

Final Project - Twitter Adapter Report

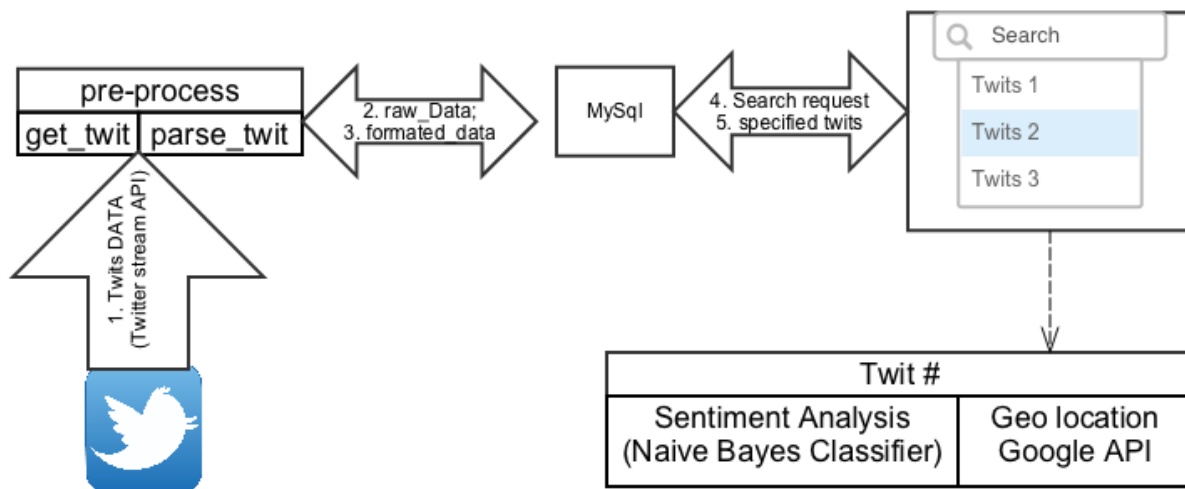
Team Member: Deep Patel, Hongye Gong, Yukun Liang
github link: <https://github.com/hongyegong/CS6083>

Project Definition:

The project definition is to work on Twitter API get tweets from the twitter server for the given keyword and store it to My-sql database on cloud. The Scope of the project is to explore the APIs and find out a way to get GEO location (latitude, longitude) from the tweet and also do sentimental analysis on the tweet that is it negative, positive or neutral. Moreover ,User should be able to view, who tweeted and what was tweeted in temporal order.

Introduction:

This is a twitter adapter project with both server side in php and mysql database and front-side user interface by html/css/JS. Basically, we retrieve bunch of data from twitter server in real-time down to our local database and then parse them into our local normalized database. Then we render all data onto the webpage through html/css/js and then analyze sentimental and location information of every single tweet by the php_nlp bases classifier and google geo coding API.



workflow

Here I present the whole procedure in 3 parts regarding to the functionality and mechanism.

I. Data processor (Backend):

1.1 Get_twits.php is ran, it gets realtime twits into the data flow.

1.2 Parse_twits.php runs in order to save the twits into local mysql DB form the flow and normalize them into formatted tables.

II. User Interface

Index.php logs the user in and shows the records in the DB. When new twits arrive realtime, it notifies to have them shown.

III. Extended function

For each twit shown in the twits list, there are two functional extentions implemented as buttons optional:

1.1 Sentiment Analysis: this NLP process gives out an integrated sentiment result in general, reflexing the attitude implied in the twit. The result is represented in points, more directly as a facial expression beside the twit.

1.2 Geo location: this sends location keywords (previously fetched from former NLP process) to Google geo API and gets the returning long/lat, then use the pair as parameter to Google map API to further show the location in a embedded google map.

Detailed Explanation:

There are Three parts of this project.

