Working with Streaming Data: Analysis of Twitter Tweets

Anand Raghuraman
anand.raghuraman@wsu.edu

Ehdieh Khaledian
ehdieh.khaledian@wsu.edu

December 12, 2016

Abstract

As more data is generated, it is becoming increasingly important to be able to work with real-time data. In this project, we used twitter streaming API in two different ways to capture tweets-user streams and public streams. Then, we applied sentiment analysis on the captured tweets using Python and R to find how much people are happy in each U.S. state, which celebrity from *BreakingBad* series is the happiest and how frequent a certain word occurs in tweets.

1 Introduction

Social networks today has become a very popular communication tool among Internet users.Millions of messages are appearing daily in popular web-sites that provide services for social networks such as Twitter, Tumblr and Facebook. Authors of those messages write about their life, share opinions on variety of topics and discuss current issues. Because of a free format of messages and an easy accessibility of microblogging platforms, Internet users tend to shift from traditional communication tools (such as traditional blogs or mailing lists) to these services. As more and more users post about products and services they use, or express their political and religious views, these web-sites become valuable sources of people's opinions and sentiments.Such data can be efficiently used for marketing or social studies [**1**] .

Furthermore, as more data is generated, it's becoming increasingly important to be able to work with real-time data and being able to work with streaming data has become a critical skill for any aspiring data scientist. Real-time, or streaming, data is generated continuously, and in the case of the stock market, there can be millions of rows generated every hour. Due to size and time constraints, there often isn't a neat dataset that you can analyze and we'll need to either store the data to analyze later, or analyze it in real time, as we get it. It means, we are not dealing with historical data anymore [**2**].

In this project, we fetch the real-time data from twitter. In Twitter, the users create everyday very large number of short messages. The contents of the messages vary from personal thoughts to public statements. Therefore, a lot of information can be obtained from twitter as the users post everyday about what they like/dislike, and their opinions on many aspects of their life.

The list of different ways to use Twitter could be long, and with 50 million of tweets per day,there is a lot of data to analyze and to play with.
To work with twitter's data, Twitter Streaming API gives developers and data scientists access to multiple types of streams (public, user, site), with the difference that the streaming API collects data in real-time as opposed to the search API, which retrieves past tweets. Public streams are streams of public data that flow through twitter. We can use this for following specific users or topics, and for data mining.User streams contain data corresponding to a single user's stream. Lastly,site streams are multi-user version of the user stream [**3**].

In this project, we use the twitter streaming API to fetch two type of tweets from twitter using Python. Then we use Python and R for sentiment analysis. The results includes the term frequency, the happiest and unhappiest U.S. states and happiest actor in Breaking Bad Tv series. In this case, we used the positive and negative scores of each tweet to find the happiest and unhappiest people. It can be used for another purposes like approval and disapproval of people about a new decision, person, brand, movie, etc.

2 Problem Definition

Sentiment analysis over Twitter offer organizations, politician and social developers a fast and effective way to monitor the public's feelings towards their decision, brand, business, directors,etc. In most areas, online opinion has turned into a kind of virtual currency that can make or break a product or person or decision. Yet many companies and organizations struggle to make sense of the caterwaul of complaints and compliments that now swirl online. As real time sentiment analysis tools begin to take shape, they could help in improving their underlying or ultimate outcome or criterion. In this project, we apply Sentiment analysis on real-time fetched tweets to find a tweet that is most positive or negative.

3 Algorithms and implementation

We have split the entire portion into three parts namely
1)Twitter API.
2)Computing the frequency of each term in the tweets
3)Sentiment Analysis and applications.

3.1 Twitter API

Twitter has provided some APIs for developers and data scientists. Three APIs that we used in this project are:

Streaming API: The Streaming APIs give developers low latency access to Twitter's globalstream of Tweet data. A proper implementation of a streaming client will be, pushed messages indicating Tweets and other events have occurred, without any of the overhead associated with polling a REST endpoint. fig.1 shows the process of using Streaming API. In This project we fetched the small random sample of all public statuses.

Search API: The Twitter Search API allows queries against the indices of recent or popular Tweets and behaves similarly to, but not exactly like the Search feature available in Twitter mobile or web clients. Before getting involved, it's important to know that the Search API is focused on relevance and not completeness. This means that some Tweets and users may be missing from search results. We used search API to get the tweets that are related to a search term.

User API: User API helps us to fetch the tweets of a special user.In this project we used it to search the most recent tweets of breaking bad series actors. The output of this part will be a csv file including $username, tweet$.

