Twitter Sentiment Analysis using Apache Storm

Ishana Raina, Sourabh Gujar, Parth Shah, Aishwarya Desai, Balaji Bodkhe

Abstract- In today's highly developed world, every minute, people around the globe express themselves via various platforms on the Web. And in each minute, a huge amount of unstructured data is generated. This data is in the form of text which is gathered from forums and social media websites. Such data is termed as big data. User opinions are related to a wide range of topics like politics, latest gadgets and products. These opinions can be mined using various technologies and are of utmost importance to make predictions or for one-to-one consumer marketing since they directly convey the viewpoint of the masses. Here we propose to analyze the sentiments of Twitter users through their tweets in order to extract what they think. We classify their sentiments into three different polarities – "positive", "negative" and "neutral." Since, 6000 tweets are generated every second and this number is increasing, we need a robust system to process these tweets in real-time. Here, batch-processing would have its limitations and therefore a real-time and fault tolerant system, Apache Storm is used. After classifying the tweets, we represent the analysis in the form of graphs and charts which will enable our system users to understand public sentiments on the fly. This process as a whole is also called as Opinion Mining or voice of the customer.

Index Terms- Batch-processing, Microblog, Opinion Mining, Polarity, Sentiment, Storm, Tweets, Unstructured data.

I. INTRODUCTION

Earlier, natural language processing was applied to news sites and blogs where the quantity of data was large. But in today's time, micro blogging sites are becoming more popular where length of the post is shorter but the number of posts per day has shot up. Users have lesser number of words to express their opinion and hence tend to convey their sentiments through emoticons and acronyms, making sentiment analysis a more difficult task. We overcome this problem by making use of dedicated dictionaries for lexical words and emoticons associated with their respective polarities. Alongside, two dictionaries are used to identify stop words and acronyms. For instance, when a movie is released, the viewer tweets about his/her opinions. With the help of our system the director or the marketing team of the movie can modify their marketing strategy according to the generated analysis. In another case, when the customer tweets about a particular product, whether he finds the product appealing or he desires any specific changes, the owner/developer of the product can use our system to get a better understanding of the mass opinion and make any changes if necessary so as to increase the sales.

Manuscript Received on November 2014.

Ishana Raina, Department of Computer, Modern Education Society's College of Engineering ,University of Pune, Pune, India.
Sourabh Gujar, Department of Computer, Modern Education Society's College of Engineering ,University of Pune, Pune, India.
Parth Shah, Department of Computer, Modern Education Society's College of Engineering ,University of Pune, Pune, India.
Aishwarya Desai, Department of Computer, Modern Education Society's College of Engineering ,University of Pune, Pune, India.
Prof. B. K. Bodkhe, Department of Computer, Modern Education Society's College of Engineering ,University of Pune, Pune, India.

On a more individual front, if a Twitter user wants to analyze his own behaviors or mood over a certain period of time based on his tweets, he can use the system to view his average happiness over that period of time. Since users around the world regularly tweet about recent happenings (whether in their personal lives or in a public domain), it demands a fast, scalable, reliable and fault-tolerant system that processes tweets continuously. This is the reason we chose our platform to be Apache Storm. We provide input to the Storm cluster which streams the tweets in real time and provides the robust system we need, in order to process the thousands of tweets being generated every second. The fully developed application will provide live statistics. These statistics can be in the form of graphs, charts or relationship maps and will provide easy readability and understandability of user sentiment (whether positive, negative or neutral).

II. DATA DESCRIPTION

The tweets on Twitter are limited to 140 characters. This results in users using acronyms, emoticons to express their opinion. Following are the elements of a tweet:

A. Hashtags

The hashtag symbol (#) used in the tweet is for categorizing the tweet into domains. You can use multiple hashtags in a single tweet.
E.g. #Independenceday : This tag refers to the domain Independence Day, all the tweets containing this hashtag will be under this domain.

B. Acronyms

The tweets that are posted are not always in full length words. There are abbreviations and acronyms which are used due to the limited characters allotted for a tweet. During analysis these acronyms need to be expanded and analyzed as well since they convey a considerable amount of information. For this purpose we maintain a separate dictionary for acronyms.
Eg1. LOL : Laughing out loud. Category: Positive
Eg2. Fab : Fabulous. Category: Positive
Eg3. Plz : Please. Category: Neutral

C. Emoticons :

These are symbols like various operators put together to signify the emotions through the formed emoticon to keep the tweet concise. Since the emoticons represent significant amount of information regarding the polarity of that tweet, we maintain a dedicated dictionary for emoticons.
Eg1. :) Smiley. Category: Positive
Eg2. :(Sad. Category: Negative
Eg3. :'(Crying. Category: Negative

D. URLs:

URLs in a tweet are shortened with URL shortners (eg. Bit.ly and tinyurl.com) due to the tweet constraint. These URLs point to some external citations (where the tweet includes a reference and the external source. URLs are detected by the character strings "http://", "https://" and "www."
Eg1. http://goo.gl/l6MS

E. Twitter Handles :

The @ sign is used to call out usernames in Tweets: "Hello @SourabhGujar!" also known as handles. People will use your @username to mention you in Tweets, send you a message preceding or succeeding the handle along with a link to your profile.

