Hashtag investor – Perception Analysis with Relation to Geographical Location in Twitter

Samitha Kolambage*, Hasath Tillekeratne*, Niroshan Chathuranga*, Hasanthi Devendra*, Muditha Tissera Prince
Department of Information Technology, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka

Article Info:
Article history:
Received 25 July 2017
Received in revised form
20 August 2017
Accepted 28 August 2017
Available online 15 December 2017

Keywords:
Twitter; Sentimental analysis; machine learning; Clustering; Graph mining; Data mining

Abstract

Hashtag investor is a system that can analyze twitter data to generate useful information including some predictions. Machine learning techniques have been used for this research which falls into data mining to archive sentiment analysis to categorize and identify tweets based on the contents. Twitter has an enormous collection of data. If these data is converted into some useful information, accurate decisions can be made using this data. That is our main objective, which can be very helpful to users, and this system works with respect to four specific objectives. One objective is sentimental analysis of twitter data and finding false tweets. Supervised learning has been used and NLTK and also the naïve Bayes classifier has been used as techniques. The output will be display percentage wise, negative positive and neutral percentages of the given keyword. Twitter data is analyzed according to the given keyword. False tweets identification is done by analyzing user profile. If the user profile criteria does not match with our assumptions this profile is marked as a fake profile. Second objective is comparing two similar products and getting the popularity according to the time. The output is displayed by charts. Similar keywords will be grouped. Clustering algorithms has been used for grouping. Our forth objective is finding some latest ongoing events and the number of users who were active at certain time periods, ARIMA model has been used as the technique. Our final objective is to analyze retweets comments and tweets on particular two products. Output is displayed as a graph. Propagation topology is used as the technique for retweet analysis and exponential regression function is used for popularity prediction.

1. Introduction

Social media websites have evolved to become a source of varied kind of information. This is due to nature of Social Medias on which people post real time messages about their opinions on a variety of topics, discuss current issues, complain, and express positive sentiment for products they use in daily life. In fact, companies manufacturing such products have started to poll these Social Medias to get a sense of general sentiment for their product. Many times these Companies study user reactions and reply to users on Social Medias. One challenge is to build technology to detect and summarize an overall sentiment.

In this paper, one such popular Social media Called Twitter. Twitter is a popular Social media service where users create status messages (called\tweets"). These tweets sometimes express opinions about different topics. Information available from social networks is beneficial for analysis of user opinion, for example measuring the feedback on a recently released product, looking at the response to policy change or the enjoyment of an ongoing event. Manually identify this data is difficult and potentially expensive.

In this paper models was built for classifying “tweets” into positive, negative and neutral sentiment, identify time evolution in products from tweets, analyze events through tweets , analyze twitter accounts , analyze likes comments retweets on tweets and analyze geo relations with these tweets.

By getting twitter data using twitter API call, For that we used twitter streaming API which can get large amount of tweets at a time rather than using other API’s for data saving part we used mongo db.

One advantage of this data, over previously used data-sets, is that the tweets are collected in a streaming fashion and therefore represent a true sample of actual tweets in terms of language use and content. Our new data set is available to other researchers.

The rest of the paper is organized as follows: Section II is a literature review of the work carried out so far. Herein, the drawbacks as well as advantages of these paradigms have been discussed. Section III discusses the proposed system and its aspects. Section IV is on the results that were shown by the proposed method and finally Section V is on the conclusion and future work.

***Corresponding Author**
E-mail address: kolambagesamitha32@gmail.com; maxhasath@gmail.com; nioshan5677, hani@gmail.com; muditha.t@sliit.lk

2. Literature Review

Throughout the phase of literature survey several applications were found which use sentimental analysis, Data mining and Machine learning to analysze and interact with the social media. Still there are lots of researches carried out on Social media interactions and analyses. Sentiment analysis has been handled as a NaturalLanguage Processing task at many levels of granularity. Sentiment analysis of Twitter data are by Go et al. (2009) they use tweets ending in positive emoticons like “:)” “:-)” as positive and negative emoticons like “:(” “:-(” as negative. They build models using Naive Bayes, MaxEnt and Support Vector Machines (SVM), and they report SVM outperforms other classifiers. In terms of feature space,They try a Unigram, Bigram model in conjunction with parts-of-speech (POS) features. They note that the unigram model outperforms all other models. Specifically, bigrams and POS features do not help [1].

