John Deere Tweet Analysis Programming in R

Lee Goodlove, Michael Coyle, Matt Knabel,
Austin Cappaert & Reed Harris

The Initiation

Social media allows for people and businesses around the globe to communicate freely and instantly. When running a major corporation or a small business it is important to understand what information is being shared about the business. For this analysis, the project team has set out to understand what people are saying about our employer, John Deere, on one of the most popular social media platforms, Twitter. We created a simple app that uses Twitter's APIs which call John Deere's twitter mentions through a R programming package known as rtweet (for detailed information on rtweet please visit the project's website.) In order to complete the analysis, a series of steps had to be completed to clean the data so the analysis could take place. The project team has completed a deep dive into what words and types of words are most commonly associated with John Deere, what hashtags are used when with John Deere, what language are used, and when do people most commonly share John Deere information.

Investigation

Data Source

The source of our data came from Twitter, utilizing one of the free API's the platform provides. This requires utilizing an active Twitter account, registering as a developer and creating an App. Having created the app you can then generate a key and secret key allowing you to authenticate with Twitter through an OAuth Token.

We have three main datasets we created for the project - deerestats.csv (containing Tweets from September 9th to the 22nd) and newer dataset deerestatsNew.csv (containing Tweets from October

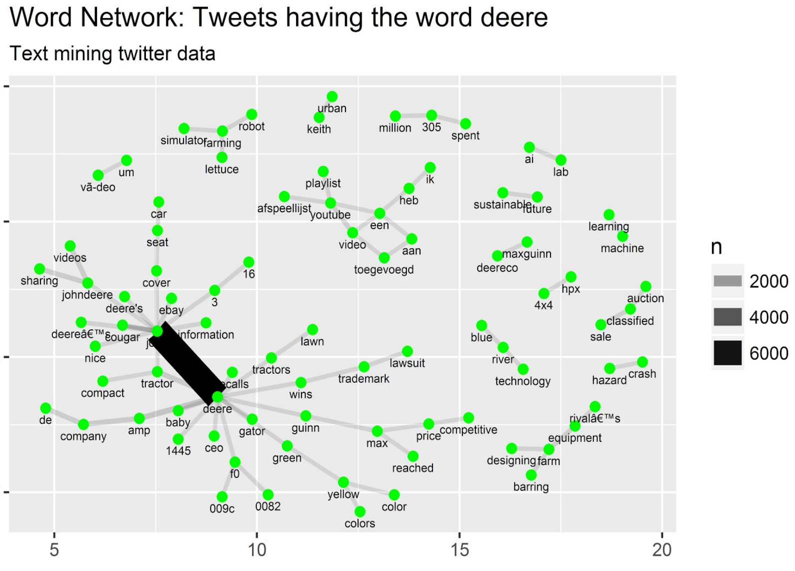
15th to the 24th) and deerestatM.csv combines both the former. The combined dataset originally had over 11,000 tweets, before data cleansing and transformation.

Transformation

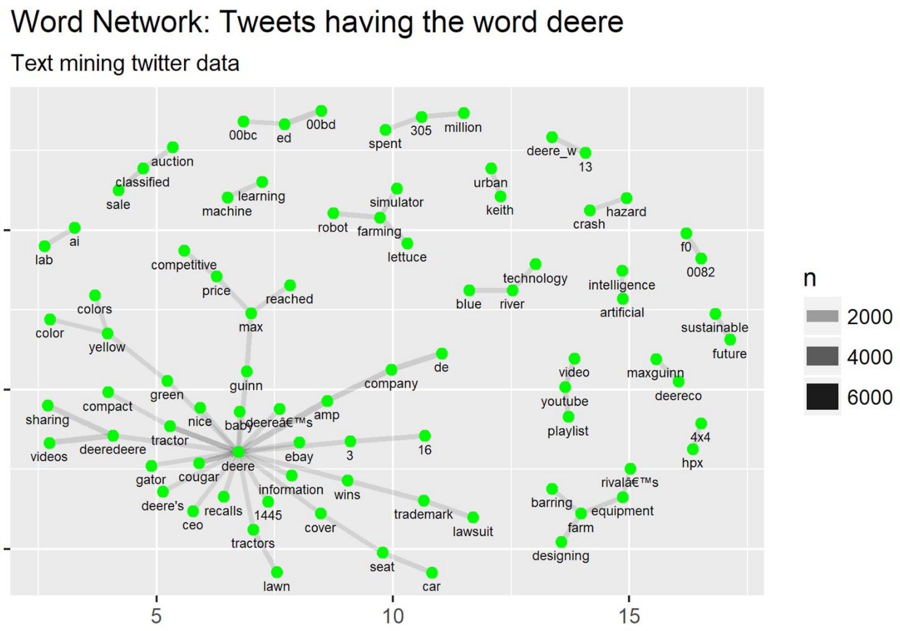
Transformation	Code
stripped hyperlinks	gsub
assign words IDs per association	dplyr
subsetted	subset
flatten	unlist
remove stop words (extremely common words such as “the”, “of”, “to”, and so forth in English)	data(“stop_words) - anti_join
string split	strsplit
rtweet - API calls, JSON to data frame conversion	search_tweets
Language Translation	translateR & Google Translate API

