Sentiment analysis of Android and iOS using Twitter text mining

Final Project Report

CSC 522 1 | R Language Programming | Fall 2017

Under the guidance of

Dr. Patricia Hoffman

Submitted by,

Sumitha P K

92065

ACKNOWLEDGEMENT

I would like to express my heartfelt appreciation and sincere gratitude to our Professor, Dr. Patricia Hoffman for giving me this opportunity and encouraging my research. I would also like to thank International Technology University for providing me with resources necessary for the research.

Lastly, I would like to thank my family and friends for their continued support throughout the course of my project.

Table of Contents

List of Figures

1 Introduction

Twitter is a popular micro-blogging site where users express their opinions about various topics via tweets.

The aim of my project is to perform a lexicon based sentiment analysis of the two major Mobile Operating Systems – Android and iOS by extracting tweets with hash tags Android and iOS. Sentiment Analysis is the process of computationally determining whether a tweet is positive, negative or neutral.

This project compares the sentiments of iPhone and Samsung Galaxy Series by analyzing their twitter feeds overtime. iPhone is a line of smartphones that run Apple's iOS and the next big player in the market is Samsung Galaxy series of smartphones which use Android OS, produced by Google. The most important decision that we need to make before buying a smartphone is – which operating systems do we want to use. It would be very interesting to compare the sentiments over time of the top mobile Operating Systems and also, the leading smartphones: Samsung Galaxy and iPhone.

2 Problem Definition

The two major Mobile Operating Systems in the Market are iOS by Apple and Android by Google. About 99.6 percent of smartphones use Android or iOS. Apple's iOS are currently included only with iPhone, iPod and the iPad, where as Google allows Android on any number of devices. Both Operating Systems have come a long way since their introductions, but deciding which of the two Mobile Operating Systems is the best can be difficult. This project presents an investigation into the sentiments of Android and iOS and the leading smartphones. A sentiment lexicon is used to extract sentiment scores from twitter feeds.

3 Research Methodology

This project performs sentiment analysis of each OS and leading smartphones by extracting as many tweets as possible and using sentiment lexicons and suitable functions to calculate their sentiment score. Sentiment lexicons are lookup tables or dictionaries that map words to sentiment scores. The results are analyzed statistically by visualizing the results using bar graphs/histograms. The major functions used in this investigation are as follows.

The Source code, README.md and other related files are available in the following link:

https://github.com/Sumitha123/Sentiment-analysis-of-Android-and-iOS-using-Twitter-text-mining

3.1 Twitter App Authentication

We should install packages "twitter" and "RCurl" and load them into the session. Twitter API requires authentication to retrieve tweets. Go to the url https://apps.twitter.com/,

sign in and create a new app. To set up the connection, go to Keys and Access tokens tab and copy api_key, api_secret, access_token and access_token_secret. To set up twitter handshake authorization, we use the function:

setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)

Next, we set SSL certificates globally by specifying RCurlOptions and then initialize Twitter handshake by calling the function: twitCred$handshake().

3.2 Search Tweets

searchTwitter function is used to search tweets based on the search string.

tweet <- searchTwitter(searchterm, n, lang = 'en', resultType = 'recent')

searchTerm is the search string for which tweets should be extracted.

n is the maximum number of tweets to be extracted. 'en' restricts tweets to English language.

3.3 Twitter text processing

Step 1: Extract the text content of all the tweets using sapply() function:

sapply(tweet, function(x) x$getText())

Step 2: Using R Base function iConv(), we convert character string encoding from the locale dependent encoding to UTF-8.

Step 3: The output character vector is given as input to a function clean.text(), which cleans up sentences with R's regex-driven global substitute, gsub().

Step 4: The data is converted into a corpus. Corpus handling and pre-processing requires text-mining package – "tm".

Step 5: After that, the corpus needs few transformations, including changing letters to lower case, removing punctuations/numbers, stripping white spaces and removing stop words.

Step 6: In several cases, words need to be stemmed to retrieve their radicals. For instance, "iPhone" and "iPhones" are both stemmed to "iPhon".

Step 7: Then we build a term document matrix from the corpus using the function: TermDocumentMatrix(myCorpus).

Step 8: Obtain words and their frequencies.

3.4 Create simple wordclouds

A wordcloud is one of the best tools that allows us to visualize most of the words and terms contained in tweets. Creating a wordcloud requires the wordcloud and RColorbrewer package. Following steps are followed to construct the wordcloud:

Step 1: Search tweets with the required search term.

Step 2: The text is cleaned and formatted for further processing.

Step 3: Create a lexical Corpus and a TermDocumentMatrix using tm package.

