Life Time of a Trend

Sukshitha Pulijala
University of South Florida
Tampa, U.S.A
spulijala@mail.usf.edu

ABSTRACT

This project presents a framework to collect tweets for trending topics, the challenges that come with it and methods used to overcome them. The results are analyzed to show the kind of analysis that can be done and opportunities are discussed to extend the framework to collect other related data. Finally some relevant real world work is discussed to highlight the importance of collecting and analyzing such information.

KEYWORDS

Twitter, Trending, Trend life time

ACM Reference Format:
Sukshitha Pulijala. 2019. Life Time of a Trend. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Trends in twitter have a certain lifetime. It will be very valuable to be able to predict the lifetime of a tweet. For any prediction, historic data is required. Twitter, for very valid reasons only provides data that goes back to just a couple of weeks. In order to be able to do any meaningful prediction we either need to pay a service to provide that data or collect the data ourselves.

This project aims at providing the framework that will enable one to collect the data that is required, set a number of filters on data while collecting it, and also persist that filtered data. This framework was used to collect the data for the week of November 17^{th} to 24^{th} to show a few examples of the kind of analysis that can be done.

2 METHODS

As noted in the introduction, data needs to be collected continuously for however long the user requires. A robust framework that takes into account any downtime on Twitter's web service, network interruptions, and also code exceptions was developed.

2.1 Trending Topics

Twitter defines a topic as trending when it has a gradual sustained increase in volume of tweets. Twitter has an API to get a list of topics. Using the 'GET trends/place' API, and specifying the region as world wide, the top trending tweets at the time of request were collected.

2.2 Streaming Tweets

Twitter provides APIs to request for tweets containing a specific word, however the data received can go back at the most for a couple of weeks. In order to collect the tweets as they come in for the trending topics, this method would not be useful as we would have to continuously poll and in this method there is a high chance that we would miss counting on some tweets. Further, Twitter has a limit on the number of requests that can be made in a certain period.

To solve this issue, Twitter provides a streaming API in which the requester can register to filter tweets on certain words and Twitter would keep the connection open and continuously send the tweets back to the requester. This functionality was achieved using the 'POST statuses/filter' API.

2.3 Storing Tweets

The data received in each message from the Twitter web service contains the actual tweet, the time of the tweet, the user who posted the tweet, if it was a re-tweet or original tweet, etc. In this project, count was updated for the tweets that contained a word that was being tracked, so only that information was stored in the database. The framework can be extended to store other information like the location of the user, Original/Re-tweet details and other such information for it to be used for different kinds of analyses.

In order to continuously record the data as it is received, SQLite APIs were used to first store the 'word' that is being tracked, and then update the corresponding count in columns labelled with ordinal of the day.

A separate thread was created to keep track of time where in it would update the current day variable after 24 hours. The main thread would then use this variable to decide which column to increase the count in.

2.4 Categorize Data

After the data was collected for a week, every trending topic was manually categorized to further help in the analysis of data. In order to categorize each topic, the context in which the word was used had to be understood. In order to do understand the context, Twitter website was directly used with filters set to the dates when tweets were collected and the resulting tweets were analyzed.

This analysis for categorization was required because the same word could have been used once in the context of sports and/or in the context of movies the next time the same word is trending. This manual step of categorizing is a time taking process, but provides valuable insight.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2.5 Data Visualization

Using Python's NumPy, Matplotlib, and SQLite libraries the data stored in SQLite database was filtered and plotted by day and topic. This visualization was particularly useful to understand how few trends die quickly over a weekend and few are sustained and stay relevant throughout the week.

This visualization also leads to the question of 'why?' behind the trend. Understanding the 'why' can be further used to classify data or used in prediction using different data models, but that is beyond the scope of this project.

2.6 Track Every Top Trend

Twitter could update the top trends asynchronously. This framework allows one to poll for new top trends every few seconds and add to the list of new topics that are to be tracked. This project however did not enable that option. This was a feature that was thought to be of value after starting the data collection for the week of Nov 17-24, 2019. This feature can be further refined to understand when Twitter would drop a topic out from the top trending list.

2.7 Fail Safe Precautions

This framework can be run on a personal laptop or a server with no downtime, in either case there were many precautions taken to keep the data count accurate and take into account any disruptions that may happen over the data collection period.

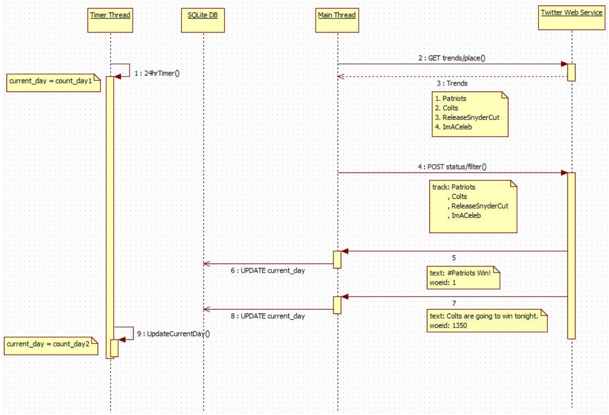
Topics to be tracked were collected at the start of data collection period and stored in a database. This was done to eliminate the failure case of storing the topics to be tracked in python's heap memory and losing that information in case of a program crash.