F. Character Repetitions:

Usually people, to bring their feelings into focus tweet with words which have repetitive letters. "@IndianCricketTeam Today's game was longggggg." The word 'longggggg' is identified as a character repetition.

G. Intensified Words:

An uppercase word to intensify the meaning. "@BCmbatch I LOVE today's weather." Here the word LOVE is intensified and hence helps in categorizing the polarity of the tweet.

III. OVERVIEW OF APACHE STORM

Apache Storm, a distributed computation framework, created by Nathan Marz, adds reliable real-time data processing capabilities to Apache Hadoop. It is fast, scalable, reliable and can be programmed using a variety of programming languages. Its architecture consists of three primary node sets:

A. Nimbus Node

This is the master node. It uploads the computation to be performed in the cluster, launches worker nodes and even re-assigns worker nodes in case of failure. There is only one master node in a cluster.

B. Zookeeper Nodes

These nodes are assigned on every slave machine. The basic function of the zookeeper nodes is to keep a check on the processing happening on the worker nodes. The nimbus communicates with the worker nodes through the zookeeper.

C. Supervisor Nodes

There nodes, assigned on every slave machine, start and stop workers according to commands from the nimbus. There can be multiple worker nodes on a single slave machine. A key abstraction in Storm is the topology which is in fact, the program that keeps running in the Storm cluster. A visual representation of the topology is a network of the spout and bolts that Storm employs to perform its computation. A spout is an input stream that generates input tuples. Spouts are the source of data in a Storm cluster. In order to receive real-time data, a spout can be configured with an API or a queuing framework like Kafka. The spout sends data to bolts. These bolts are where the actual processing takes place and a cluster may have multiple bolts for the various processing steps required to achieve the desired result. Bolts can pass data further on to another bolt or to a location of data storage. But since, Storm involves real-time computation of a huge volume of data, storage is not a common practice.

IV. DATA COLLECTION

Twitter offers three ways to access its data and these are:

A. Twitter Search API

Twitter Search API helps to access a data set that exists from tweets previously written. In the Search API, users ask for tweets that match a search criteria or even a part of it. The criteria could be keywords, usernames or locations. But there is a limit to the number of tweets that can be accessed. For an individual user, the maximum number of tweets that can be received is the last 3,200 tweets, regardless of the query criteria. With a specific keyword, Twitter only polls the last 5,000 tweets per keyword.

B. Twitter Streaming API

Twitter Streaming API is a push of data as tweets happen in near real-time unlike the Twitter Search API. With Twitter Streaming API, users register a set of criteria (keywords, usernames, locations, named places, etc.) and as tweets match the criteria, they are pushed directly to the user. This API is free of cost although the percentage of total tweets users receive with the Twitter Streaming API varies heavily based on the criteria users request and also on the traffic. Therefore, use of this API cannot be relied upon to generate a large number of tweets for real-time processing.

C. Twitter Firehose

Twitter Firehose guarantees delivery of 100% of the tweets that match the criteria. It is very similar to the Twitter Streaming API as it pushes data to end users in near real-time. But the difference lies in the cost. Twitter Firehose is not free of cost, unlike the Streaming API.

D. Implementation

The Storm spout is linked to the Twitter API and is responsible for the streaming of tweets into the Storm cluster. The spout does not perform any processing on the data. It simply streams it. These tweets are then sent by the spout to the bolt in the cluster so that they can be processed.

V. SENTIMENT CLASSIFIER

The tweets are broken down into tokens where each token is assigned a polarity, which is a floating point number, in the range from -1 to 1: negative sentiments being -1, neutral being 0 and positive being 1. The overall sentiment is then calculated by adding the polarities of each token. The evaluated sum is then rounded off to the nearest integer, thus assigning the final polarity to the tweet.