Another significant effort for sentiment classification on Twitter data is by Barbosa and Feng (2010).

They use polarity predictions from three websites as noisy labels to train a model and use 1000 manually labeled tweets for tuning and another 1000 manually labeled tweets for testing. They however do not mention how they collect their test data. They propose the use of syntax features of tweets like

retweet, hashtags, link, punctuation and exclamation marks in conjunction with features like prior polarity of words and POS of words. Their approach was extend by using real valued prior polarity, and by

combining prior polarity with POS. Our results show that the features that enhance the performance of our classifiers the most are features that combine prior polarity of words with their parts of speech. The tweet syntax features help but only marginally[1].

Researchers have also begun to investigate various ways of automatically collecting training data. Several researchers rely on emoticons for defining their training data (Pak and Paroubek 2010; Bifet and Frank 2010). (Barbosa and Feng 2010) exploit existing Twitter sentiment sites for collecting training data. (Davidov, Tsur, and Rappoport 2010) also use hashtags for creating training data, but they limit their experiments to sentiment/non-sentiment classification[2]. But in ours rather than trying negative positives trying to get emoji classify neutral tweet removing, key word popularity and event handling and many more improvement .

Tweeter is already gives a Tweeter analyzer application to users who have tweeter account can logon to their tweeter account through it.

But they can only check positive negative tweets of their own account, rather than analyzing all tweet accounts.

3.Methodology

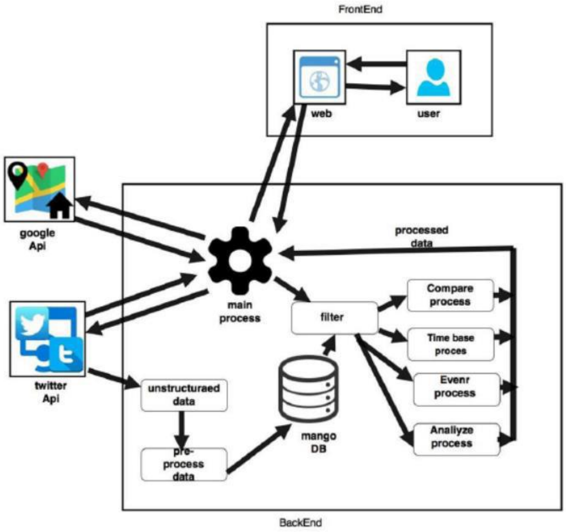


Fig. 1. High Level Dataflow Diagram

In this research our main objective is Twitter has an enormous collection of data. If these data is converted into some useful information, accurate decisions can be made using this data. That is our objective, which can be very help to users. This is more useful as the details of a particular location are displayed using Google maps for accurate decision making.

This research uses python as the programing language and mongo DB as the database. Our first step was to collect data from twitter which is not an easy task. To use the Twitter API was planned to collect tweets and other data from their site. The REST API, the Search API, and the Streaming API are the three methods to get this data. The Search API allows you to search old tweets the REST API allows you to collect user profiles, friends, and followers, and the Streaming API collects tweets in real time as they happen. There were some restrictions when getting data through API. So, that to keep our own database.

First, get API keys from twitter developers' site. The authentication requires that an API key from the Twitter developers' site. The authentication gives permission to our program to make API calls.

These are rather comprehensive with the amount of data, but hard to use without them being parsed first. Using NoSQL database like MongoDB to store and query tweets.

To use the twitter API, a set of "keys" are required. To obtain these keys, a twitter account is required. These keys can be obtained by visiting ***http://apps.twitter.com,*** and logging in using a twitter account credentials**.** It appears as though twitter
has acknowledged the use of their older API for various purposes.

The user is then required to create a twitter "application", which in turn will provide all the required keys to access the twitter API functionality.

These tokens seemingly add more security to the use of the API, since now all the API calls are made through the twitter account connected to the application. Therefore, even twitter itself advises to keep theses keys hidden, and not human readable.

