Janine Heiser

Info257

Dec 27, 2013

**Tubes Trends- Trending the Tubes**

**Description of Database**

A current online trend is that websites and social media sites display “trending” data about users who are using these the site or social network; its possible to go online and see what users across the wolrd are talking and thinking about. Website “trending” data typically can be described as a high number of user searches or mentions of particular person, place, thing, or idea on a given search engine or social network. Most recently, on August 30, 2013, Facebook released a new feature called, “Trending” that appeared in the homepage news feed of its website. This new feature, appears to mimic Twitter’s “trends box”, which highlights popular hashtag and topics on the twitter platform.[[1]](#footnote-2) The concept of a keeping track of user searches and user mentions of particular people, places, things, or ideas was started over 5 years ago by Google in August 2008, which eventually evolved into Google Trend. Although widely used websites have “trending” data, there is no website or entity on the internet that is aggregating this data across multiple websites[[2]](#footnote-3) For my database, I created a database that stores trending” data from Google Hot Trends, Twitter, Instagram and Youtube. Trends do not to not exist in a vacuum; the internet and mass media help move ideas and products to many countries and geographical places. Taken alone, individual trends could come and go and might not seem significant, , however, if one could map trends onto a map by geographical areas, larger patterns world patterns (human behavior and otherwise) and regional patterns might emerge where if one was watching for them.

**Purpose of Database**

My database about internet “trending” data from major web platforms and social networks hopes to the following purposes:

* to create an database model of world-trends that give gives a way to both see trends occur in real time (instantaneous) and think about trends in the long-term (over a course of a week, month, year, etc)
* map that trending data to the geographical cordinates of latitude and longitude how the data pieces geographically relate to each other;
* provide a way for users to compare and contrast what's “trending” on different search platforms and social networks to see if there are common themes
* show the duration of time that items/topics/people/ideas/places/stories are popular/trending on the internet
* to illuminate a potential “zeitgeist” that may or may not exist on the internet or in the world.

**Sources of Data**

For this project, my ultimate ambition was to visualize the data; I had hoped to map the trends to places on geo-cordinates. Having this end goal in mind, I focused my energy on obtaining trending data from sources that provided it for at least serveral countries. To obtain the data, I gathered it from a web API platform or I scraped it off the website directly. Below lists the APIs and websites from where I gathered data from:

**Twitter →** [**http://www.twitter.com**](http://www.twitter.com/) **→ social networking platform**

* **API resource, GET trends/available**
* Returns the locations that Twitter has trending topic information for.
* The location is returned in a woeid (yahoo where on earth id)
* When I did this project, the API returned at total of 415 locations.
* EX of API resource, GET <https://api.twitter.com/1.1/trends/available.json>
* [http://dev.twitter.com/docs/api/1.1/get/trends/available](https://dev.twitter.com/docs/api/1.1/get/trends/available)
* **API resource, GET trends/place**
* Each API call returns the top 10 trending topics for a a specific yahoo WOEID that trending data is avaible for.
* trends/place data is refreshed by twitter every 5 minutes, but users can only make 15 calls to this API in a 15 minute period
* Since this API is rate limited, users have to authenticate when they make API calls.
* **Required parameters**:
  + WOEID (The Yahoo! Where On Earth ID of a location)
* EX of API resource call: GET https://api.twitter.com/1.1/trends/place.json?id=1
* [http://dev.twitter.com/docs/api/1.1/get/trends/place](https://dev.twitter.com/docs/api/1.1/get/trends/place)

**Google Hot Trends →** <http://www.google.com/trends/hottrends> **→ search engine site**

* Google releases the number of people that searching for different topics using google search engines
* On the google hot trend website, google publishes the hot trend information for the following 8 countries: USA, UK, Australia, Canada, Germany, Hong Kong, India, Isreal, Japan, Russia, Signapore, Taiwan
* Google only makes the US hottrends data availible for via an RSS feed at <http://www.google.com/trends/hottrends/atom/hourly>; no other countries are availible there
* I was able to get the trend data for the other 7 countries by writing a ruby script that scraped the google hot trend web site.
* Google updates their data about what's trending in differnet countries about every 1-5 hours, depending on the country.