Step 4: Obtain words and their frequencies.

Step 5: Visually attractive Wordcloud can be constructed using words and their frequencies obtained in above step.

Step 6: Usage of wordcloud function:

```
wordcloud(words = d$word, freq = d$freq, min.freq = 10,
        max.words=200, scale = c(10,0.5), random.order=FALSE,
        colors=brewer.pal(8, "Dark2"), random.color= TRUE)
```

3.5 Create a barplot of frequent terms

Next, we create barplot of frequently used terms in the tweets related to each search term. This gives us an understanding of user sentiments and the words with highest frequencies being used for the corresponding terms. We use barplot() function to create the barplots.

3.6 Calculate Sentiment Score

We calculate the sentiment score using score.sentiment() function. It requires plyr and stringr packages.

Words with positive and negative sentiments are listed in positiveWords.txt and negativeWords.txt respectively. The input to the function is a vector of sentences.

This function implements a very simple algorithm to estimate sentiment, assigning an integer score by subtracting the number of occurrences of negative words from that of positive words. plyr will handle a list or a vector as an "l" for us. Since we want a simple array of scores back, so we use "l" + "a" + "ply" = laply. Next, we compare our words to the dictionaries of positive & negative terms. The function match() returns the position of the matched term or NA. We obtain true/false according to the match which is treated as 1/0 by sum() function. Score is computed as sum of negative matches subtracted from that of positive matches. The function returns a dataframe of text and corresponding score.

Based on the score obtained, we calculate the positive, neutral and negative sentiment scores for the input search term.

3.7 Create ggplot2 barplots based on the sentiment score

We create a barplot for iOS and Android OS by filetring tweets with hashtags iOS and Android. This helps us to analyze the positive, negative and neutral sentiment score for

both the Mobile Operating Systems. Next, we create stacked barplots of all the iPhone series overtime and compare how and why the sentiments have changed over time. Also, we compute the sentiment scores of Samsung Galaxy smartphone series and investigate them. This project analyzes the iPhones: iphone4,4S,5,5S, 5C,6,6S,6SPlus, 7,7plus,8 ,8plus,x and Samsung galaxy series: galaxy s4, s5, s6, s6, s6edge, s7, s7edge, s8, s8plus.

4 Experimental Results

4.1 Wordclouds

Figure 1. iOS Wordcloud

The above wordcloud shows that app, iPad, iPhone, game, features, Apple are the most important words in the tweets with hashtag iOS. The popularity of the word “game” shows Apple's iOS is a popular platform for great mobile games across every genre. “Feature” word stresses that iOS is also popular because of all the unique features that it has to offer. The appearance of “android ” word showcases the major competitor of iOS.

Figure 2. Android Wordcloud

The most important words in Android's wordcloud are app, news, linux, Samsung, google, smartphone and mobile.The appearance of app word signifies the popularity of apps based on android platform. Samsung is the leading market player in Android based samrtphones, hence is prominent in the wordcloud. Linux also appears in wordcloud indicating that it is a linux based operating system.

Figure 3. iPhone8 Wordcloud

The most popular words in iPhone8 wordcloud are switch, price, unlimited, iphone plus and worksform. The word “switch” is popular as many companies such as AT&T, T Mobile provided deals on iPhone8 if customers chose to switch carriers. Sprint, meanwhile, offered 50% off its lease rate for people who switch to Sprint by trading in an iPhone 7 or 7 Plus. The price of iPhone 8 has also jumped higher when compared to the previous tweets, and thus was widely tweeted about. “Worksforme” was also very popular in the tweets. The Wordcloud shows “Worksform” as the tweets are stemmed to extract the radical words.