3.2 Computing the frequency of different terms

We have used the Tweets that have been fetched and computed frequency of the terms using the formula:

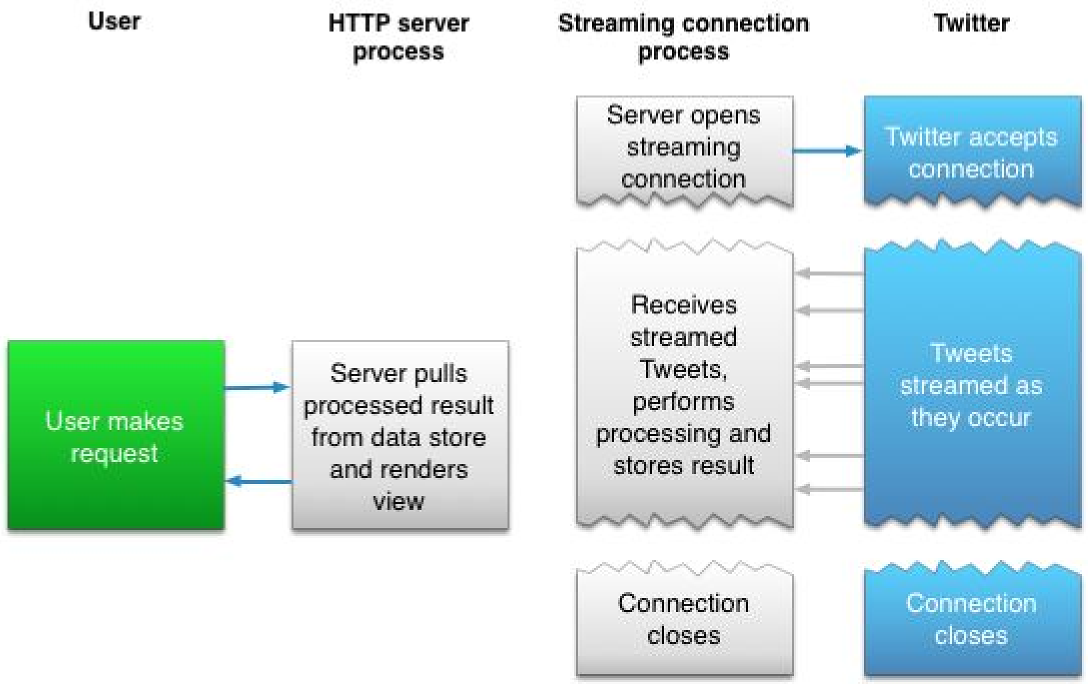

User
HTTP server process
Streaming connection process
Twitter
Server opens streaming connection
Twitter accepts connection
Receives streamed Tweets, performs processing and stores result
Tweets streamed as they occur
User makes request
Server pulls processed result from data store and renders view
Connection closes
Connection closes

Figure 1: Twitter Streaming API

Number of occurrences of the term in all tweets/Number of occurrences of all terms in all tweets

The inputs for this part is a file containing tweets and a file of stopwords. The program ignores the word that are listed in stop-word file.We converted all tweets to lower case and then we sorted our output so that the most frequent terms appear at the top.The output for this part is one pair per line as follows:
$[term][Frequency]$

We converted all tweets to lower case and then we sorted our output so that the most frequent terms appear at the top.

3.3 Sentiment analysis and its applications

For this part, we computed the sentiment of each tweet based on the sentiment scores of the terms in the tweet.The sentiment of a tweet is equivalent to the sum of the sentiment scores for each term in the tweet. We have a file that contains a list of pre-computed sentiment scores for every word. Each line in the file will contain a word or phrases followed by a sentiment scores. Each word or phrases found in a tweet, but not in our file will be a given a sentiment score of 0.

Algorithm used for Sentiment analysis:

1. Store the sentiment scores for words from the given sentiment file.
2. Then, store each tweet as a key in a dictionary.
3. Then parse through each tweet.
4. Pick each word from the tweet.
5. Check if it the entire string is an alphabetic string.
6. If not, try removing the extra characters like –(;!().) and test for the remaining word.
7. Then add scores from each word to the total tweet score.
8. Add the corresponding tweet score to the tweet's score in the dictionary.
9. After compiling the score for every tweet, sort a list based on the score.
10. Then print according to the expected output ; (score: tweet).

We have added a few more words to the sentiment words in a file like smileys. Furthermore, we trimmed down words like Love!!!–Love, Wow!!!!–Wow, Bad.–Bad (i.e, taking off the fullstop). This helped in including the score for these words which weren't detected because of the special characters or punctuations.