When comes to the geo locations. It shows the geo location using Google maps according to Twitter accounts. To do that task, used Google Map Geocoding API. Geocoding is the process of converting addresses (like a street address) into geographic coordinates (like latitude and longitude), which you can use to place markers on a map, or position the map

The research has four core main components.

A. Topic analysis.

B. Event analysis,

C. Time analysis,

D. Topic comparison.

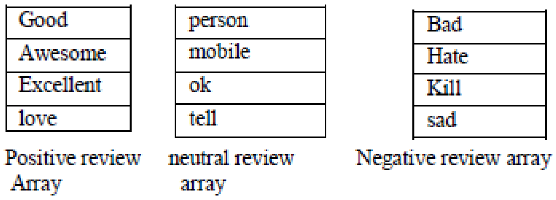
Our goal is to build a system that can be analyze twitter data and give the meaning of it. T got twitter data using twitter api calls, For that use twitter streaming api which large amount of tweets could be got at a time rather than using other api's.for data saving part mongo dB is used.

The input of this Topic analysis component is a keyword according to the user's choice. Then the system will be analyze that keyword and find tweets which contains that keyword. Then the system analyze that particular tweets and finds out whether it is a positive negative or neutral and also the system detects the false tweets.

When selecting tweets only the English language. This can be done by sending request to the twitter site requesting only English language tweets. Then the system will start to analyze the twitter details. User details, geolocation text etc. when it comes to the text analysis system needs to be identified if the particular tweet is a positive negative or a neutral one. For that we have come up with an algorithm. And this algorithm has been created using NLTK and naïve bays classifier which is already in build algorithm in NLTK.

NLTK is a leading platform for building Python programs to work with human language data which provide many facilities. Text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

The greatest challenge in this component is to find a data set and train that data set.so and accurate data set was found and it has been labeled by an expert weather it is positive negative and neutral. Hence received this data and remove unwanted data and are store them in an arrays.



Good
Awesome
Excellent
love

Positive review Array

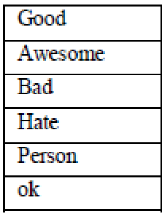
person
mobile
ok
tell

neutral review array

Bad
Hate
Kill
sad

Negative review array

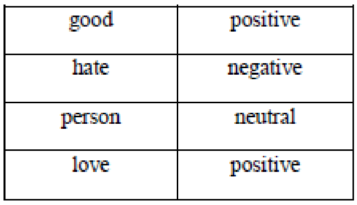
Then, obtain the word from the array and create our own vocabulary. The initial part in this is to match that twitter text with our vocabulary and to find out the matching words and parse them to naïve bays classifier.



Good
Awesome
Bad
Hate
Person
ok

Vocabulary list

Then the word are got from the word in each arrays and a list is created called training _data. Each words is labeled weather it a positive negative or neutral.

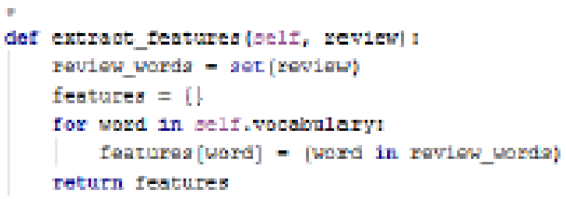


good	positive
hate	negative
person	neutral
love	positive

Now, come to the probability calculating part. For this use naïve bays algorithm. This algorithm is in build functioned in the NLTK.using naïve bays classifier this algorithm can be used.

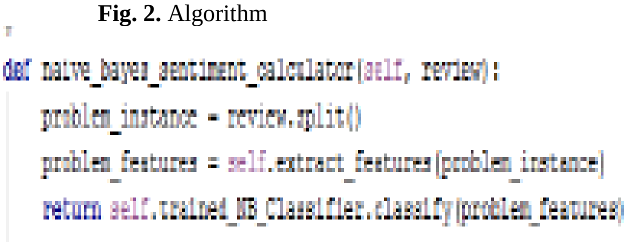
Naïve bays algorithm uses bays theorem, bays theorem stared that "It works on conditional probability. Conditional probability is the probability that something will happen, given that something else has already occurred. Using the conditional probability, calculating the probability of an event using its prior knowledge" .this theorem used in naïve bays classifier. It predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is