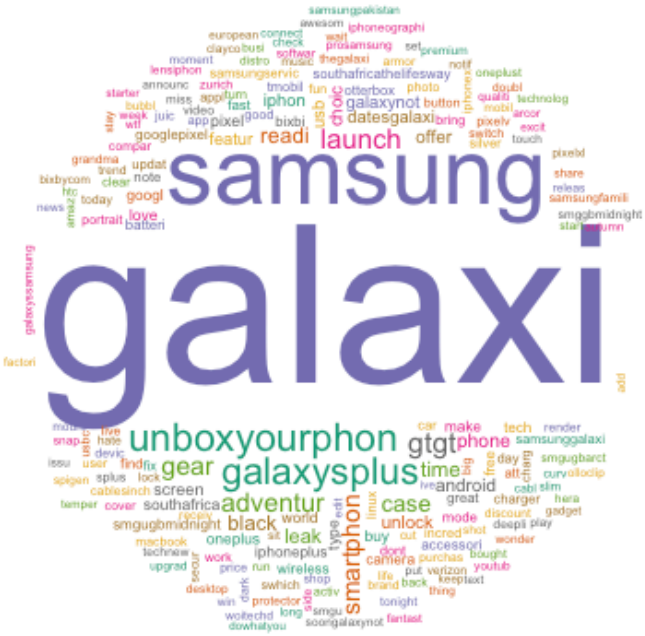

Figure 4. Samsung Galaxy S8 Wordcloud

The most important words in Samsung Galaxy S8's tweets are galaxy, Samsung, unboxyourphone, smartphone, galaxysplus and launch. The word unboxyourphone was popular as Samsung Galaxy S8's teaser video was named "Unbox your phone".

Thus, all the wordclouds show the prominent words in the tweets of iOS, Android and the leading smartphones based on these operating systems. Some of the words in the wordcloud are trimmed as the corpus is stemmed to retrieve the radicals. Wordclouds help us to visually analyze the poplar sentiments with respect to each search term.

4.2 Frequency Barplots

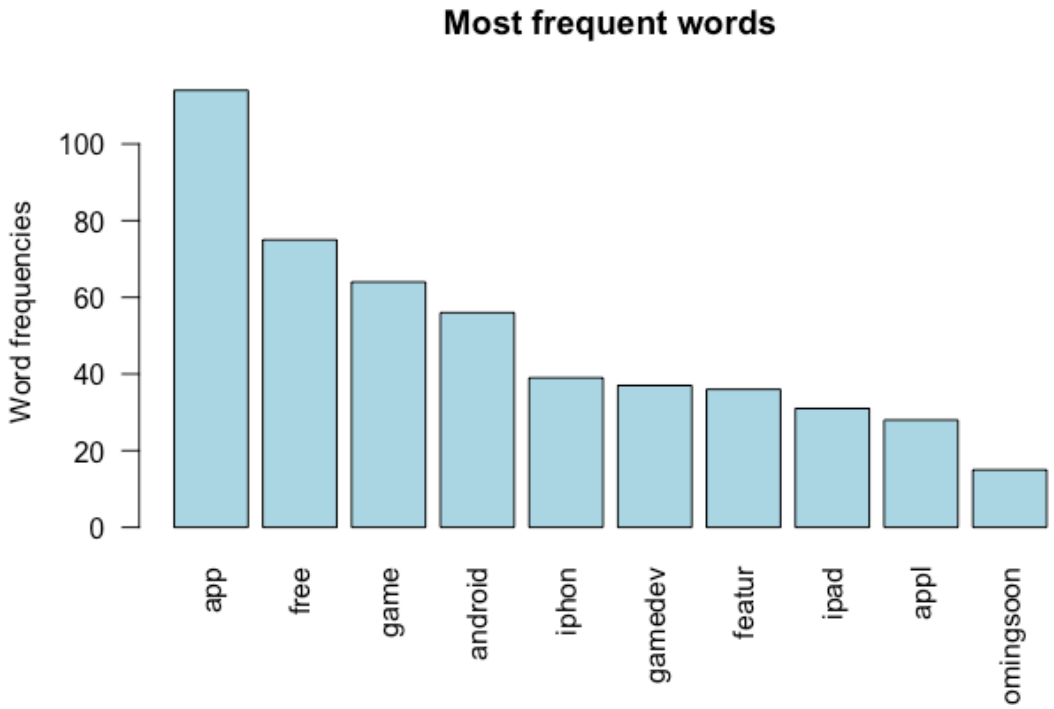

Figure 5. Most frequent words of iOS tweets

The most frequent words in iOS related tweets in decreasing order are app, free, game, android, iPhone, game development, features, iPad, Apple and Coming soon.