The first part is for retrieve data from twitter server to local database. The database is populated in two steps: getting the tweets, and parsing them into multiple tables. It is important to separate these operations, because tweets may be sent by the Twitter API at a very fast rate. If each tweet is parsed and inserted into multiple tables as it is received, the code and database may not be able to keep up with the data flow, and tweets will be lost. Our solution is to store the tweets as they are received in a simple cache table without doing any parsing. A separate process does the parsing and storage into separate tables.

The first step of collecting tweets is done by **get_tweets.php**, which is run as a continuous background process. When a new tweet is received the Twitter streaming API, get_tweets.php uses db_lib.php to insert it into the json_cache table. The connection with the Twitter streaming API is maintained by the Phirehose library.

The Twitter streaming API returns data in JSON format. To make the collection process as fast as possible, the entire JSON payload for a single tweet is saved to the database as a single string without any parsing.

A separate background process is run for **parse_tweets.php**, which gets the JSON data for each tweet from the json_cache table, parses it into it's component parts, and inserts them into

separate table for use by the other modules in the Twitter adapter. Once again, **db_lib.php** is used to manage the MySQL code.

The rest of the modules of this project then are able to rely on this database of tweets and supporting data without having any direct contact with the Twitter API. Once you install the Twitter database server code, then can build my own Twitter apps that are assured of a real-time source of Twitter data.

The second part is for displaying which is a webpage the users directly interact with. Let's follow the flow of tweet data from web server to browser using the sample Web page included with this code, `index.php`:

When **index.php** is opened by a web browser, it uses `require_once()` to call **twitter_display.php** in the plugin directory which loads `tweet_list_template.txt`, which defines the HTML structure of the list of tweets. This template has macros that show where to include all the tweet data.

When the macro for the list of recent tweets is reached, `get_tweet_list.php` is called with `require_once()`. **Get_tweet_list.php** extracts the most recent tweets from the tweets table in the MySQL database, and loops through them. For each set of tweet data it uses `tweet_template.txt` to assemble an HTML version of the tweet. The text of each tweet is passed to the `linkify()` function. The `linkify()` function in **display_lib.php** converts each entity (@mentions, tags, and URLs) into formatted hyperlinks.

When `get_tweet_list.php` has a complete set of tweets in HTML format, it returns it to `twitter_display.php`, and the list appears in the Web page. After the page loads, **site.js** starts and sets up parts of the interface, including a More Tweets button at the bottom of the tweet list and a count of new tweets link at the top of the list. It also establishes a refresh interval for calling the Web server with Ajax to get the count of new tweets.

Every time the refresh interval is reached, `site.js` calls `get_new_tweet_count.php` with Ajax, gets the count of new tweets, and displays them in the new tweets count. If the user clicks the More Tweets button, `site.js` requests a set of older tweets from `get_tweet_list.php`. These are returned as HTML, and appended to the end of the tweet list. All of the interactions with the Twitter database server are done through the `db_lib.php` script in the db directory.

After we retrieve the data from the twitter server with tweeter streaming API and then render the content onto our webpage. The next is to do the data analysis in both **sentiment analysis** and the **geo-location analysis**.

We use the PHPInsight API which uses Naive Bayes trainer and classier to gather all the words in positive, negate or neutral and then classify the single sentence based on the probability of being positive or positive. This process including language prepress and classifying is done by invoke the API which is provided by `php_nlp_tools`.

As far as the geolocation part, we first get the dataset of all cities over the world which contains about 2 or 3 millions records and store them into our database for later detecting if the particular word in a tweet is a location information or not. For the query efficiency we need to create the index for the cityname column in that table we just created since there is too much data there

and It's necessary to create such a index for that column. Then we just look up every word in one sentence in database. Suppose we have 100 words in one tweet(which is not alway the case) and we have 20 tweets showing on the page. Then we most have 2000 words in one page and the time for each select query is approximately 0.0001 seconds so it won't take above 1 seconds to execute these operations which process all words on the page at the same time. After we get he location words then simply merge them together and then input the location information into google geo API. Then we will get the latitude and longitude and invoke the google API again, then we can finally show the tweet location on the google map. That's basically the workflow of the whole project.