Research & Insight

Word Network



A word network is a diagram that looks at how unique words are being used in sequence of one another. In order to get the data into a proper format a lot of cleansing was required. The first step is to use the dplyr package and unnest so each word is its own row in a new data frame. The rows are sequences in order of the original construct of the tweet. The row indices use a decimal format to identify where the unique words exists in the original data frame and where the word exists in the original tweet string (i.e. The 5th word of the string in row 8 would be numbered 8.5). The next step is to remove any punctuation and move all text to lowercase. We are not interested in looking at how different punctuations are shared with this data and we do not want the analysis to be case sensitive. The next step is to pair the consecutive unique words together using the separate command in the tidyr package. This creates a second column in the new data frame where the unique word that is listed below the row 1.1 is inserted into column 2 of row 1.1. This pattern is continued throughout the data frame. The data is now ready to create the word network.

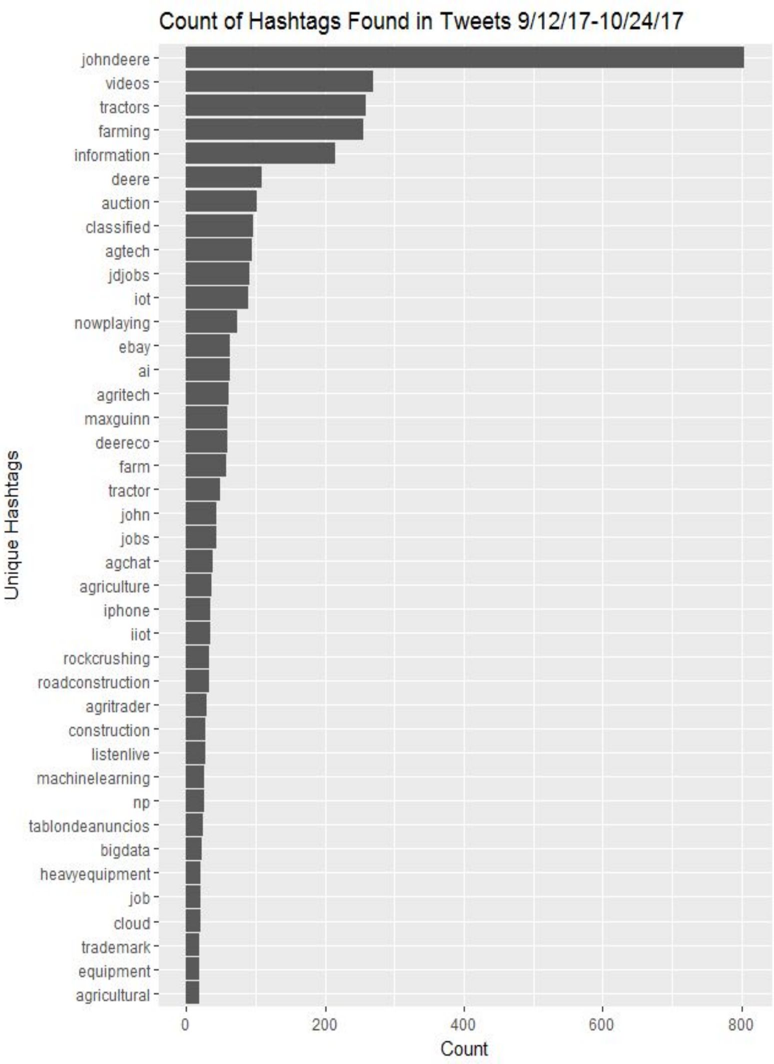


There is a thick black line between the words John and Deere. This is an obvious connection and one that we do not need to see in our analysis. We then converted all mentions of the word 'John' to 'Deere' and create a new plot.