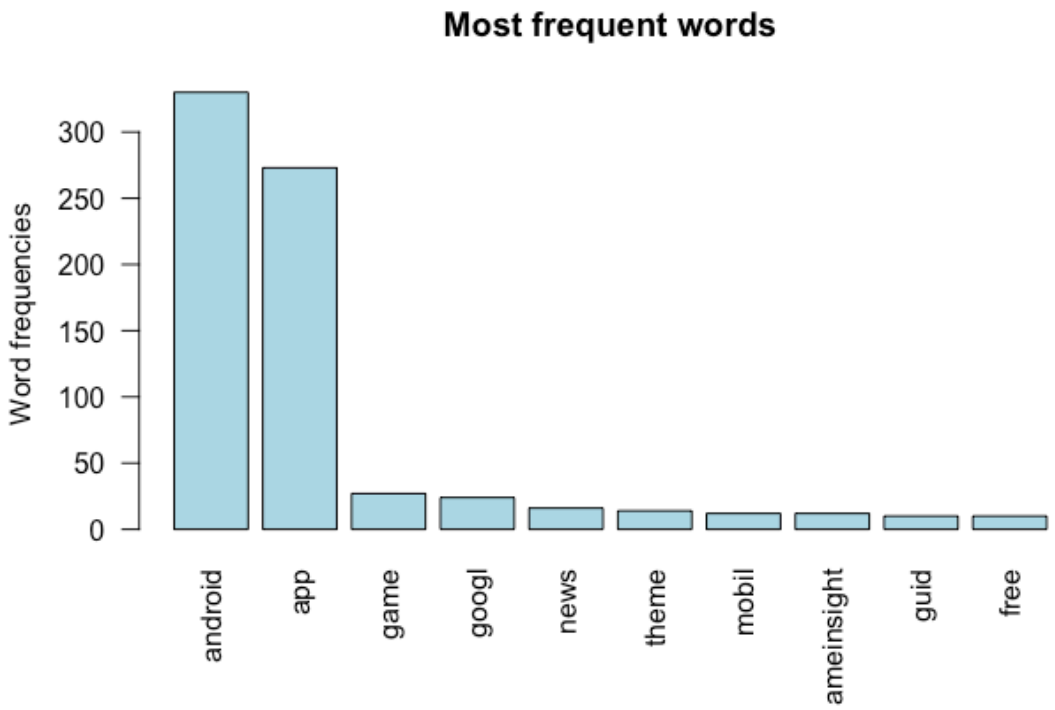

Figure 6. Most frequent words of Android tweets

The most frequent words in Android related tweets in decreasing order are android, app, game, google, news, theme, mobile, gameinsight, guide and free.

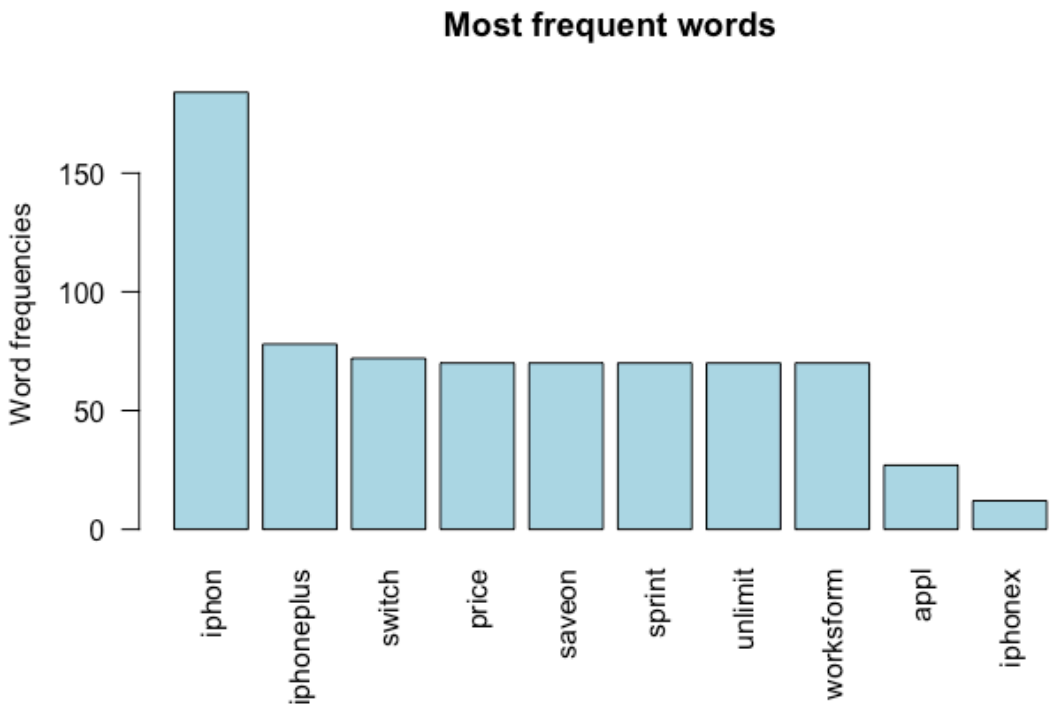

Figure 7. Most frequent words of iPhone 8 tweets

The most frequent words in iPhone 8 related tweets in decreasing order are iphone, iphoneplus, switch, price, saveon, sprint, unlimited, works for me, apple and iPhone X.