considered as the most likely class. Naive Bayes classifier assumes that all the features are unrelated to each other. Presence or absence of a feature does not influence the presence
or absence of any other feature. Using Wikipedia example for explaining the logic
“A fruit may be considered to be an apple if it is red, round, and about 4″ in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.”[5]
This classifier does taking tweets text as an input and compare with the trained data set, then the output will state weather this is text is a positive negative or a neutral one. Here it is shown how weather the particular words occurrence are according to the probability. Trained data set will act as the evidence of a problem. According to this evidence that the output will be given. If it’s possible to lactate a big data set the output will be very much accurate.



```
#
def extract_features(self, review):
    review_words = set(review)
    features = {}
    for word in self.vocabulary:
        features[word] = (word in review_words)
    return features
```

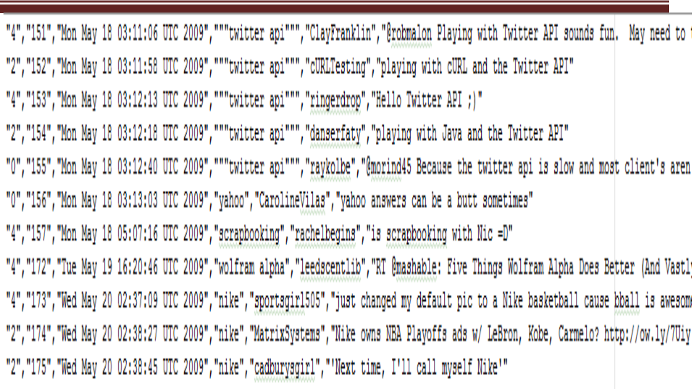
Fig. 2. Algorithm



```
#
def naive_bayes_sentiment_calculator(self, review):
    problem_instance = review.split()
    problem_features = self.extract_features(problem_instance)
    return self.trained_NB_Classifier.classify(problem_features)
```

Fig. 3. Algorithm

In figure 2 the lines of codes performs given text of the tweets words checking with our vocabulary and the matching words will be taken. Figure 3 shows the lines codes is ask to execute figure 2 code and get the output of that code and execute the trained classifier which performed the comparing the matching words with the trained data set. Then the main output will send.
To find out what are the false tweets analyze the tweets user accounts. An assumption is made by us and if the user account criteria is not satisfied then the tweeter user account is not acceptable. User profile will be analyze. Through this user profile users location nationality tweets time, DOB, the initiative data of the profile, text likewise can be obtained. Assume that in case there is any error or does not tally. Then assume that is profile is fake one.
If the user profile is newly created one and some link has been shared from that profile, if the user profile is not active one it means it has posted something before prolonged period, if tweeting pattern does not tally. The user profile user’s slang and user profile is used only to insult someone. If one or more criteria were matched and assume that the user profile is not acceptable.
Emoji Sentiment Ranking v1.0 has been used for emoji sentimental analysis. The system captures the emoji and is passed on to emoji analyzer. The emoji analyzer will checks with the Emoji Sentiment Ranking and give the meaning of that emoji. Then the normal procedure will be continue.
In event analyze system will mainly analyze particular event through our application and can identify which are the time periods users are mainly active in those events. Our system has
capability to analyze tweets based on given event period to forecast and guide user’s behaviors. Real time analysis on the tweet to predict and forecast current affairs, based can get locations and categories.
In event analyze I have used Time series analysis and forecasting methods. In time series analysis I have used ARIMA model to analyze events. ARIMA means Autoregressive Integrated Moving Average (ARIMA) Model.



```
"4","151","Mon May 18 03:11:06 UTC 2009","""twitter api""","ClayFranklin","@robmalon Playing with Twitter API sounds fun.  May need to 
"2","152","Mon May 18 03:11:58 UTC 2009","""twitter api""","cURLTesting","playing with cURL and the Twitter API"
"4","153","Mon May 18 03:12:13 UTC 2009","""twitter api""","ringerdrop","Hello Twitter API ;)"
"2","154","Mon May 18 03:12:18 UTC 2009","""twitter api""","danserfaty","playing with Java and the Twitter API"
"0","155","Mon May 18 03:12:40 UTC 2009","""twitter api""","raykolbe","@morind45 Because the twitter api is slow and most client's aren
"0","156","Mon May 18 03:13:03 UTC 2009","yahoo","CarolineVilas","yahoo answers can be a butt sometimes"
"4","157","Mon May 18 05:07:16 UTC 2009","scrapbooking","rachelbegins","is scrapbooking with Nic =D"
"4","172","Tue May 19 16:20:46 UTC 2009","wolfram alpha","leedscentlib","RT @mashable: Five Things Wolfram Alpha Does Better (And Vastl
"4","173","Wed May 20 02:37:09 UTC 2009","nike","sportsgirl505","just changed my default pic to a Nike basketball cause bball is awesom
"2","174","Wed May 20 02:38:27 UTC 2009","nike","MatrixSystems","Nike owns NBA Playoffs ads w/ LeBron, Kobe, Carmelo? http://ow.ly/7Uiy
"2","175","Wed May 20 02:38:45 UTC 2009","nike","cadburysgirl","'Next time, I'll call myself Nike'"
```