While streaming the tweets, there were multiple scenarios in which the connection could be disrupted both from the client and server side. Twitter occasionally could undergo maintenance or the network connection could break from the client side or it could also be that the server is under heavy load for a temporary time period. In all these scenarios appropriate error codes are generated, which were accounted for and the framework made impervious to. Some of the solutions, Twitter suggests that the client wait for a while before sending another request. This means that the tweets generated in this time period are not accounted for. The sleep periods were made small enough for the program to resume counting as soon as possible and long enough for Twitter to not throw the same error as the reason why the connection broke in the first place.

The tweets received are first persisted in a SQLite DB so a disruption in the stream does not require a restart of the program. As per design, the main thread is the only thread that accesses the SQLite database so the program does not run into file locking while writing or reading from it.

2.8 Sequence Diagram

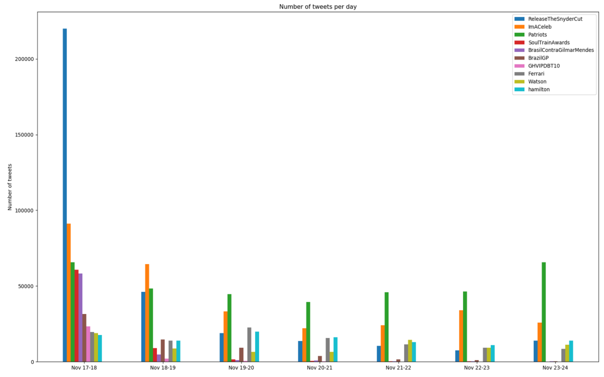
Below is a sequence diagram explaining the the program's execution flow. This gives a high level view only and does not describe the exception and fail safe handling.



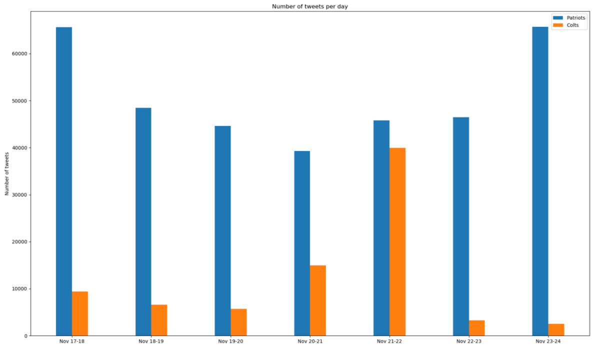
3 RESULTS

One of the distinctive features of data collected for the week of November 17th-24th was that the highest trending topics were not the ones that had sustained volume of tweets after the weekend. The highest trending topic was under the movie category and it is understandable that people would post more tweets about it over the weekend.

The graph below highlights how topics related to sports have sustained number of tweets all through the week.



The green bar is for the topic 'Patriots', which had a high number of tweets over the weekend. This topic picked up volume as the week went along and this was because the Patriots had another game scheduled that week. This topic was shown in Twitter as a trending topic, however it is very likely that this topic is relevant only for users in the US. For this topic to be among the top trending with the filter set for 'world wide trending' shows that Patriots have a huge fan base. This point can be proved even further by comparing the tweet volume for the word Patriots against the tweet volume for the word Colts, who also happen to have two games in the same week. Below is a graph showing that comparison.



Data was collected on November 17th after the Colts game which explains the low tweet volume even on the day when they had a game. Colts had a game on November 21st and Patriots on the 23rd and still the tweet volume for Colts was less than Patriots on the day when Colts had a game.

It can also be seen that the tweet volume gradually picks up before a game and maintains that high level if a team wins their game. Such patterns can be taken advantage of when targeting ads to the users or promoting products on social media. Since Twitter provides us information of where a certain user posted the tweet from, that data can also be gathered and fan base of a team can be analyzed to see of how spread out they are geographically.

4 DISCUSSION

In order to analyze the complete lifetime of a trend, data needs to be collected starting from 0 to the count when it is considered trending along with analysis done in this project where tweets are analyzed after it is declared trending. Since we do not know ahead of time which topics would be trending, we need to interact with the Twitter web service to get past data of the tweet and then analyze the time line.

Geographical locations of the users and their networks can be used to predict the contagion of a tweet. This data if collected and the patterns analyzed, it can be used to predict if a topic will be trending in the future. A study[1] was conducted using such methods to understand the spread of Flu. In a similar vein, another study[2] focused on collecting real time data just like this project conducted a case study to showcase how recognizing patterns can detect and predict the advent and changes of social issues.

This project focused on creating a framework that can gather data for trending topics in twitter and store it in a format that can be analyzed, however a topic that is not trending can also be tracked continuously and that can be used to understand the pulse of a product or company and classify positive or negative trends.

5 REFERENCES

1. Achrekar H, Gandhe A, Lazarus R, Ssu-Hsin Yu, Benyuan Liu. Predicting Flu Trends using Twitter data. 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on. April 2011:702-707.
doi:10.1109/INFCOMW.2011.5928903.

2. Min Song, Meen Chul Kim. RT$\hat{2}$M: Real-Time Twitter Trend Mining System. 2013 International Conference on Social Intelligence and Technology, Social Intelligence and Technology (SOCIETY), 2013 International Conference on, International Conference on Social Intelligence and Technology. May 2013:64-71.
doi:10.1109/SOCIETY.2013.19.

6 CODE

Here is the link to code and the database generated as a result of running the program for a seven day period.

https://github.com/sukshithapulijala/SMMTweetLifetime/