Twitter Visualization Map

LARP745 Yujun Jiang

Introduction

Twitter, an online social network that allows users to upload short text messages—tweets—of up to 140 characters. This restriction encourages users to construct focused, timely updates.

Tweets basically offer two “layers” of information:
-----The obvious direct information within the text of the Tweet itself;
-----Tweets’ metadata:
It is not directly perceived,which is the the large number of additionalinformation like user data, retweet count, hashtags, etc. This metadata can be leveraged to experience data from Twitter in a lot of exciting new ways.

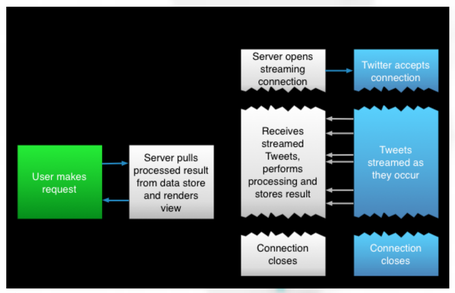
This project studies ways to visualize twitter in real time so that we could explore the spatial and temporal pattern of people’s reaction towards an event or a specific keyword.

Application Function Structure

Part 1:Data Collection

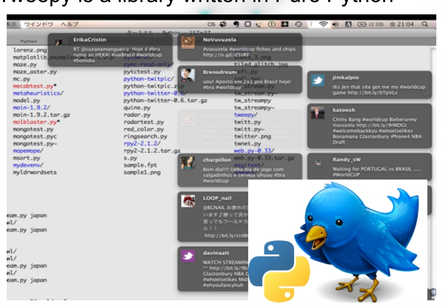
Twitter Streaming API:

The set of streaming APIs offered by Twitter give developers low latency access to Twitter's global stream of Tweet data.



tweepy:

Tweepy is a library written in Pure Python



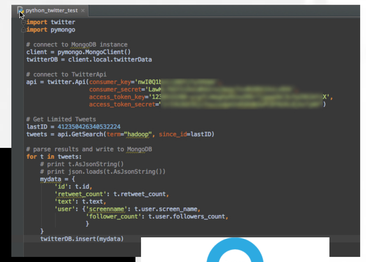
Part 2:Data Storage

MongoDB:

MongoDB is a cross-platform document-oriented database system

Flask:

a microframework for Python



Redis:

Redis is an open source, BSD licensed, advanced key-value store. It is often referred to as a data structure server since keys can contain strings, hashes, lists, sets and sorted sets.

Part 3:Web Map Implementation

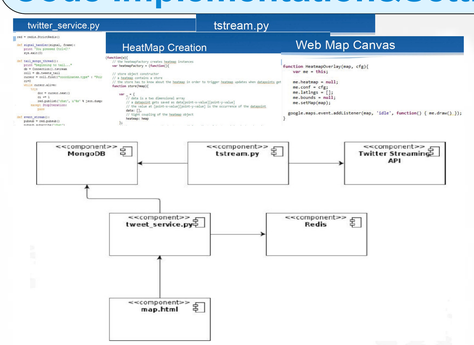
jQuery Eventsource:

It gives developers the power of the EventSource API across browsers.

Google Maps API&heatmap-gmaps.js:

It provides the access and script to enable the browser to display a google map canvas with basic functions like zooming in and out.

Code Implementation&Setting



The code part of this project is written in Python and Javascript, and it consists of mainly three components:

tstream.py:

A small service based on tweepy that implements a StreamListener which inserts incoming data in a MongoDB capped collection, which can also set filter terms.

tweet_service.py:

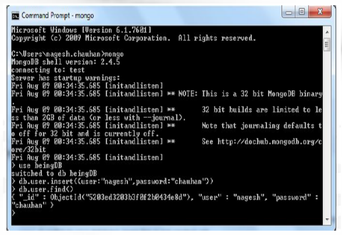
A Flask based web app which gets new data from MongoDB and makes use of the publish subscribe pattern.Incoming Tweets are published to a redis channel for which there is also a listener that returns a “text/event-stream” “Content-Type” header for connecting clients.

map.html:

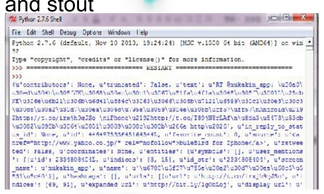
A few lines of HTML and JavaScript which bring up a Google Maps canvas and a listener for server-sent events. When new data, basically consisting of Lat/Lon tuples, arrives the new point is added to a heat map overlay based on heatmap-gmaps.js in real-time.

Start to run:

Start Mongodb&Redis



Start the tstream.py so incoming tweets are logged and stout



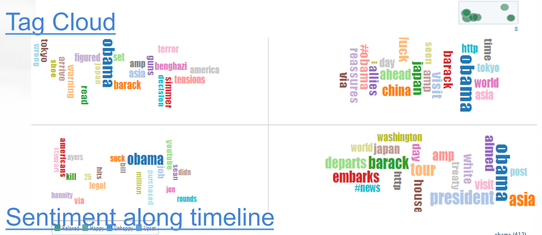
Start tweetservice.py and connect the browser

Analysis Sample

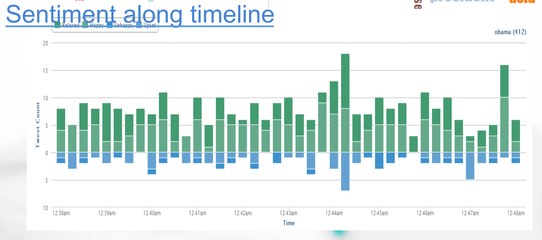
Refine data retrieved to derive useful knowledge

Clustering topics concerning keywords

Tag Cloud



Sentiment along timeline



Discussion

In essence, this project will highlight anything with high variance that changes. I am able to get information from interactions instead of demographics, meaning that we are able to see how people actually feel about certain topics instead of stereotyping based on race and gender

My next step is to insert more analysis functions into the web part instead of doing them separately