**Youtube →** [http://www.youtube.com](http://www.youtube.com/) **→ video search engine**

* **API resource, GET Most Popular**
* API returns the feed of the most popular videos for the past day
* Also can return the most popular videos for a particular geographical region for the past day:
* The most popular videos for each day is available for over over 39 countries.   
  EX of API resource call: <https://gdata.youtube.com/feeds/api/standardfeeds/most_popular>
* EX of API resource call: http://gdata.youtube.com/feeds/api/standardfeeds/regionID/most\_popular?v=2
* [http://developers.google.com/youtube/2.0/developers\_guide\_protocol\_video\_feeds](https://developers.google.com/youtube/2.0/developers_guide_protocol_video_feeds)

**Instagram → Instagram.com → photo sharing social networking platform**

* **API resource, GET Most Popular**
* This API gets a list of what media is most popular at the moment. Can return mix of image and video types. API will also return the latitude and longitude coordinates of popular item when availible.
* EX of API resource call: https://api.instagram.com/v1/media/popular?access\_token=ACCESS-TOKEN
* <http://instagram.com/developer/endpoints/media/>
* Instagram updates this API with new data at different rates; when the information is released is more sporadic than google hot trends or twitter.

**Yahoo Where On Earth Id → Geographical/Mapping Data Source →** <http://developer.yahoo.com/geo/geoplanet/guide/concepts.html>

* A yahoo Where On Earth ID (*WOEID*). Is a unique identifier that is assigned geographical entities on earth; this id is never changed; it acts as unique identifier for a specific place on earth.
* Using the yahoo WOEID API, you can retrieve the latitude and longitude, as well as other geographical information for a specific place. To call the WOEID API:
* '<http://where.yahooapis.com/v1/place/'woeid?[developerapikey>]
* The pieces of trend dating that I collected had to normally related; when taken directly from the datasource, that could relate pieces of trending data from different sources to one another. To solve the problem of normally relating the data, I used the yahoo woeid to map the data to a specific geographic place.

**Implementation of the Data Collection and Deliverables**

**General Design:**

To gather the data, mentioned above, I wrote ruby scripts that collected the trend data, parsed the trend data, and then inserted the data into my database. From an amazon ec2, server, I deployed these the ruby scripts to run at regular interverals using the cron-tab. Below, I list a few details about how I implemented of my database system. Additionally, all the scripts for this project were turned in and referenced below, incase you'd like to see how they were exactly implemented.

**Server:** for my project, I didn't think I could use the harbinger server because I need root access to run cron jobs continuously. As a result, I deployed a an amazon ec2 ubuntu 12.10 server instance.

Specifically, I set up a dual LAMP stack, as well Nginx and Ruby on Rails. As a deliverable, I created a shell script that does this whole process automatically. The file is called **setup\_ruby\_stack\_aws\_ubuntu12.sh.**

**Setting up the Geography/Modeling relation:** In a ruby script, I called the twiter, GET trends/available API to get a list of 415 places; these 415 places are distributed through out the world and also are countries. In the same ruby script, after getting the 415 places, I sent them to the Yahoo woeid API to get more information about them, like what there names were, the latitutude and longitude coordinate, etc. Then, I parsed the returned yahoo woeid API results, and inserted the location data into the data base. Some particularily helpful ruby gems that I used where:

the json gem: → this parsed the jason that API's returned.

There was also another great library called nokogiri that also can parse json and html.

To see the exact process I used, please refer to the script:

The pieces of trend dating that I collected had to normally related; taken directly from the datasource , there was nothing that inherently did this. To solve this problem, I related the trend data to geographic place using the the yahoo woeid

Yoeid. Using the twitter get\_availible

**Specific specs of the database project :**

For this project, I used an amazon

Specifically, I deployed the scripts on an amazon ec2 ubuntu 12.10 server instance. Using the cron-tab, I ran the scripts serveral times

1. <http://blogs.wsj.com/digits/2013/08/30/facebook-tests-trending-section-in-news-feed/> [↑](#footnote-ref-2)
2. [↑](#footnote-ref-3)