Working with Streaming Data: Analysis of Twitter Tweets

Anand Raghuraman
anand.raghuraman@wsu.edu

Ehdieh Khaledian
ehdieh.khaledian@wsu.edu

November 27, 2016

Abstract

1 Introduction

2 Problem Definition

3 Algorithms

We have split the entire project into four parts namely
1)Fetching tweets.
2)Finding the frequency of each term in the tweets
3)Sentimental Analysis on the tweets.
4)Analyzing the happiest actors in breaking bad tv series
5)Finding out the top 5 happiest states.

3.1 Fetching Tweets

Algorithm 1 is used for fetching tweets. We have used Twitter API in order to fetch the tweets.

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: [number of occurrences of the term in all tweets]/[number of occurrences of all terms in all tweets]

3.3 Sentimental analysis

Algorithm 3 will compute 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.

Algorithm 3:

Given: a sentiment file having scores for individual words and the tweets file

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).

3.4 Analyzing the happiest actor

Algorithm 4 is used to find out the happiest actor in breaking bad TV series.

Algorithm 4:

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. If the tweet has the name of any actor in it, copy it into a new file named 'ActorTweet'.
5. Pick each word from the tweet.
6. Check if it the entire string is an alphabetic string.
7. If not, try removing the extra characters like ("-,-'-!-#-(-)-.) and test for the remaining word.
8. Then add scores from each word to the total tweet score.
9. Add the corresponding tweet score to the actor's score.
10. Compute the average sentiment score by using the formula - (Total Score/Total number of tweets by the actor).
11. After compiling the score for every actor, sort a list based on the score.
12. Then print according to the expected output -> (score: username).

3.5 Analysing the top happiest and unhappiest states

Algorithm 5 is used to find out the top happiest and unhappiest states in the US.

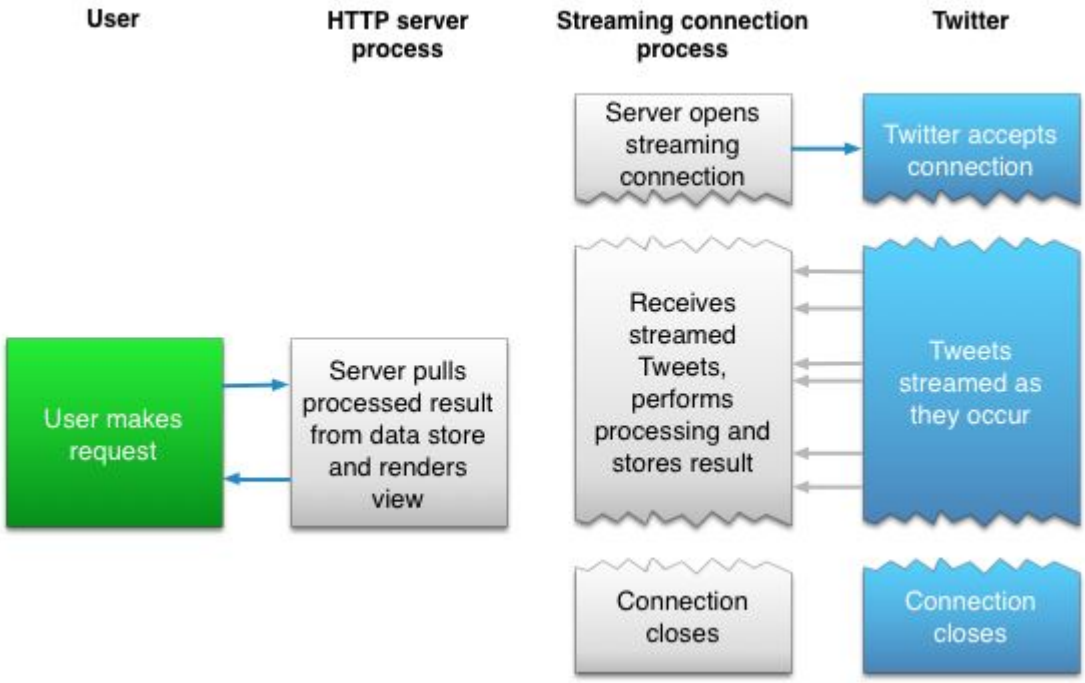
Algorithm 5:

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 attribute.
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/Total number of tweets related to the state).
11. Sort the list according to average score.
12. Print according to the expected output: <State score: State abbreviation>.

4 Implementation

The Dataset for this project is derived from live twitter feeds using Twitter's Streaming API.The Streaming APIs give developers low latency access to TwitterâĂŹs global stream 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.Connecting to the streaming API requires keeping a persistent HTTP connection open.

Figure 4.1 gives a schematic representation of the process:



The streaming process gets the input Tweets and performs any parsing, filtering, and/or aggregation needed before storing the result to a data store. The HTTP handling process queries the data store for results in response to user requests.

Our Analysis had three different parts.So, we used three different types of data from the tweet file we derived from the API.

For the first part, we used the entire data in order to compute the term frequency.Using the algorithm stated above, we were able to compute the term frequency. For finding out the happiest and unhappiest actors in the breaking bad TV series, we had to use only part of the dataset that was relevant for the analysis and thus had to first store the required data from the dataset into a separate file and then use the algorithms stated above to perform the analysis.For the last part of the project in which we found the happiest and unhappiest states in US, we did the a similar process as we had done for finding the happiest actors.In addition to these data we had used two datasets, one of which had the list of all stopwords and another which had the list of sentiment scores for individual words.