A. Positive Tweets:

- Tweets indicating an achievement or celebration
- Tweets wishing or congratulating someone
- Tweets using emoticons such as :) , :D , =) , =D, ^_^ , <3

E.g.: Chasing 275 to win, India comfortably reached the target with 28 balls to spare due to some excellent batting from the top order. #IndvsSL

B. Negative Tweets:

- Tweets indicating boredom (E.g. if a movie wasn't entertaining)
- Tweets indicating dejection (E.g. due to a tragedy in one's personal life)
- Tweets indicating anger (E.g. due to ongoing riots in an area)
- Tweets using emoticons such as :(, :'(, :/ , -_-

E.g.: I just lost $203 and I'm so angry! How do you deposit $ into the wrong account after I gave you the correct information! #Angry

C. Neutral Tweets:

- Tweets including both positive and negative reviews
- Tweets including neither positive nor negative sentiments
- Tweets presenting facts or theories

E.g.: I have been studying whole day long. Hope it helps!!!!

VI. RESOURCES

For analyzing the tweets, we will make use of various dictionaries from where the polarity of any part of the tweet can be decided. We are using four kinds of dedicated dictionaries:

A. Lexical Dictionary:

We will make use of a lexical dictionary which will have most of the English words listed. This dictionary will help us analyze the tweet in an error free manner by matching the word in the tweet with the one in the lexical dictionary. If any given word is not found in the dictionary then, we will check whether the word contains any repetitions and will be categorized accordingly. In our system we will use the Dante database of analyzed text of English. It will contain the headwords, multiword, idioms and phrases.

B. Acronym Dictionary:

This dictionary is used to expand all the abbreviations and acronyms. Expansion of these acronyms will generate words which require further analysis using the Lexical Dictionary to classify them into polarities.

C. Emoticon Dictionary:

A tweet may contain emoticons. These are textual portrayal of the Tweeter's mood put together to convey a meaning. An emoticon dictionary will play this role. We will have a wide range of emoticons along with their meaning.

D. Stop Words Dictionary:

In any given tweet, not all the words will have a polarity and they need not be analyzed. Hence they are eliminated and tagged as stop words. We maintain a dictionary with a list of all of the stop words, without their meaning since it is not needed. E.g. Able, both, welcome

VII. PROCESSING OF DATA

A. Tokenization

All the words in a tweet are broken down into tokens. This is the tokenization process. For example, '@username I had an amazing time today!' is broken down into individual tokens such as '@username', 'I', 'had', 'an', 'amazing', 'time', 'today'. Emoticons, abbreviations, hashtags and URLs are recognized as individual tokens. Each word in a tweet in a separated by a space. Therefore, on encountering a space, a token is identified.

B. Normalization

Firstly, the normalization process verifies each token and performs some computing based on what kind of token it is.

- If the token is an emoticon, its corresponding polarity is taken into account by searching the emoticon dictionary.
- If the token contains the character sequence 'n't' such as in 'don't', 'can't', 'won't', then the corresponding word is replaced by 'not'. 'Not' is now identified as a token.
- If the token is an acronym, it is checked in the acronym dictionary and the full form is stored as individual tokens. For example: 'brb' is stored as 3 different tokens: 'be', 'right', 'back.'
- Intensifiers such as 'AWESOME' are converted into lowercase and the token is stored as 'I_awesome'. The basic idea is to preserve the emphasis of the user's emotions while he/she was tweeting. The 'I_' means that the word, when in the tweet, was an intensified word.
- Spellings of character repetitons such as 'veryyyyy' are first corrected into 'very' and then stored as 'R_very'. The 'R_' is used for the same purpose as that of the 'I_'.

The normalization process also discards all those tokens which, in no way, contribute to the sentiment of a tweet. Stop words such as 'this', 'while', 'because' do not indicate any sentiment in themselves and hence are discarded. Similarly, URLs specified in a tweet and the Twitter Handle can also be safely discarded.

C. Part-of-Speech Tagging

The valid tokens are then passed to the part-of-speech tagger which attaches a tag to each token, specifying whether it's a noun, verb, adverb, adjective etc. Part-of-speech tagging helps determine the sentiment of the overall tweet because words have different meanings when represented as different parts of speech. For example, the word 'net' when used as an adjective such as 'net profit' has a positive ring to it whereas the word 'net' when used as a noun in the context of fishing does not bear either a positive or a negative tone.