Following is what every file specifically dose:

- Twitter Database Server Source Code Files:

Phirehose Library - Phirehose library for capturing tweets from the Twitter streaming API.

config.php - General configuration options for the this Twitter adapter.

db_config.php - Database configuration options for the Twitter database server.

db_lib.php - Database library used by the entire Twitter adapter.

db_test.php - Simple test script for the Twitter database server.

get_tweets.php - Gather tweets in real-time using the Twitter streaming API.

monitor_tweets.php - Run as cron job that reports errors by email if the tweet collection fails.

parse_tweets.php - Parse tweets into a normalized database schema.

- Twitter Display Source Code Files:

lib - php_Insight library for sentiment analysis using Naive Bayes classifier algorithm.

NlpTools - Another tools for natural language processing library used for tokenizing sentence.

default.css - CSS style sheet for the Twitter display plugin.

get_new_tweet_count.php - Return the count of new tweets as text/HTML.

get_tweet_list.php - Return the most recent tweets as HTML.

index.html - Example of a Web page using the Twitter display plugin.

twitter_display.php - Main module of the Twitter display plugin.

display_lib.php - Functions used to display tweets: linkify() and twitter_time().

site.js - Use Ajax to refresh the new tweet count and load more tweets.

tweet_list_template.txt - Text template for formatting a complete list of tweets on a Web page.

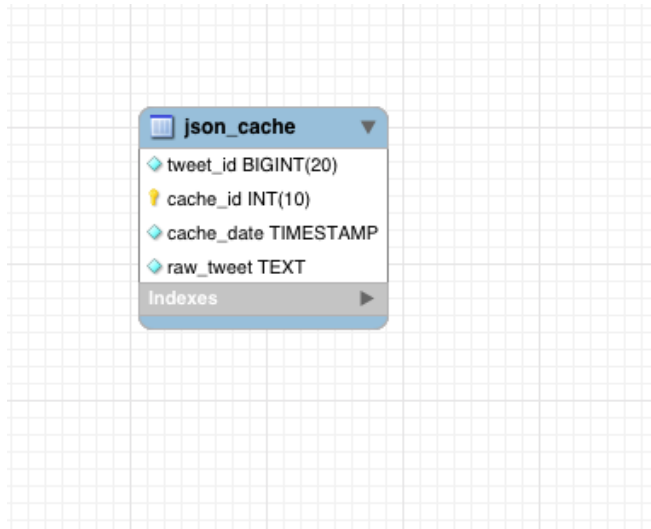
tweet_template.txt - Text template for formatting a single tweet.

twitter_display_config.php - Configuration options for the Twitter display plugin.

Database tables:

Tweets table:

For Internal use to process retrieving faster from twitter it is saved in following table in mysql. So this table is feed to parser to create actual normalized tables for our own system dataset.