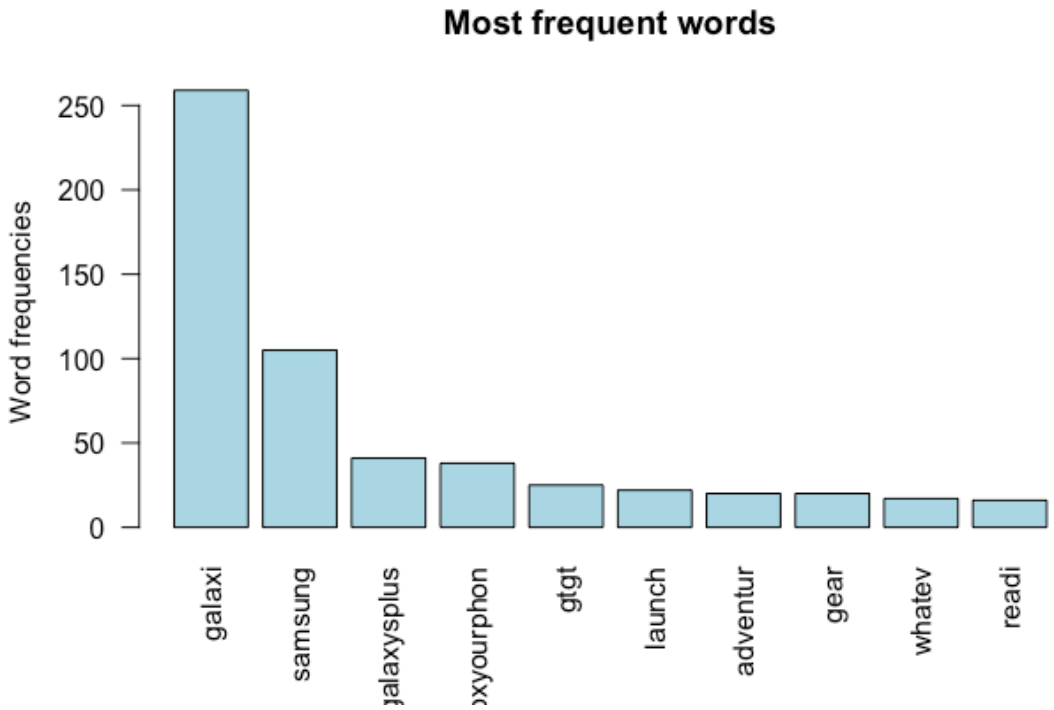

Figure 8. Most frequent words of Samsung Galaxy 8 tweets

The most frequent words in iPhone 8 related tweets in decreasing order are galaxy, Samsung, galaxysplus, unboxyourphone, gt, launch, adventure etc.

