README – Twitter keyword propagation visualizer tool

Setup & Source Code details

As per our architecture the main components include the Data Collector, Data Processor & the Data Visualizer.

Data Collector:

Simple PHP collector uses the open source phirehose library to collect live tweets / retweets / mentions of a selected list of celebrities. *Input* List of celebrities to follow/monitor & *Output* Json files for each celebrity. We had to run the script in Jinx Cluster for about 2 weeks & Json files respect to each user was collected. A sample json file with minimal data is attached as an example.

Files

The **dbCollector** folder contains the collector scripts and the related configuration files and the licenses.

Get_tweets.php is the always alive script which needs to be run. The setFollow () method needs to be initiated with the list of celebrities we need to track and on an update the enqueueStatus () would be called were we process the tweet and direct it to the respective celebrities Json file. Also the _config files need to configure with the user account so as to call the streaming api.

Data Processor:

Inverted Index takes each of the json file collected above and returns the List of keywords sorted based on frequency, we simply use this information to generate suggestion keywords for our visualizer. Converts these raw json into related Celeb Information rows, Tweets, suggestion Keywords per celebrity & retweets at level 2 all in the form of MongoDB structures. Also a geo locations dictionary is built.

Files

"TwitterKeywordPropagation\src\twitterpropagation" contains the Java files which act as the data processors.

AIC_DataProcessing.java simply reads all the json files and populates the CTweets structure with a Tweet relates it with the celebrity and in case of retweets an additional link to its parent tweet is placed.

AIC_InvertedIndex.java parses each of the Json files and creates an inverted index forming the suggestions framework for each celebrity.

CountryNametoCountryCode.java & GeoLocationConvertor.java are the files using which the Geo Map dictionary is built

DB Utility file, **DBUtils.java** contains all the MongoDB CRUD operations logic for the Data Processor and the Data Visualizer.

Data Visualizer:

The server is an apache web server on struts framework, the Web Content/WEB-INF containing all the navigation and linking configurations.
The server side is a controller **TweetOperations.java**, which manages the Ajax calls making business logic operations using the DB Utility to return the results.
Client Side, Web Content folder contains scripts and the html files needed for the data visualizations. The Geographic keyword spread is achieved using the JQuery jhere plugin where as for the TimeLine and the compare charts the jqcharts library was used extensively.
twitterprop.js contains the logic for interactions Ajax Calls and the rendering of the user interface.

Setting up the Visualizer

- Import the web application project **TwitterKeywordPropagation** project into eclipse.
- Start the apache web server the landing page of the application is **http://localhost:9090/TwitterKeywordPropagation/jsps/twitterPropVisualHome.jsp**
- Install MongoDB & start the same using mongod.exe
- Access shell if required using mongo.exe.

Note

The dump of all the processed data needs to be imported so as the Visualizer to act upon. The total size of a mongodump file is about 4GB. Kindly contact us at lokresh88@gmail.com or sriram1991mv@gmail.com in case you want to play around with the tool. You can simply do a mongorestore using this dump folder and start working on the tool.

Useful Links:

- **https://dev.twitter.com/docs/streaming-apis**
- **https://github.com/fennb/phirehose**
- **http://docs.mongodb.org/manual/installation/**
- **http://jquery.com/download/**
- **http://www.jqchart.com/**