From this word network we can quickly identify what is being said about John Deere. Reading the text strings in the word web quickly tells many stories of what information is being shared about John Deere. Some stories that can you can find in the network are: 1) "Lawsuit" "Trademark" "Wins" –Deere recently won trademark court case about green tractors. 2) Blue" "River" "Technology" – Blue River is a tech company that Deere recently purchased 3) "Cougar" "3" "16" – there is a popular Keith Urban song called: "John Cougar, John Deere, John 3:16". (Note "ebay" looks like it is associated with these words but it is not, there are two lines generated.) This report can be very helpful to the John Deere Public Relations team as they can run this analysis at any time and get a live update on what information is being shared about John Deere.

Hashtags

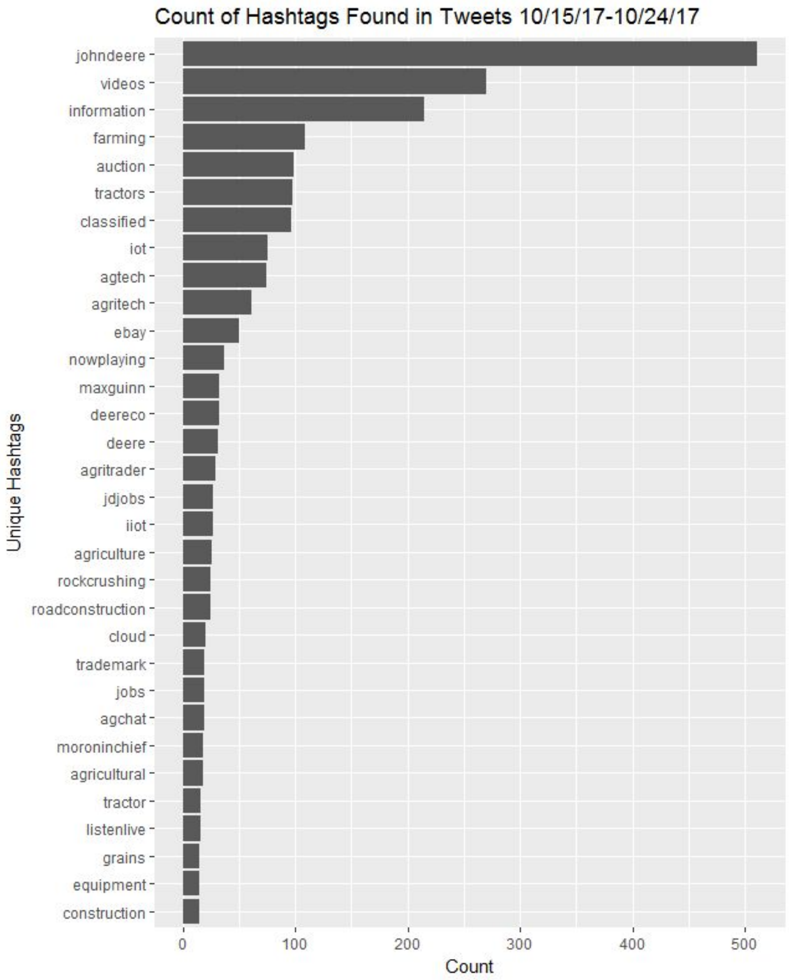
According to www.hashtags.org, for a hashtag to be successful it has to be have the following elements: catchy, short/concise, clear and relatable. By use of a hashtag you can understand what a person is thinking/feeling or want to express in just a couple of words and sometimes just one word. Since our data is from Twitter, let's investigate what we can conclude from the many hashtags embedded. Before starting out, there were a couple of questions that should be attempted to answer.



- What hashtag was used the most?
- What were the top hashtags used?
- Could sense be made of the hashtags?
- Could an event be identified through the use of the hashtags?

The data was completely raw from the public API and needed some cleaning/transformation. There were close to 6,200 unique hashtags. The bar graph below represents the combined dataset (deerestatM.csv). The first assumption was John Deere as a company has a twitter account and actively tweets so the expectation that the highest hashtags should be belong to John Deere. After looking at the data and different tweets. Anyone could use *#johndeere*, the official JD twitter page, JD dealers, JD equipment owners even to people who bought a JD hat used that hashtag. Thus explains the large number of times the #johndeere being used.

