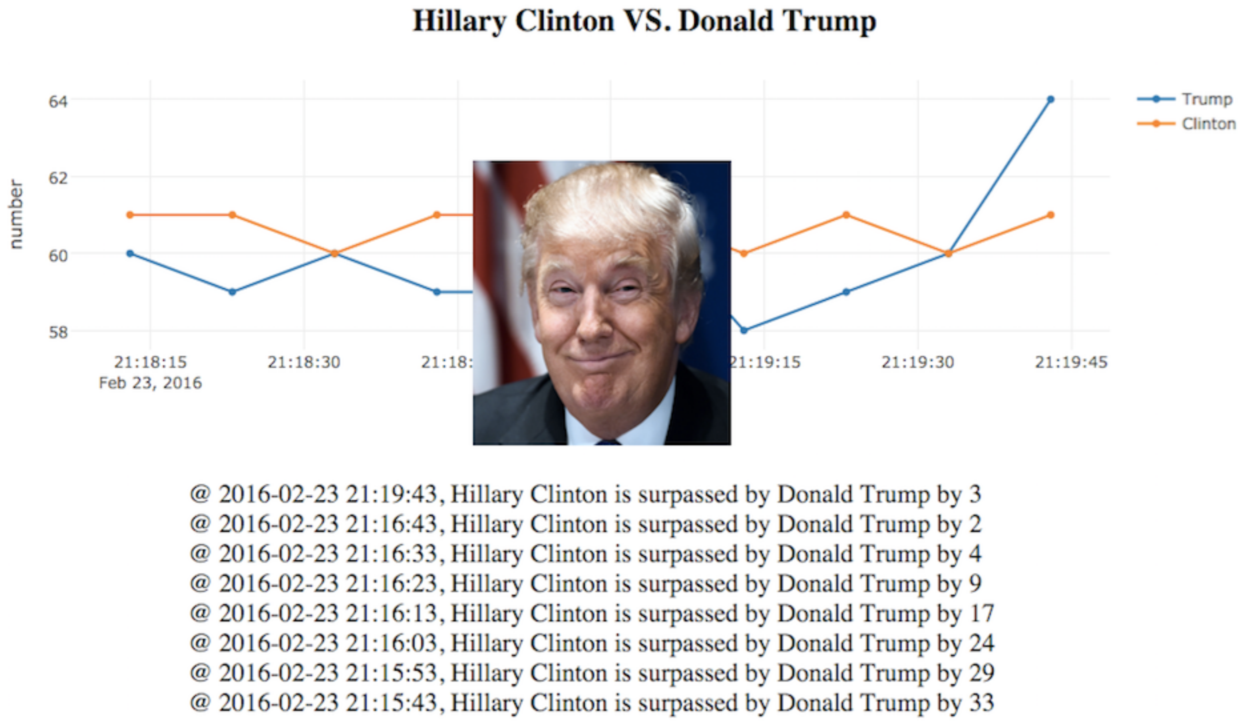
1. The rate of tweets generated in the time window
2. Compare the two rates and tell which party is more 'hot' and discussed more frequently on twitter.



```
@ 2016-02-23 21:24:32, Hillary Clinton is surpassed by Donald Trump by 2
@ 2016-02-23 21:24:22, Hillary Clinton is surpassed by Donald Trump by 4
@ 2016-02-23 21:24:12, Hillary Clinton is surpassed by Donald Trump by 4
@ 2016-02-23 21:24:02, Hillary Clinton is surpassed by Donald Trump by 3
@ 2016-02-23 21:23:52, Hillary Clinton is surpassed by Donald Trump by 5
@ 2016-02-23 21:23:42, Hillary Clinton is surpassed by Donald Trump by 4
@ 2016-02-23 21:23:32, Hillary Clinton is surpassed by Donald Trump by 4
@ 2016-02-23 21:23:22, Hillary Clinton is surpassed by Donald Trump by 5
@ 2016-02-23 21:23:12, Hillary Clinton is surpassed by Donald Trump by 5
@ 2016-02-23 21:23:02, Hillary Clinton is surpassed by Donald Trump by 7
@ 2016-02-23 21:22:52, Hillary Clinton is surpassed by Donald Trump by 4
```