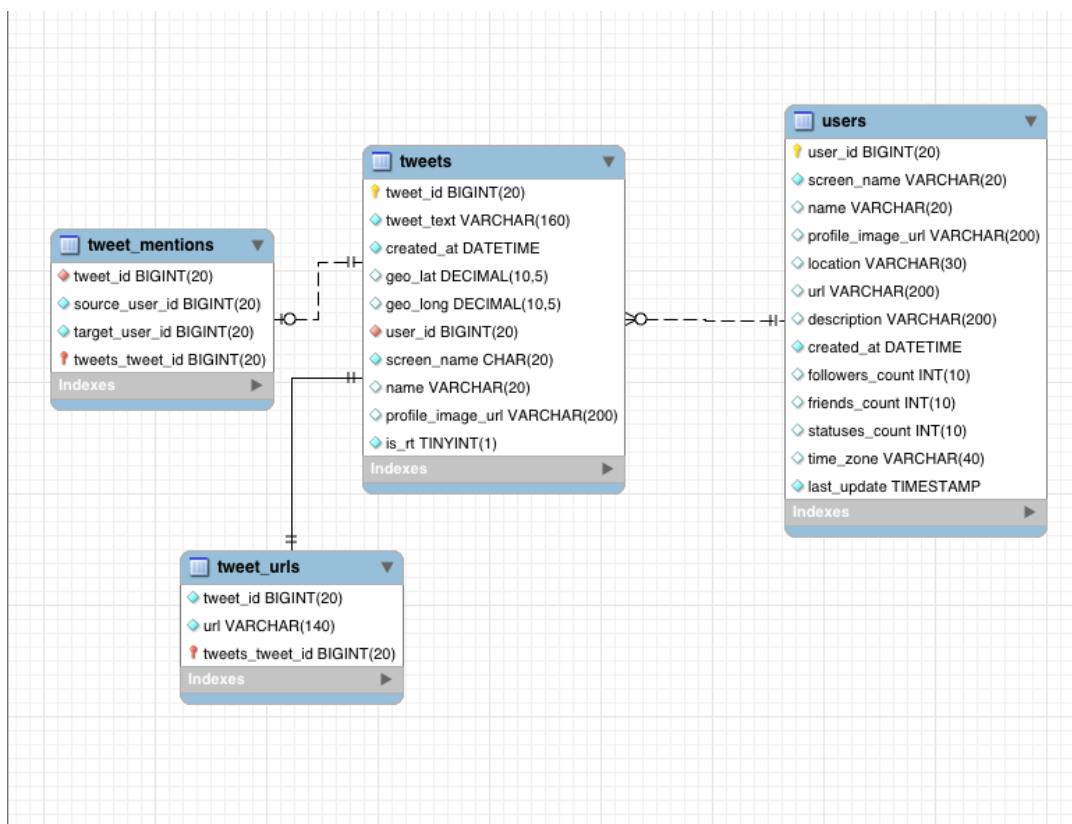


We create following tables from the Json_cath. The relations are basic explained in the class.

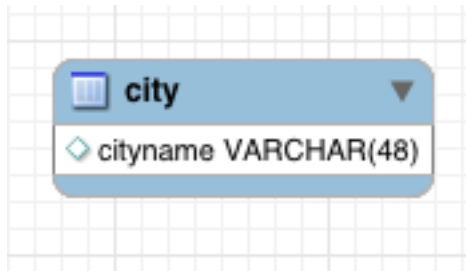
Users : It contains user information and it is connected to tweet table with tweet_id.

Tweets : it contains information related to tweets. which is connected to tweet_mentions and tweet_urls

As we are providing search funtion in our system we created index on tweet_text in tweet table to make our search function faster.



Moreover we have saperate table indexed which is used in our Geo coiding API preprocessing and make it faster.



References:


1. IBM Education initiative for MYSQL cloud
2. PhpInsight - Sentiment Analysis in PHP
3. Google Maps Geocoding API

We really want to thank IBM for providing infrastructure and Prof Raman Kanan forguiding us throughout the project.

Project Sample input/out snapshot:


Homepage index.php:

Twitter display adapter




2fantabulush Mariiizziii♥
SentimentAnalyze Geolocation

I may or may not be excited for "Pitch Perfect 2". I also may or may not wan to see "Hot Pursuit".
2 days ago
Map shows approximate location.



NorthCapitolDC North Capitol Main
SentimentAnalyze Geolocation

regram grssrtsgourmet Springy sugar cookies!!! Perfect for a springtime day and Mother's Day!!! @... <https://t.co/Jq7MbxGgzj>
2 days ago
Map shows approximate location.



Writer_HeidiG Queen_Bitch_Carly
SentimentAnalyze Geolocation

RT @CanProvelt: Florida Man Dies In Police Custody After Cops

Get tweets of specific topics from tweet server in real-time:

870 new tweets available. [Refresh](#) to see them.