The second highest hashtag was #videos which in almost every case used the #information as well.

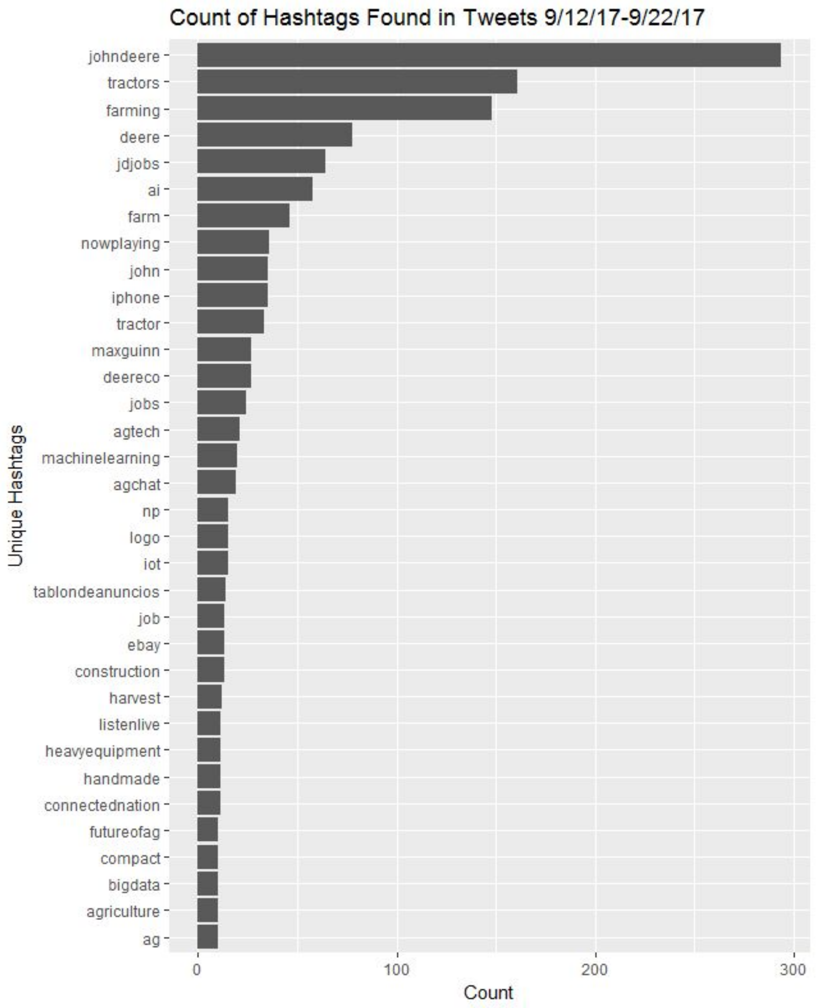


These hashtags were associated with people tagging product, personal and random videos of JD equipment.

The third highest hashtag used was #tractors and it as well was tied to *#farming*. The interesting conclusion here is that these hashtags were used when selling JD equipment via Ebay.com. Farming was the category and tractors was the subcategory. Other hashtags used in the selling/buying of equipment was *#auction* and *#classified* via equipmentauction.com.