Fig. 4. Trained Data set

ARIMA is a general statistical model which is widely used in the field of time series analysis. General ARIMA model is denoted as the ARIMA(p,d,q) where p,d, and q are non negative integers. In the above notation p parameter basically refers to the autoregressive part ,d parameter refers to integrated part and the last parameter q refers to the moving average part.[8]
Mathematically the pure ARIMA model is written as

$$W_t = \mu + \frac{\theta(B)}{\phi(B)} a_t \quad (1)$$

The series Wt is computed by the IDENTIFY statement and is the series processed by the ESTIMATE statement. Thus,Wt is either the response series *Yt* or a difference of Yt specified by the differencing operators in the IDENTIFY statement.
For simple (nonseasonal) differencing, Wt = (1 - B)dYt . For seasonal differencing
Wt = (1 - B)d(1 - Bs)DYt, where *d* is the degree of nonseasonal differencing, *D* is the degree of seasonal differencing, and *s* is the length of the seasonal cycle.[9]

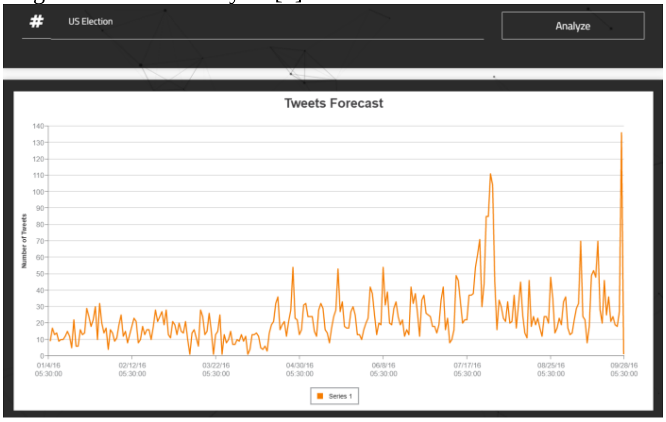


Fig. 5. Results of Event analysis

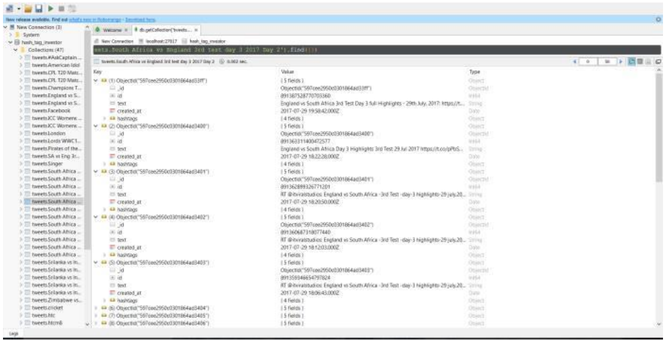


Fig. 6. Dataset of Event analysis

As shown in graphs for the representation of these event twitter data in ARIMA model .As a general data structure, graphs have become increasingly important in modeling event analyze structures and their interactions.[9]

Graph mining techniques and tools are then used to discover the required information for data visualization. Graph mining is the process of gathering and analyzing data represented in graphs.

