**Understanding Sentiment Analysis**

UNDERSTANDING SENTIMENT ANALYSIS

It is the process of finding out the sentiment of a particular sentence/paragraph/text.

Examples

– I loved this restaurant's fried rice (positive sentiment)

– This mayor has done a really good job in his tenure (positive sentiment)

– The product I have received is red in colour (neutral sentiment)

– I had a bad experience during this cab ride (negative sentiment).

The data sources for sentiment analysis includes the following:

* Twitter tweets
* Facebooks comments
* Online reviews
* Sms/emails/messages
* Novels, books
* Any other text based sources like books/newspapers/magazines/etc

WHY VISUALIZATION?

Data visualization is the presentation of data in a pictorial or graphical format. It enables decision makers to see analytics presented visually, so they can grasp difficult concepts or identify new patterns. With interactive visualization, you can take the concept a step further by using technology to drill down into charts and graphs for more detail, interactively changing what data you see and how it’s processed. Tables, bar plots, histograms and pie charts can be used for visualization.

WHAT IS TWITTER SENTIMENT ANALYSIS?

Twitter is an online news and social networking service that enables users to send and read short 140-character messages called "tweets". Registered users can read and post tweets, but those who are unregistered can only read them. Twitter Sentiment Analysis is the process of determining the emotional tone behind a series of words, used to gain an understanding of the attitudes, opinions and emotions expressed within an online mention.

WHY TWITTER SENTIMENT ANALYSIS?

The applications for sentiment analysis are endless. ​It is extremely useful in social media monitoring as it allows us to gain an overview of the wider public opinion behind certain topics However, it is also practical for use in business analytics and situations in which text needs to be analysed.

Sentiment analysis is in demand because of its efficiency. Thousands of text documents can be processed for sentiment in seconds, compared to the hours it would take a team of people to manually complete. Because it is so efficient (and accurate – Semantic has 80% accuracy for English content) many businesses are adopting text and sentiment analysis and incorporating it into their processes. Applications: The applications of sentiment analysis are broad and powerful. Shifts in sentiment on social media have been shown to correlate with shifts in the stock market.

For example, the Obama administration used sentiment analysis to gauge public opinion to policy announcements and campaign messages a head of 2012 presidential election.

The ability to quickly understand consumer attitudes and react accordingly is something that Expedia Canada took advantage of when they noticed that there was a steady increase in negative feedback to the music used in one of their television adverts

Tweets are imported using R and the data is cleaned by removing emoticons and URLs. Lexical Analysis is used to predict the sentiment of tweets and subsequently express the opinion graphically through ggplots, histogram, pie chart and tables.

SYSTEM REQUIREMENTS

The following are the major requirements for performing sentiment analysis is mentioned below:

Installation of R

Twitter Authentication to access API

FEATURES

1. Extraction of Tweets

(i) Create twitter application

(ii) twitteR - Provides an interface to the Twitter web API

(iii) ROAuth - ​R Interface For OAuth

(iv)​Create twitter authenticated credential object, It is done using consumer key, consumer secret, access token, access secret.

(v) During authentication, we are redirected to a URL automatically where we click on Authorize app as shown in the image below and enter the unique 7-digit number to get linked to the account from which feeds are being taken.

​2. Cleaning Tweets

The tweets are cleaned in R by removing:

● Extra punctuation

● Stop words (Most commonly used words in a language like​ the, ​is, ​at, which, and, ​on,.)

● Redundant Blank spaces

● Emoticons

● URLS

3. Loading Word Database

A database, created by Hui Lui containing positive and negative words, is loaded into R. This is used for Lexical Analysis, where the words in the tweets are compared with the words in the database and the sentiment is predicted.

For movie tweets, Naive Bayes Machine Learning Algorithm is used. AFINN is a list of English words rated for valence with an integer between minus five (negative) and plus five (positive). The words have been manually labeled by Finn Årup Nielsen in 2009-2011. The file is tab-separated. The version used is: AFINN-111: Newest version with 2477 words and phrases.

4. Algorithms used

● Lexical Analysis:​ ​By comparing uni-grams to the pre-loaded word database, the tweet is assigned sentiment score - positive, negative or neutral and overall score is calculated.

5. Calculating percentage

​I have presented the scores, the tweets as well as the percentage of positive/negative emotion in the text. This calculated using simple arithmetic to understand the overall sentiment in a more better manner

6. Histogram tab : histogram plot

Histograms of positive, negative and overall score are found under the Histogram tab for graphically analyzing the intensity of emotion in the tweeters.

PACKAGES USED

● twitteR​: Provides an interface to the Twitter web API

● stringr​: String operations in R

● ROAuth​: Provides an interface to the OAuth 1.0 specification allowing users to authenticate via OAuth to the server of their choice.

