**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 large number of additional information 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.

**Part 2: Data Storage**

MongoDB:

MongoDB is a cross-platform document-oriented database system

Flask:

a microframework for Python

**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**

Start to run:

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

Start the tweet_service.py and connect the browser

Start to run:

Start Mongodb&Redis

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.