In account, analyze component users able to identify verified account so they can select correct account. In twitter, there are verified accounts and non-verified accounts and fake accounts as well. In order to select correct account, users able to identify verified accounts easily. Also, users can identify how their followers are geographically represented and what are the sentimental aspects of tweets in accounts.

Users able analyze account so they can examine the account. In order to analyze a specific account, focus on various numbers of criteria. Those criteria which I have taken to analyzing accounts are as below. How Followers Geographically represented and what are the sentimental aspects of the tweets in certain accounts. Also, through data clustering concepts users can get Average tweets per day, Most recent tweets, Top used hash tags, Top user mentions, Most re-tweeted tweets, Calculating Twitter Like rate (Favorite Rate), and Tweet reach percentage – How many of your followers do you reach Impressions by time of day.

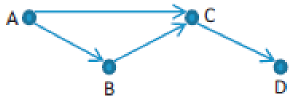
As shown in graph mining techniques and tools are then used to discover the required information for data visualization. Graph mining is the process of gathering and analyzing data represented in graphs.

In Topic comparison system will analyze two topics given by the user and the display popularity of relevant topic according to no of followers, likes, comments and retweets. And also show the popularity of each keywords with geographical view. The input of this module is 2 keywords according to user choice. The system will analyze all tweets which includes the keywords given by the user and view its popularity as a percentage. And also it shows the popularity of the topics given by user within hours as a graph. Collect these data from tweeter I have use sentimental Analysis methods.

Sentiment analysis refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information [1]. To take popularity by the comments analyzed the comments and extract positive comments. Sentiment analysis is common to express this as a classification problem where a given text needs to be labeled as Positive, Negative or Neutral [2].

Retweet process is divide to two parts
-Retweet from original tweet
-Retweet from retweeted tweet

Twitter API only gives retweet from original tweet. Therefore, using a method to get a count of Retweet from Retweeted tweet. To study the popularity of the topic from super nodes, we need to build the retweet propagation topology for a piece of news, where nodes are connected by retweeting paths. However, as mentioned previously, Twitter API does not provide the previous parent retreats of a user's retweeted, so it is difficult to identify the entire retweeting path of a retweet.



For example, in Fig. 2, user D receives the news from user C. User C can receive the news from user A, from B, or from both. Then, it is difficult to determine whether the retweet that user D received from user C is originally from A or B. Thus, it is a challenge to determine where the retweet is from for each retweeted for each tweet in our trace. Then built the retweet propagation topology indirectly with two assumptions without the loss of generality listed below.

1. A user retweets a piece of news only when (s)he sees the tweet at the first time.

2. There is a time delay between when a user sees a piece of tweet and when the tweet was created/published.

The users that a user A follows are called user A's friends. For a given tweet, suppose V is a set of retweet nodes sorted by the retweeting time for a tweet, *v*0 is the source node (i.e., supernode) of the tweet, F*vi* is the set of friends of *vi* and L*vi* is the set of retweet nodes retweeted before node *vi*. t*vi* is the time that *vi* retweeted the news since the publish time t0. Get the values of all the aforementioned parameters from our crawled data. Uses to denote the users' response delay, defined as the time elapsed after a user sees a tweet and before (s)he retweets the tweet. It was assumed that the retweeter of a user is the user's friend, which is true in most cases. In order to find the retweet topology relationship for the topology construction of a given tweet, for each retweeter *v*i for the tweet, to find *v*i's parent retweeter in the topology that retweets to *v*i. Based on the above assumptions, the following algorithm was developed for this purpose:

For each *v*i, if L*v*i F*v*i= ∅, *v*i retweeted the news from *v*0 because none of *v*i's friends retweeted the tweet.

2. If L*v*i F*v*i∅, the subset was sorted L*v*iF*v*i*v*0 by retweeting time and select the node with the latest retweeting time that is smaller than t*v*i− as the parent of node *v*i.

Recall our trace data includes all retweeters for a tweet, by finding each retweeter's parent using this algorithm, we can finally construct the retweet propagation topology of this tweet. In this model, the user's response delay is the main factor that might lead to an imprecise topology since the users' response delay may change in a relatively large range due to various reasons. Since the latest retweeting happens much earlier than their children's retweeting time in most situations, is very relatively small and negligible. Therefore, this algorithm can help find precise retweeting path in most situations.[3]

After visualizing popularity of two keywords given by user system will show a popularity prediction for next hours . For that I used logistic regression Method to show a cumulative growth of prediction with 99% accuracy[4].