3.3.1 Analyzing the happiest actor

Using the algorithm we have implemented for sentiment analysis, we have found the happiest actor in "Breaking Bad" Tv series:

The algorithm we have implemented for this is:

1. Store the sentiment scores for words from the given sentiment file.
2. Then, store each actor as a key in a dictionary.
3. Then parse through each tweet.
4. Pick each word from the tweet.
5. Check if it the entire string is an alphabetic string.
6. If not, try removing the extra characters like$(-, -39; !.)$ and test for the remaining word.
7. Then add scores from each word to the total tweet score.

8. Add the corresponding tweet score to the actors score.
9. Compute the average sentiment score by using the formula: (Total Score/Total number of tweets by the actor).
10. After compiling the score for every actor, sort a list based on the score.
11. Then print according to the expected output—(score:username).

3.3.2 Analysing the top happiest and unhappiest states

we found the average sentiment score of each U.S. state using the algorithm implemented for sentiment analysis.The algorithm we implemented for this is:

1. Collect all states data as abbreviations and full names into a dictionary.
2. Collect all word sentiment score into a dictionary.
3. Parse each tweet text and compute the score for that particular tweet.
4. Immediately, find the related states to each tweet.
5. For finding related state, parse through the location and place.
6. In location attribute, find words of the format—($city, State$) or ($State, USA$) or ($StateAbbreviation, USA$) or ($City, State, USA$).
7. In the place attribute, split up the words and search for states or state abbreviations. Eg: I love Illinois.
8. Then evaluate the found state with the dictionary and that state to the related state list for that tweet.
9. For each state in the related state list, assign the tweet score to it and increase the tweet count for that state as well.
10. Compute the average sentiment score by using the formula—($TotalScore\ /Totalnumberoftweetsrelatedtothestate$).
11. Sort the list according to average score.
12. Print acording to the expected output: $Statescore$: $Stateabbreviation$.

Here, we performed some data cleaning and parsing before implementing the algorithm.They are as follows:

- The data fetched was done so in the json format.

- While checking each word in the tweet for states, all trailing and leading symbols were removed to make sure data isn't lost. Eg: $Miami - Miami, LandedinTexas$!!!! $- LandedinTexas$

- All the format checking was done after making sure there were a majority of them when compared to the dirty data.

- Words without symbols were trimmed to make sure sentiment scores were not missed. Eg: $Wow!!!! - Wow, Iamgood. - Iamgood$

- The final score is printed upto 4 decimal places after careful observation of the data to make sure efficient comparisons could be made.

- A pre analysis of the data showed that there was a higher probability of finding a state in the “user location”attribute than the “place” attribute.

4 Results and Discussion

Figure 2 shows the result for term frequency. It shows that like is the most frequent word in Twitter.Using this analysis, we can find how important a particular word is to a document in a collection or corpus.It is often used as a weighing factor in information retrieval or text mining.

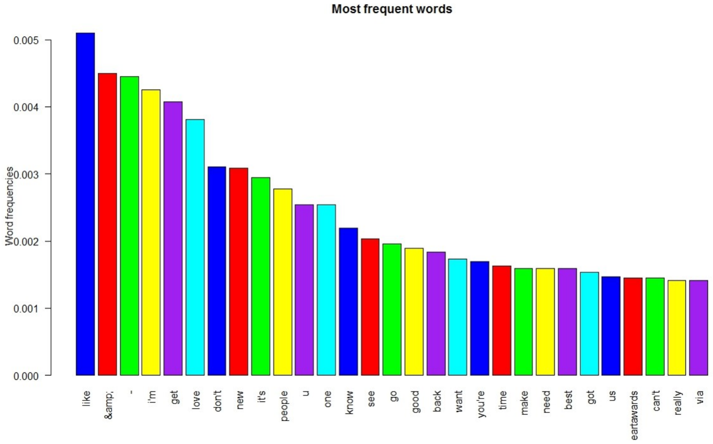


Figure 2: Term Frequency

Figure 3 shows the output for happiest actor of “Breaking Bad” series.It is very important to have a check on the happiness factor.We feel that results like these will be helpful for the director of a series to have a check of the actors of the show. There are many occurrences of actors leaving a show midway, these types of analysis would be helpful so as to make sure that no actor is left unhappy.