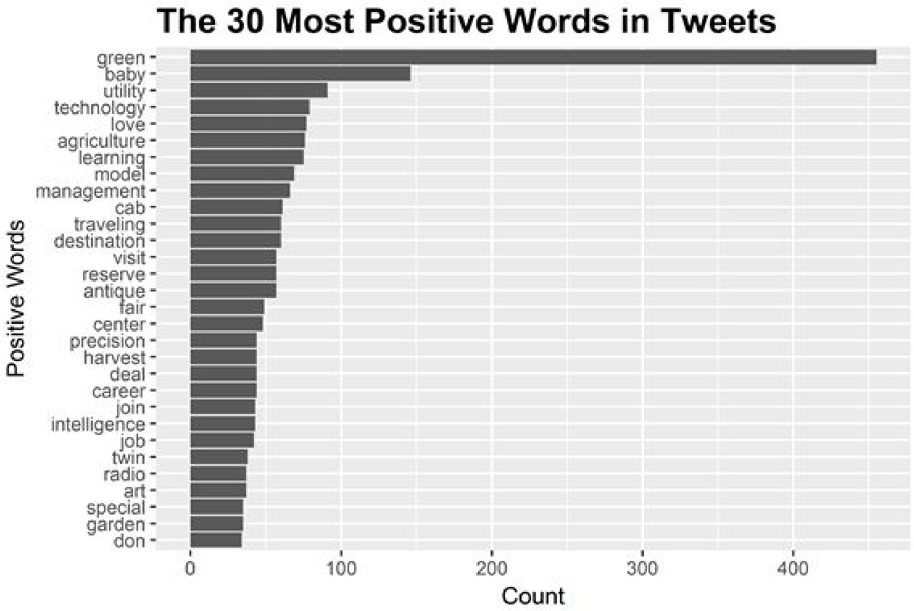
Below are the charts of the individual datasets. Using these charts we were able to tie the usage of a hashtag to an event that occurred involving Deere & Co. For example, an internal event would be John Deere's recruiting season and the month of September is prime time for those activities, *#jdjobs* is the hashtag used within Deere and on campuses abroad for any recruiting efforts. It is even used on JD’s recruiting shirts. This could explain the high number of *#jdjobs* hashtags used in September with a decline in October.

An example of an external event would be John Deere’s acquisition of Blue River Technology. Hashtags *#ai*, *#agritech*, *#agtech*, *#deereco* and *#machinelearning* contributed to the high number of tweets from this business activity. Other hashtags associated with this purchase are *#iot* and *#iiot*. From researching the different hashtag data, John Deere has a diverse population of tweets from financial gurus, other companies (big and small), orators/speakers to the people that matter the most which are our customers.



Sentiment

While researching text mining we discovered that analysis of the sentiment in text can be very telling. The examples in our research focused on sentiment changes in novels from the beginning of the story to the end. We thought we could gain some insight into the sentiment in tweets that reference John Deere. We compared the words in the tweets with the nrc sentiment lexicon dataset in the tidytext R package and plotted the results. The nrc lexicon categorizes English words into categories of positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise and trust. The sentiment analysis of



the Deere tweets were compared to positive, joy, trust, negative, anger and disgust words in the nrc lexicon dataset. The results showed more positive sentiment in the tweets than negative sentiment. The greatest number of negative words in the tweets were found when comparing to the anger words in the nrc lexicon. Words lawsuit and court were the most prevalent anger words in the tweets. This can be attributed to the recent lawsuit that John Deere won against a South Dakota agricultural sprayer manufacturer that was painting their sprayers green and yellow, a John Deere trademark color combination. Sentiment analysis could be used by Deere marketing to determine the feelings people have about John Deere and their products to help them tailor the focus of marketing strategies.

Tweet Time Analysis

The tweet data contains a column that is a date and time stamp indicating when each tweet was created. We grouped the tweets in 4 hour time blocks and plotted the count of tweets created in those blocks. We expected there would be more tweets in the middle of the day and in the evening. Our research has shown that people use Twitter most often from noon to 1:00 PM. Our data aligns with this. John Deere could use this information to help them determine the best times of the day to advertise.

We also used the date time stamp data to plot the days of the week when people tweet most about John Deere. We were surprised to see that Wednesday is the day with the greatest number of tweets.

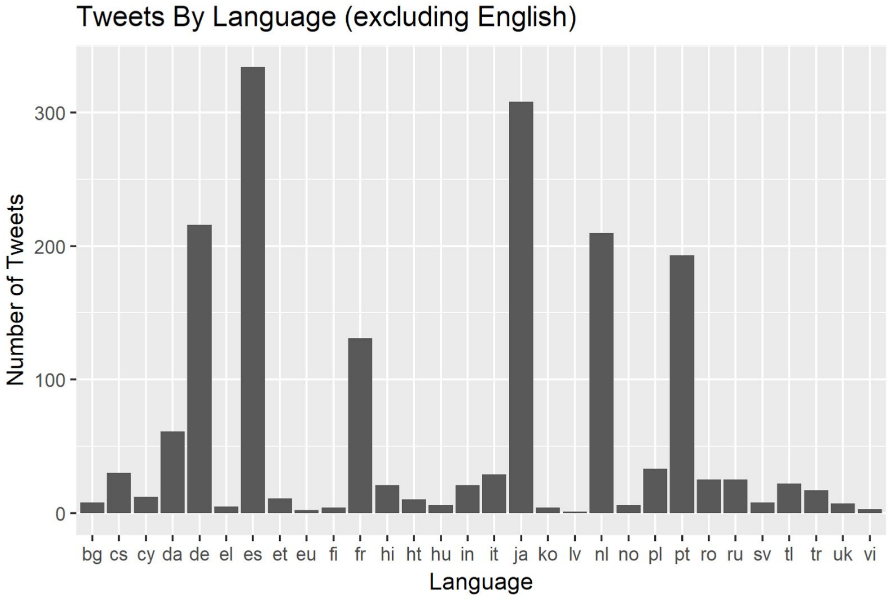
We expected most tweets to occur on the weekends. We researched this and found that Wednesday is the most common day of the week that people tweet. We wrote a function that takes a date range from the user to plot this data. Users could use this function to see trends in the days of a week that people tweet about John Deere.