An explanation of logistic regression can begin with an explanation of the standard logistic function. The logistic function is useful because it can take any real input t, (t |R) , whereas the output always takes values between zero and one[14] and hence is interpretable as a probability.

The logistic function is defined as follows:

Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary values (0 or 1) rather than a numeric value.

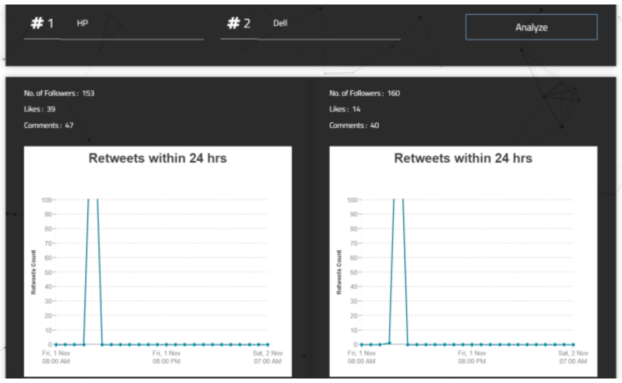
Below is an example logistic regression equation:

y = e^(b0 + b1*x) / (1 + e^(b0 + b1*x))

Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). Each column in your input data has an associated b coefficient (a constant real value) that must be learned from your training data.

The actual representation of the model that you would store in memory or in a file are the coefficients in the equation (the beta value or b's) [4].

Graphs represents all these data and predictions with the help of the above methods. Because, as a general data structure, graphs have become increasingly important in modeling event.


1 HP

2 Dell

No. of Followers : 153
Likes : 39
Comments : 47

No. of Followers : 160
Likes : 14
Comments : 40

Fig. 7. Results of Retweet analysis

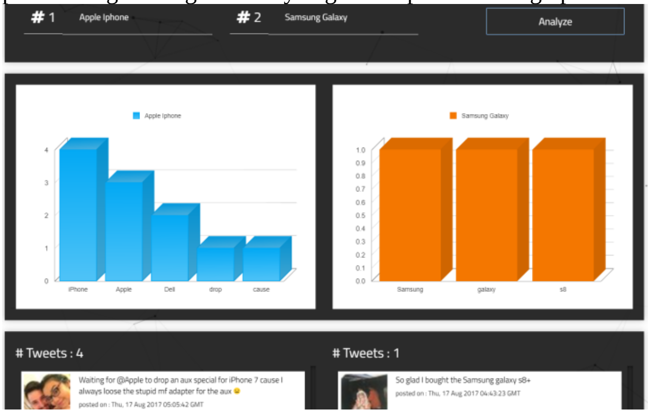
In time analysis system will analyze particular product / products and can identify user preferences on those products. the input of this module is product based keyword according to user choice. analyzing of these data can be done in single topic and two topics. system will generate clusters according to user entered keywords and view the popularity in clusters. Clustering is an important data mining technique employed in dataset exploration where one wishes to partition these datasets into related groups. Among the algorithms that are typically used for clustering, k-means is arguably one of the most widely used and most effective clustering method. In our research project followed k-means algorithm to perform the data analysis part. Other than that followed supervised and unsupervised clustering in this component.

Supervised clustering is applied on classified examples with the objective of identifying clusters that have a higher probability density to a single class. Unsupervised clustering is a learning framework using a specific object functions, for example a function that minimizes the distances inside a cluster to keep the cluster tight as possible.[11]

In supervised clustering, useing labeled data to generate clusters and in unsupervised clustering using unlabeled data. In supervised clustering only focused on two predefined labels. Those are product price and product features.

In addition to that, time evaluation is shown in relation to clusters. Graphs were used for the visualization of these event twitter data in Linear model. As a general data structure, graphs have become increasingly important in modeling event analyze structures and their interactions.[12]

Graph mining techniques and tools are then used to discover the required information for data visualization. Graph mining is the process of gathering and analyzing data represented in graphs.