fig 1. Rate Graph

The figure above demonstrates real time tweets about “Hillary Clinton” and ”Donald Trumps”. The graph show the rate trend in each time slot. Once there is a crosspoint on the graph, meaning that one party has surpassed the other and becomes more popular in twitter topic. An alert would be presented on the webpage upon the status changes(figure 2)



```
@ 2016-02-23 21:19:43, Hillary Clinton is surpassed by Donald Trump by 3
@ 2016-02-23 21:16:43, Hillary Clinton is surpassed by Donald Trump by 2
@ 2016-02-23 21:16:33, Hillary Clinton is surpassed by Donald Trump by 4
@ 2016-02-23 21:16:23, Hillary Clinton is surpassed by Donald Trump by 9
@ 2016-02-23 21:16:13, Hillary Clinton is surpassed by Donald Trump by 17
@ 2016-02-23 21:16:03, Hillary Clinton is surpassed by Donald Trump by 24
@ 2016-02-23 21:15:53, Hillary Clinton is surpassed by Donald Trump by 29
@ 2016-02-23 21:15:43, Hillary Clinton is surpassed by Donald Trump by 33
```

fig.2 Alert Scenario

How to Run

1. Run Redis database server

    #redis-server

2. Run the main.py python script

    #python main.py

3. Run the shell script

    # . run.sh

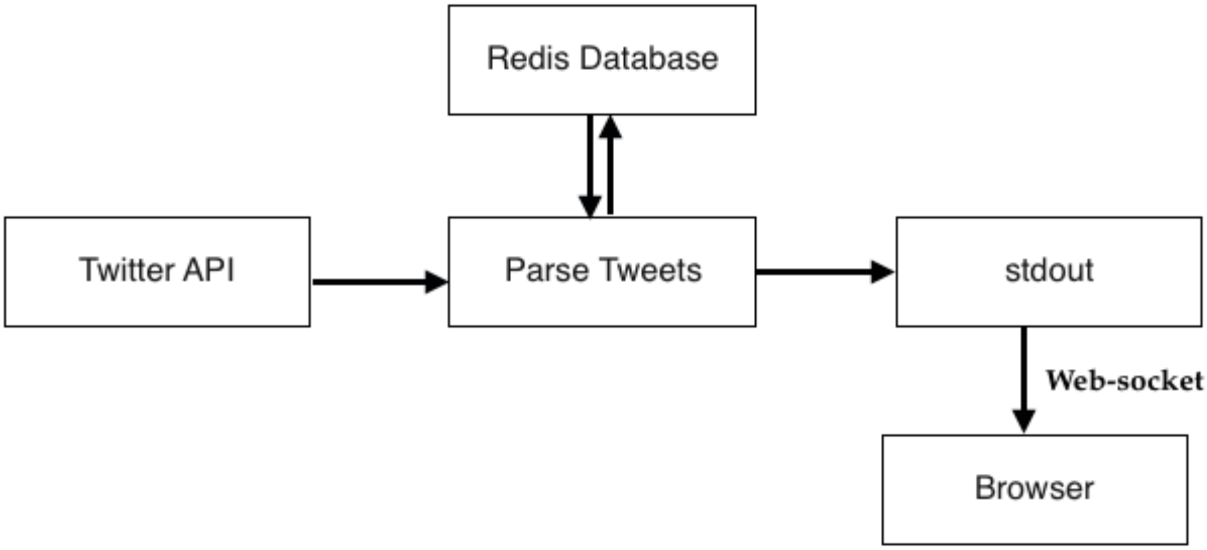
2. CD to the current project folder and build the python server :

    # python -m SimpleHTTPServer 8888

3. Open the webpage in browser by link:

    http://localhost:8888/

Processing Flow



Source Code

- main.py
  - Fetch the twitter streaming data
  - Filter tweets about “Hillary Clinton” and ”Donald Trumps”
  - Send twitter data to analyze_tweet.py
- analyze_tweet.py
  - Category tweets into the two groups
  - Store data in Redis database for rate calculating
- compute_avg.py
  - Reads data in Redis
  - Calculate rates for two parties and send data to websocket
- script.js
  - Listens the specified port and add data points on the graph
  - Send alert if data trigger the pre-defined condition
- run.sh
  - Build the web socket connection, push stream data into the specific port
- index.html
  - Web front-end page to demonstrate the result and graph

Note:The functionality of source code is also explained in the comments of each file.