4.3 Sentiment score plot of iOS and Android

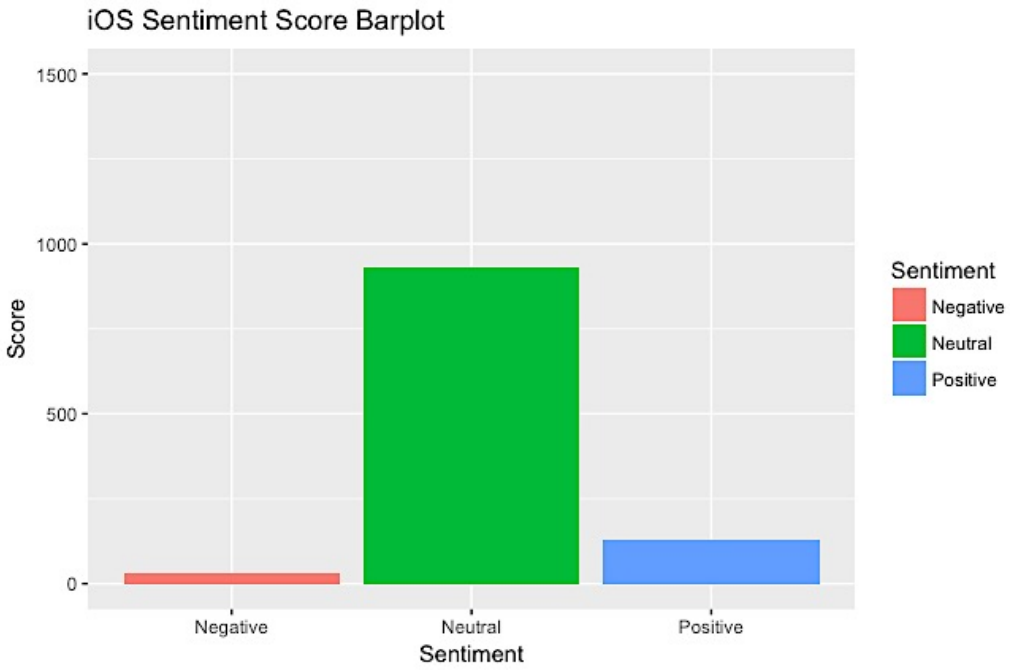

Figure 9. iOS Sentiment Score Barplot

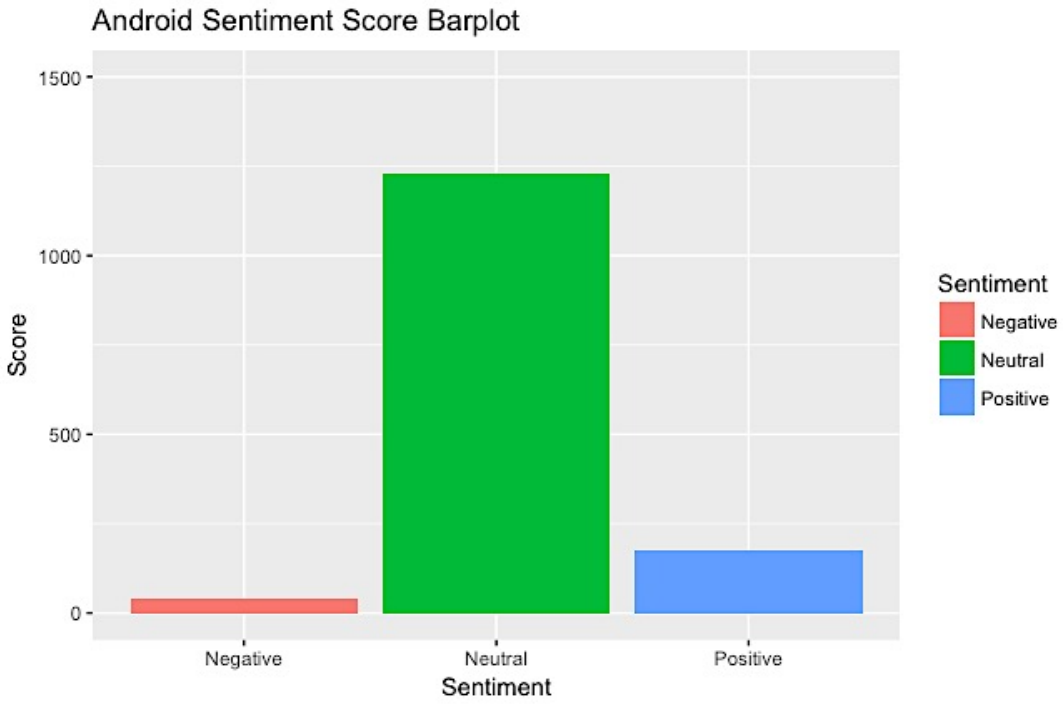

Figure 10. Android Sentiment Score Barplot

Sentiment	Count for iOS	Count for Android
Negative	32	42
Neutral	930	1227
Positive	126	175

Table 1. Comparison of sentiment score of iOS and Android

As we can observe from the above plots and table, the overall positive and neutral sentiment of Android is more than that of iOS for the recent 1000 tweets that were analyzed. One of the reasons for this could be that Android would let us do so much more with our phone than iOS does. Another advantage of Android is that it offers an open platform with tons of possibilities on Google Playstore. The negative sentiment of Android exceeds that of iOS and one of the main reasons for this observation is user-interface. Despite slower processors on Apple hardware, user interface of iOS feels faster and more fluid to most users.