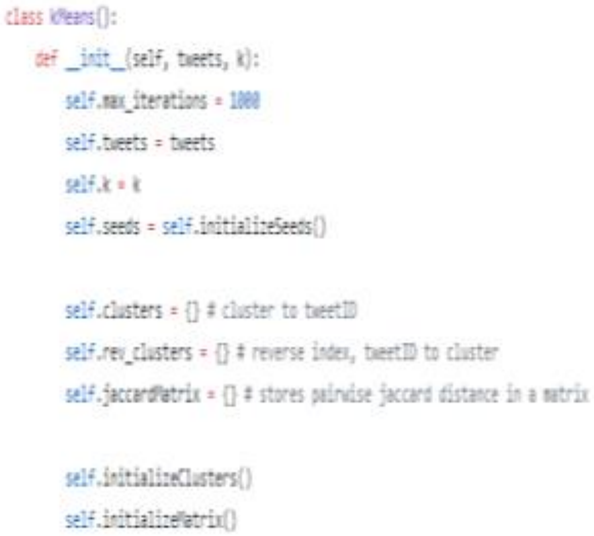
Tweets : 4

Waiting for @Apple to drop an aux special for iPhone 7 cause I always loose the stupid mf adapter for the aux 😑
posted on : Thu, 17 Aug 2017 05:05:42 GMT

Tweets : 1

So glad I bought the Samsung galaxy s8+
posted on : Thu, 17 Aug 2017 04:43:23 GMT

Fig. 8. Results of Time analysis



```
class KMeans():
    def _init_(self, tweets, k):
        self.max_iterations = 1000
        self.tweets = tweets
        self.k = k
        self.seeds = self.initializeSeeds()

        self.clusters = {} # cluster to tweetID
        self.rev_clusters = {} # reverse index, tweetID to cluster
        self.jaccardMatrix = {} # stores pairwise jaccard distance in a matrix

        self.initializeClusters()
        self.initializeMatrix()
```

4.Conclusions

This paper has discussed our experiments on twitter perception analysis show that Tweet keywords features may be useful for perception analysis in the microblogging domain. More research is needed to determine whether the Tweet keywords features are just of good quality for the perception analysis in this domain.

This tool is more realistic and useful than the other available software such as financial market prediction tools. The system itself is a unique approach towards research conducted in order to enhance the experience in Twitter analyses.

Using Mongo DB to collect training data did prove useful, as did using data collected based on positive and negative emoticons. However, which method produces the better

Training data and whether the two sources of training data

are complementary may depend on the type of features used.

Due to too many API requests the response time of this system is too slow. So that in future, developed a new way to do that. In the study only English was used, but trying to expand the system then it will support other languages as well. This system can only analyze some predefined set of events in future to give user authority to give their own events to analyze.

References

[1] Sentiment analysis https://en.wikipedia.org/wiki/Sentiment_analysis [Accessed: Jul. 19, 2017.]

[2] Predicting the Future with Social Media http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5616710&tag =1 .

[3] Analyzing and predicting news popularity on Twitter, B Wu, H Shen. Department of Electrical and Computer Engineering, Clemson University, United States.

[4] Logistic Regression for Machine Learning http://machinelearningmastery.com/logistic-regression-for-machine-learning/

[5]How Naive Bayes classifier algorithm works in machine learning http://courseweb.sliit.lk/pluginfile.php/76746/mod_resource/content /1/eguid.pdf

[6] Natural Language Toolkit . http://www.nltk.org/ . [Accessed: Jul. 26, 2017.]

[7] Stop words with NLTK https://pythonprogramming.net/stop-wordsnltk-tutorial/ [8]time analysis and forecasting algorithms [Online] Available https://www.slideshare.net/tharindurusira/time-series-prediction-algorithms-literature-review

[9]TheARIMAProcedure,http://www.dms.umontreal.ca/~duchesne/ chap7.pdf

[10] Univariate Time Series Analysis; ARIMA Models http://www.econ.ku.dk/metrics/Econometrics2_05_II/Slides/07_univariatetimeseries_2pp.pdf

[11] Supervised k-Means Clustering http://www.cs.cornell.edu/~tomf/publications/supervised_kmeans-08.pdf

[12] python-cluster Documentation. https://media.readthedocs.org/pdf/python-luster/latest/python-cluster.pdf