Jusara__ Sarah

SentimentAnalyze

Geolocation

RT [@vinit_mittal](#): Sad but true.. <http://t.co/Pj3EjXrz4r>

less than a minute ago

Map shows approximate location.



c_alayna Layna

SentimentAnalyze

Geolocation

I work myself up so much before a speech and then I literally do perfect

less than a minute ago

Map shows approximate location.



cellabrations hannah

Asynchronously Get more tweets from database



BraderzClayts Bradley Clayton

SentimentAnalyze

Geolocation

RT @stillblazingtho: Overthinking is the biggest cause of our unhappiness. Keep yourself occupied. Keep your mind off things that don't hel...

about a minute ago

Map shows approximate location.



Antiindustry AntiIndustry.com™

SentimentAnalyze

Geolocation

Someone tell @S_C_ to start a distribution co. For positive hiphop its already been proven, the only rap music thats selling #JustFacts

about a minute ago

Map shows approximate location.

60 tweets displayed - View More

Search whatever topics you want:

search



DimitryJacobs Dimitry Jacobs

SentimentAnalyze

Geolocation

RT @OmniFocus: OmniFocus 2.5.2 for iOS, featuring Apple Watch and crash fixes, has been approved and should be available shortly: <https://t...>

less than a minute ago

Map shows approximate location.



LordJackTaylor Jack Taylor

SentimentAnalyze

Geolocation

@SantanderCycles are you going to make an Apple Watch app? That would make it much easier...

less than a minute ago

Map shows approximate location.



HelloHeartApp Hello Heart

SentimentAnalyze

Geolocation

@BRGLiving Excellent! We also think that the #AppleWatch is going to be a game changer, here is why

Geolocation analysis of a single tweet:



Tigerbudy Cassie Snyder

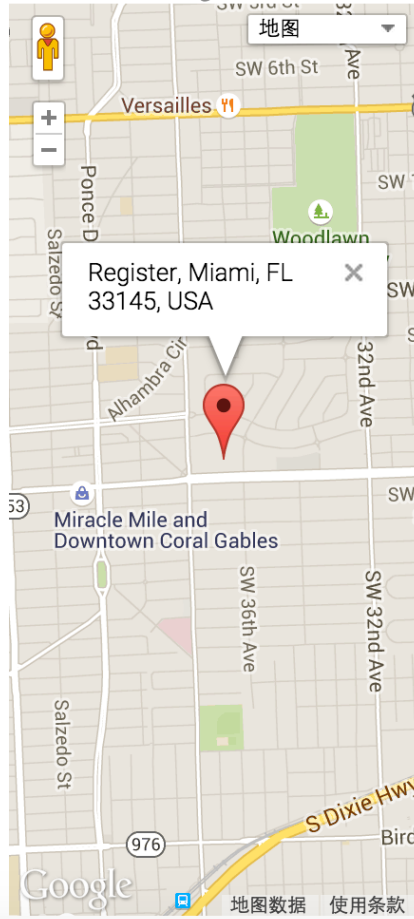
SentimentAnalyze

Geolocation

Please help me support Best Buddies Florida. Visit my page on the Bowling for Buddies website and register or...

<http://t.co/6Gu2grOYD8>

less than a minute ago



latitude: 25.7506651, longitude: -80.2529918

Sentiment analysis of a single tweet:



Tigerbudy Cassie Snyder

SentimentAnalyze

Geolocation

Please help me support Best Buddies Florida. Visit my page on the Bowling for Buddies website and register or...

<http://t.co/6Gu2grOYD8>

about a minute ago

Map shows approximate location.



kcin1122g Nicholas S. Gonzalez

SentimentAnalyze

Geolocation

RT @ontarioeda: Top 10 Reasons to Visit Ontario, California at @ICSC_RECon. 1) There is incredibly plush carpet at the booth.

☺ <http://t.co...>

about a minute ago

Map shows approximate location.



DOLL_BABIIE90 China Charisse♥