4.4 Comparative analysis of iOS and Android based Smartphone series

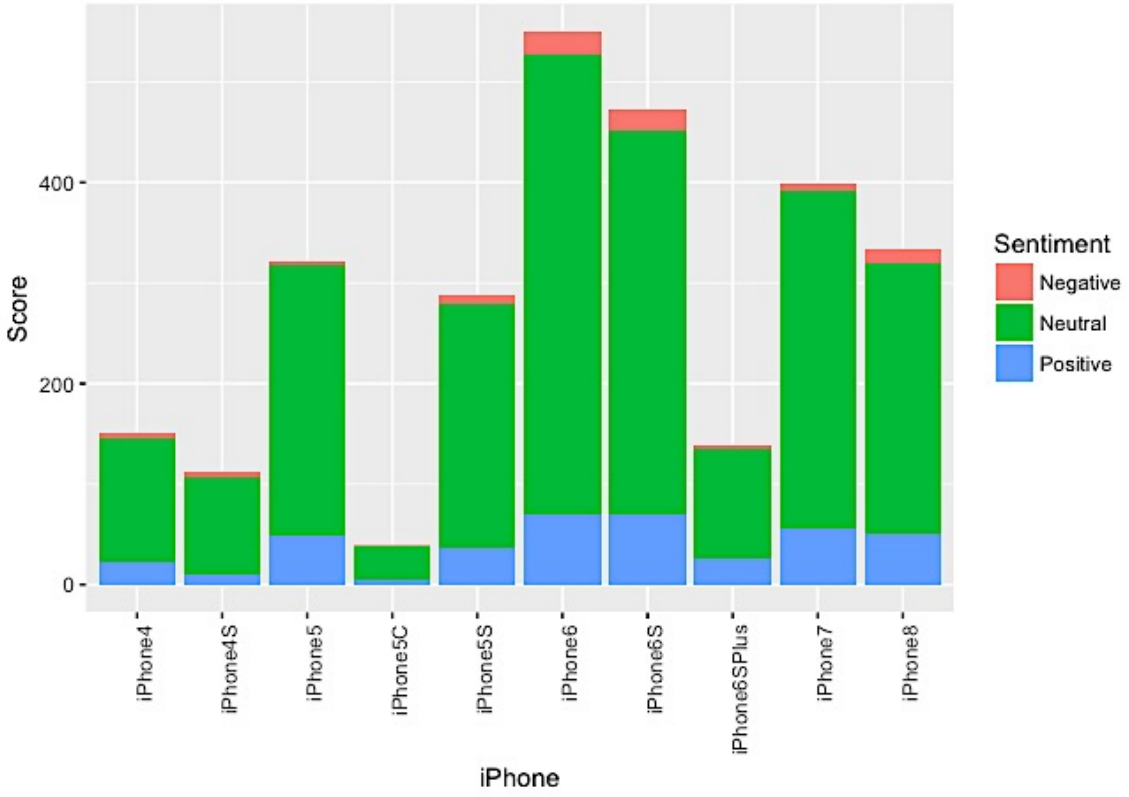

Figure 11. Sentiment Plot of iPhone series

The above plot shows the sentiments of iPhone series has changed overtime.

The iPhone4 shows good overall user sentiments as it came with a new, flat design with an integrated antenna, a high-resolution Retina display and a superior 5MP camera with HD video recording. It had internal performance improvements over iPhone 3G (third generation). The user sentiment dipped slightly for iPhone 4s as it was more expensive than the previous release and absence of large screen, not much difference in design over iPhone 4 being the other reasons. The sentiment for iPhone 5 increased as it was 20% lighter than its predecessor with a 4-inch screen, running at 640 x 1136 pixels. The next step was the evolution of iPhone 5C which was slightly cheaper and plastic backed model made to compete with Android phones in the market. The bigger changes arrived with

iOS 7, the most radical revamp of the mobile operating system since the App Store arrived back in 2008. The user sentiment increased greatly for iPhone 6 which was the revolutionary best phone based on iOS. It had some really great features such as larger screens, dual domain pixels, VoLTE, Camera HDR etc. The overall user sentiment dipped for iPhone7. Also, iPhone 8 does not seem to be as popular as iPhone 7 and this could be due to the lack of any significant enhancements in the new phone.