D. Implementation

The bolts in Storm are programmed in order to carry out the above mentioned processing. Bolts are a component of the Storm topology which can receive input, process data as well as send output. This output can either be sent to another bolt or to a data storage location. Here, we propose using six bolts, on each worker node. The stream of tweets obtained from the spout is sent to the first bolt. It performs the tokenization and these tokens are then sent to the second bolt. In the second bolt, the normalization process is carried out and only valid tokens remain. These tokens are then sent to the third bolt which performs part-of-speech tagging. On the fourth bolt, each token is assigned its polarity by performing a keyword search in the dictionaries and extracting its corresponding floating point polarity. The overall polarity of the tweet is then calculated and sent to the fifth bolt.

The fifth bolt now receives the polarities of each tweet and computes its average. This mean value is then passed to the sixth bolt, where it rounds off the mean value to the nearest integer and displays the overall sentiment in the form of a graph or pie chart.

VIII. DATA REPRESENTATION

The tweets after being processed using Apache Storm for their polarities are generated and produced in the form of various outputs. These outputs help the user to analyse the tweets conveniently and can be understood easily.

A. Bar Graphs:

These graphs will have the usual bars with two components on either of the axes. For instance, to provide the average happiness of a given user, the X-axis will have the happiness marking and the Y-axis will have the duration, usually in the form of days. With these two, the bar graph will be plotted.

B. Pie Charts:

Pie charts are circular graphs which represent the statistics in the form of percentile sectors. For example, the number of tweets having that polarity will be classified into the sectors of the pie chart.

C. Timeline:

The timeline representation of the tweets helps in plotting the tweet's corresponding time along with the polarity of that tweet. The duration of the time can be varied from n number of hours of given a day to number of days of any given month.

D. Maps:

The map will show where the tweet came from and will also indicate its polarity. This will be very helpful in case of local polls since it will indicate the polarity area wise. We will use the Google Maps Javascript Library to render the maps.

IX. CONCLUSION

Micro blogging websites are a common platform where people all over the world can share their opinions and views on various issues and topics. The most obvious advantage that results from analyzing user sentiments is in advertising- be it one-to-one consumer marketing to ensure that the user is shown products related to where his/her interest lies and to provide a good experience or an online shopping site extracting user opinion from the various reviews people post about a particular product. Hence, by using our proposed system, potential customers can be turned into regular ones.

ACKNOWLEDGMENT

We would like to express our deep sense of gratitude towards our project guide, Prof. B. K. Bodkhe for guiding us through the entire process right from doing the research till penning our thoughts in the form of this paper. We would also like to thank our parents and colleagues for their valuable time and inputs that have helped us making this paper a reality.

REFERENCES

1. Changbo Wang, Zhao Xiao, Yuhua Liu, Yanru Xu, Aoying Zhou, and Kang Zhang, "SentiView: Sentiment Analysis and Visualization for Internet Popular Topics", IEEE Transactions On Human-Machine Systems, Vol. 43, No. 6, November 2013
2. Efthymios Kouloumpis, Theresa Wilson, Johanna Moore, "Twitter Sentiment Analysis: The Good the Bad and the OMG!", Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media
3. Rushabh Mehta, Dhaval Mehta, Disha Chheda, Charmi Shah and Pramila M. Chawan, "Sentiment Analysis and Influence Tracking using Twitter" in International Journal of Advanced Research in Computer Science and Electronics Engineering, Vol 1, Isuue 2, May 2012
4. Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow and Rebecca Passonneau, "Sentiment Analysis of Twitter Data", Department of Computer Science, Columbia University
5. Bo Pang and Lillian Lee, "Opinion Mining and Sentiment Analysis", Foundations and Trends in Information Retrieval Vol. 2, No 1-2 (2008)
6. Aditya Pal & Scott Counts, "Identifying Topical Authorities in Microblogs", WSDM'11, February 9–12, 2011, Hong Kong, China, Copyright 2011 ACM
7. Jianshu Weng, Ee-Peng Lim, Jing Jiang, Qi He, "TwitterRank: Finding Topic-sensitive Influential Twitterers", WSDM'10, February 4–6, 2010, New York City, New York, USA Copyright 2010 ACM

AUTHORS PROFILE

Ishana Raina, Department of Computer, Modern Education Society's College of Engineering ,University of Pune, Pune – 411001, India.

Sourabh Gujar, Department of Computer, Modern Education Society's College of Engineering ,University of Pune, Pune – 411001, India.

Parth Shah, Department of Computer, Modern Education Society's College of Engineering ,University of Pune, Pune – 411001, India.

Aishwarya Desai, Department of Computer, Modern Education Society's College of Engineering ,University of Pune, Pune – 411001, India.

Prof. B.K. Bodkhe, Department of Computer, Modern Education Society's College of Engineering ,University of Pune, Pune – 411001, India.