● RCurl​: Provides functions to allow one to compose general HTTP requests and provides convenient functions to fetch URIs, get & post forms, etc. and process the results returned by the Web server.

● ggplot2​: An implementation of the grammar of graphics in R. It combines the advantages of both base and lattice graphics: conditioning and shared axes are handled automatically, and you can still build up a plot step by step from multiple data sources.

● reshape​: Flexibly restructure and aggregate data using just two functions: melt and cast

● tm ​: A framework for text mining applications within R.

● RJSONIO​: This is a package that allows conversion to and from data in Javascript object notation (JSON) format. This allows R objects to be inserted into Javascript code and allows R programmers to read and convert JSON content to R objects

● plyr​: Tools for Splitting, Applying and Combining Data

LIMITATIONS

1. The Twitter Search API can get tweets upto a maximum of 7 days old.

2. Cannot get 100% efficiency in analysing sentiment of tweets.

FUTURE WORK

* Detect sarcasm in tweets
* Analyse images for emotions.
* Add Telugu words to dataset.
* Find no of mentions of n particular organizations.
* Parallelizing code.
* Apply better Machine Learning Algorithms (Like Support Vector Machine).

REFERENCES

● <https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>

● ​ <https://www.quora.com/How-can-I-read-Twitter-data-directly-in-R>

● <https://www.r-bloggers.com/emoticons-decoder-for-social-media-sentimentanalysis-in-r/>

● <https://eight2late.wordpress.com/2015/11/06/a-gentle-introduction-to-naive-bayes-classification-using-r/>

R Code

#Used Libraries

library(RColorBrewer)

library(wordcloud)

library(tm)

library(twitteR)

library(ROAuth)

library(plyr)

library(stringr)

library(base64enc)

library(SnowballC)

library(ggplot2)

library(maps)

#Inserting Access Keys

consumerKey <- "eOtcK9qJpjcObGUPv6SfTFZFE"

consumerSecret <- "TedGta8dDigiQcpKJ68AleIlfaHL7nqdVdCH2qppgD7wLYE0Tm"

accessToken <- "896758586243653632-6RCrl6Yia8uR0YbjBor38RxbjCWCqFm"

accessTokenSecret <- "QYLajXrqdbMYFKwno9itPNo4LVPsVuccOBSyxzfMfdCUF"

requestURL<- "https://api.twitter.com/oauth/request\_token"

accessURL<- "https://api.twitter.com/oauth/access\_token"

authURL<- <https://api.twitter.com/oauth/authorize>

#Setting Up Authentication

setup\_twitter\_oauth(consumerKey,consumerSecret,accessToken,accessTokenSecret)

#Search Twitter

users<- searchTwitteR("@ravithejamuthu", resultType="recent",n=3000, lang="en")

#Converting into DataFrame

tweet.df = do.call("rbind",lapply(users,as.data.frame))

#Plotting data on map

map('world')

points(tweets.df$longitude,tweets.df$latitude, pch=20, cex=1, col="red")

#Viewing the data

View(tweet.df)

#Reading sentiment analysis data from Txt document

pos.words = scan('./positive-words.txt', what='character', comment.char=';')

neg.words = scan('./negative-words.txt', what='character', comment.char=';')

#Appending some more words to actual words

pos.words = c(pos.words, 'new','nice' ,'good', 'horizon')

neg.words = c(neg.words, 'wtf', 'behind','feels', 'ugly', 'back','worse' , 'shitty', 'bad', 'no','freaking','sucks','horrible')

#Converting Into DataFrame

test <-ldply(users,function(t)t$toDataFrame())

#Calculating Result

result <- score.sentiment(test$text,pos.words,neg.words)

#Summarizing Data

summary(result$score)

#Histogram

hist(result$score,col="yellow", main="Score of tweets",ylab=" Count of tweets")

#Count No of Tweets

count(result$score)

#Plotting the tweets on qplot

qplot(result$score,xlab = "Score of tweets")

#Score Sentiment Function

#Used to remove all unwanted data

score.sentiment = function(sentences, pos.words, neg.words, .progress='none')

{

require(plyr)

require(stringr)

scores = laply(sentences, function(sentence, pos.words, neg.words) {

sentence = gsub('[[:punct:]]', '', sentence)

sentence = gsub('[[:cntrl:]]', '', sentence)

sentence = gsub('\\d+', '', sentence)

sentence = tolower(sentence)

word.list = str\_split(sentence, '\\s+')

words = unlist(word.list)

pos.matches = match(words, pos.words)

neg.matches = match(words, neg.words)

pos.matches = !is.na(pos.matches)

neg.matches = !is.na(neg.matches)

score <- sum(pos.matches) - sum(neg.matches)

return(score)

}, pos.words, neg.words, .progress=.progress )

scores.df = data.frame(score=scores, text=sentences)

return(scores.df)

}