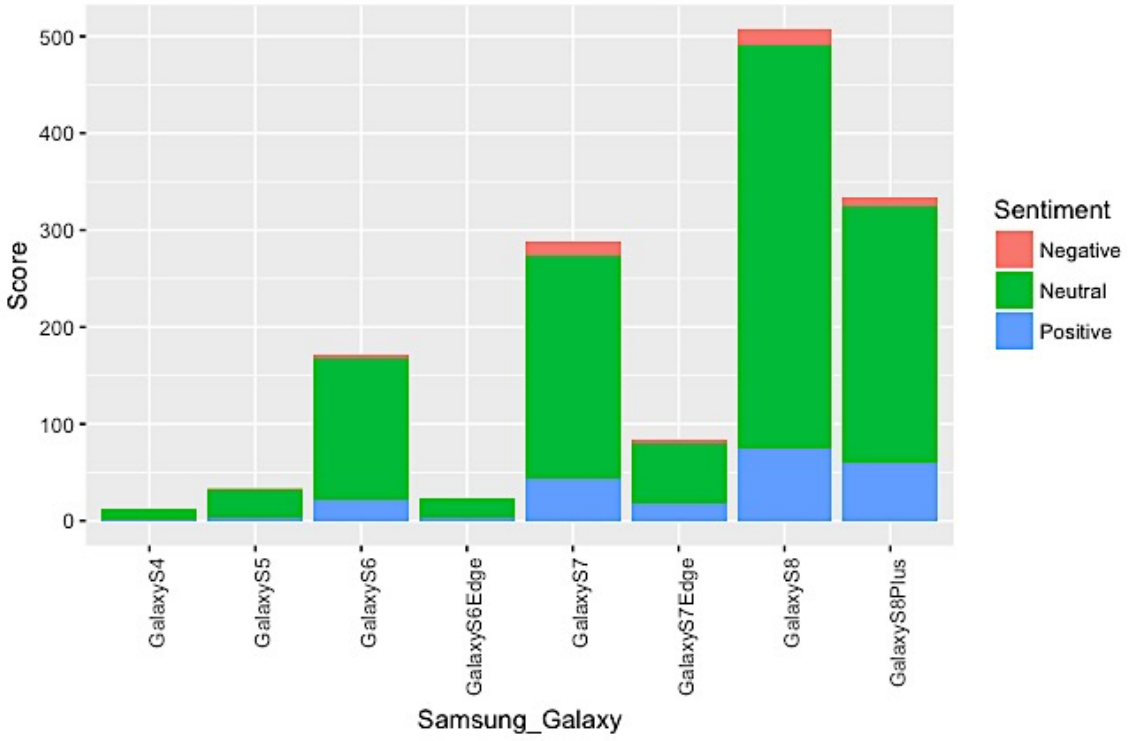

Figure 12. Sentiment plot of Samsung Galaxy Series

The above plot shows that the sentiment of Android OS based Samsung Galaxy S series phones which are the major competitor to iPhones. The populatiry of Samsung Galaxy S series has increased over the time because of the multiple UI improvements and deeper hardware software integration that Android has done in each of their new releases. If you

see the graph and compare it with iPhones, after the launch of iPhone 6 which had the first fingerprint sensor in phones there was not much new technological advancement that Apple could get into their phones. Android phones on the other hand evolved hardware wise too with each release the hardware getting better and the coexistence with the OS and hardware syncing deeper. One other example is unlocking using, voice recognition, retina scanners, dual camera setups etc. This graph clearly shows that Galaxy series phones are getting better with each year with increasing popularity.

5 Conclusions

As we know iPhones have captured the market with the sheer numbers that they have sold. But in the past couple of years we can see that Android phones are gaining more popularity because of tremendous hardware and software enhancements. Also, unlike Apple's iPhone, Android phones are made by many companies and are more affordable.

6 Scope for future work

This work can also be extended to other areas as well for user sentiment comparison and analysis. The best example being tracking the Stock market trends, where by tracking the user sentiments we can see what made the market to go up and down for each of the stocks and then also linking it to the news that those companies made during those particular times.

6 Related Work

There has been several research in lexicon based sentiment analysis by twitter text mining.

In the paper “Discovering Consumer Insight from Twitter via Sentiment Analysis”[1], by Chamlertwat et. al proposed an approach based on sentiment analysis to determine whether a micro-blog post is a positive or negative sentiment. They used Twitter to represent the micro-blog service. Both machine learning based and lexicon-based approaches were applied in their solution.

The research work “Lexicon-Based Methods for Sentiment Analysis”[2], by Taboada et al, presents a lexicon-based approach to extracting sentiment from text. The Semantic Orientation CALculator (SO-CAL) uses dictionaries of words annotated with their semantic orientation (polarity and strength), and incorporates intensification and negation. SO-CAL is applied to the polarity classification task, the process of assigning a positive or negative label to a text that captures the text’s opinion towards its main subject matter.

7 References

[1] Wilas Chamlertwat, Pattarasinee Bhattarakosol, Tippakorn Rungkasiri, “Discovering Consumer Insight from Twitter via Sentiment Analysis”, Journal of Universal Computer Science, vol. 18, no. 8 (2012)

[2] Taboada M, Brooke J, Tofiloski M, Voll K and Stede M, "Lexicon based methods for sentiment analysis",Computational Linguistics,Vol.37,Issue 2,June 2011,p.267-307.

[3] Sarlan A, Nadam C,Basri C, "Twitter Sentiment Analysis", in the proceedings of International Conference on Information Technology and Multimedia (ICIMU), 2014.

[4] Gokulakrishnan B et al., "Opinion mining and sentiment analysis on a twitter data", in the proceedings of International Conference on Advances in ICT for Emerging Regions (ICTer), 2012.