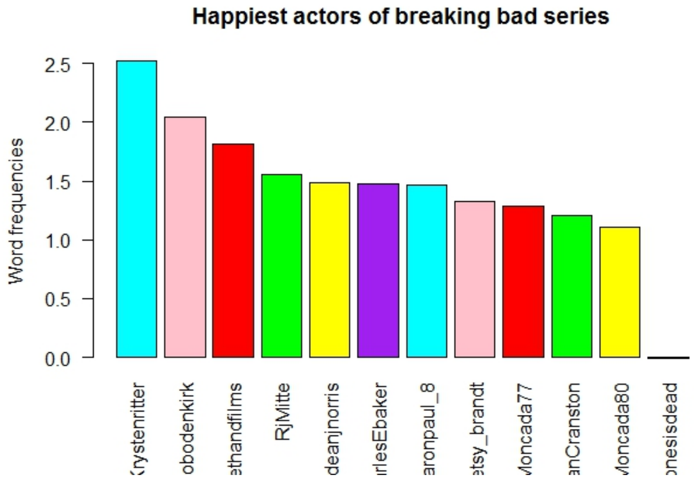
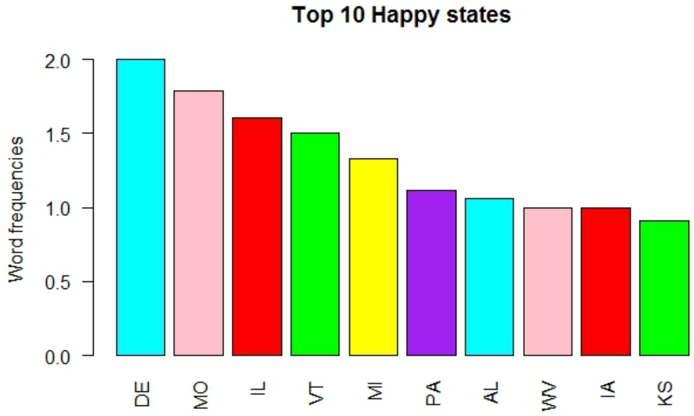


Figure 3: Happiest actor

Figure 4 presents the happiest and unhappiest states of the U.S.As a leader of a country, it is important that he/she makes sure that everyone is happy. We feel the type of analysis we did would be very helpful for him/her to find out the reason why the state is unhappy and have proper measures taken place to see that everyone in his country is happy.



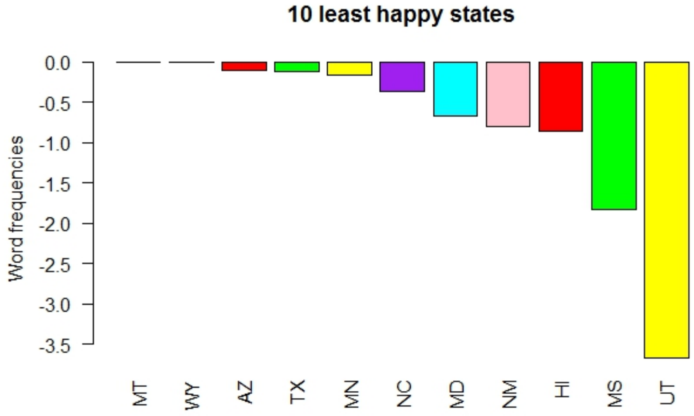


Figure 4: Happiest and Unhappiest states

5 Related Work

Sentiment analysis is widely used in analyzing many social networks data including Twitter for variant purposes such as stock market, election, etc. Although it is widely used, most of the applications use only historical data.For example: Jichang Zhao in his project[**4**], used sentiment analysis on tweets.he built a system called MoodLens, which is the first system for sentiment analysis of Chinese tweets. In MoodLens, 95 emoticons were mapped into four categories of sentiments, i.e. angry, disgusting, joyful, and sad, which serve as the class labels of tweets. They then collected over 3.5 million labeled tweets as the corpus and trained a fast Naive Bayes classifier, with an empirical precision of 64.3

In our project, rather than using a past data, we have used live data that is streaming or has been tweeted at the same time. This helps us in getting the present emotion of people rather than their emotion over the past one week or so.

6 Conclusion

In this project we benefited from Twitter Streaming API and Sentiment Analysis together in order to finding the term frequency in twitter tweets, finding the happiest actor of the Breaking Bad series and finding the happiest and unhappiest states in the U.S. In our project, we have not considered abbreviations like BFF(Best Friends Forever), BRB(Be Right Back), etc. occurring in any of the tweets. In future, we can optimize our algorithms so as to involve these abbreviations as there is a big transition towards these nowadays.

References

[1] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classiiňĄcation using distant supervision. Technical report, Stanford.

[2] Wang, Hao, et al. quot;A system for real-time twitter sentiment analysis of 2012 us presidential election cycle.Proceedings of the ACL 2012 System Demonstrations. Association for Computational Linguistics, 2012.

[3] https://dev.twitter.com/streaming/overview

[4] Zhao, Jichang, et al. quot;Moodlens: an emoticon-based sentiment analysis system for chinese tweets.quot;ÂăProceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012