Tweet Source Analysis

The source of tweets is one of the columns of the Deere tweet data, e.g. Twitter iPhone app, Twitter Android app, Twitter web page etc. There are approximately 400 sources that tweets came from in our dataset. We counted each of the tweet sources and plotted the top 30. This data could be used by John Deere to target how and where to advertise.

Translation

While reviewing the data we found that we could gain additional insight by translating tweets that were posted in other languages. In order to accomplish this we utilized the translateR package and the Google Translate API (configured via Google Cloud) to translate those tweets to English. Due to the large volume of tweets/characters we decided to only translate the top six languages. Additionally, the Google Translate API has limitations in regards to the number of characters that can be queried per second and day. This service would cost a company $20 for every 1,000,000 characters translated through the API. To determine the top six we created a bar graph of the language codes without any English or Undefined tweets (see "Tweets by Language" chart). The top six



languages include French, Spanish, Portuguese, Dutch, German, and Japanese. In order to perform the translation, we read the twitter data into a data frame called df_source_lang. After subsetting each language into separate data frames we rbinded each language together and then rbinded the translated data frame to the main data frame that excluded the translated languages. The new translated data frame was then used in the analysis of the words used in the tweets - word networks and sentiment analysis. We found that the translated data did not alter the sentiment results as much as we expected due to the small percentage of tweets compared to those in English. It did, however, affect the word network slightly as some of the words became associated with others and some words became more prevalent.

The Conclusion

Challenges & Constraints

Twitter's API's are powerful tools, the API we utilized was free but paid API's exist. This free API comes with several limitations. The main challenge was the 10 day limit on Tweets that can be pulled. The other is the 18,000 Tweet limit per 15 minutes. While our data set was much smaller than 18,000 tweets we couldn't make too many calls when troubleshooting both the API and rtweet without having to wait for 15 minutes. While the paid API's offer developer support the free ones **do not.** During our project we quickly realized the API could go down or have issues and Twitter didn't seem to care. You are on your own in dealing with this. We continually ran into issue that occurred with "Deere" Tweets. The JSON being called back did not have same number of rows in each column. R uses rbind to convert these into data frames and fails when row counts are not consistent between columns. The rtweet package includes error handling that converts the data to a "Value" in R rather than a data frame so that all the Twitter data pulled is not lost. The work around was to use search_tweets to pull smaller number of tweets than the total that occurred, then through trial and error the missing row values would then not be collected in the call (example 5698 tweets occurred - only call 5600 tweets.) Packages and different versions of R and RStudio also caused conflicts when running code. One additional challenge that we encountered is that the Asian languages such as Japanese do not translate correctly as they show up in Unicode.

If We Had More Time…

If we had a chance to do it over, we would have had more time and a better understanding of Twitter's API's and rtweet. This would have allowed us to collect a much larger data set. We also would have like to see the specific variations over the 2017 North American grain harvest. Emoji analysis, creating reports showing the most used emojis and their sentiment scores, was a stretch goal for us. While we had the data sets to compare unicode to emojis and sentiments scores we ran out of time. Another problem with emojis is the unicode standards for them are always changing. Apple released the latest iOS update a few weeks before the project deadline creating new emoji characters, and we felt our efforts were better spent elsewhere. In regards to translating the data, we would have tried to translate all of the languages and perform analysis on just the translated data. We would also try to convert the Unicode Japanese and Emjoi's to meaningful data. Eventually we would also like to refine our code and reports into more module like functions and present them to the R user group within John Deere, possibly using a more reliable and powerful paid API.