Text Analysis of MyAirtel App Google Play Store Reviews in R

Simon Sayz

2020-07-12

Table of Contents

## 1. Text Analysis of MyAirtel App Google Play Store Reviews in R

Online Recharge Mobile, Bill Payments, Money Transfer Bank & Wallet, this is what MyAirtel App promises to deliver to its customer. [Airtel Uganda](https://www.airtel.co.ug/), is an Indian global telecommunications services company based in New Delhi, India. It operates in 18 countries across South Asia and Africa, and also in the Channel Islands

In this notebook, I took a deep dive into the reviews to uncover what their customers think about the application, what they like/dislike about the application and uncover some patterns.

## 2. Source of the Data

The Dataset is freely available on [Google Play Store](https://play.google.com/store/apps/details?id=com.airtel.africa.selfcare&showAllReviews=true) and was scrapped with [Beautiful Soup](https://pypi.org/project/beautifulsoup4/), a python library for scrapping websites and Loaded into R for further Text Analysis

## 2. Load the Libraries

Some of the Packages used in the Analysis

library(tidyverse)  
library(Amelia)  
library(rebus)  
library(bbplot)  
library(tidytext)  
library(tidymodels)  
library(lubridate)  
library(patchwork)  
library(ggthemes)  
library(knitr)  
library(emo)

## 3. Read in the Dataset

reviews <- readr::read\_csv(file.choose()) %>%   
 select(-c(reviewId, userName, userImage, appId))  
  
reviews\_copy <- reviews

## 4. Quick Glimpse of the Dataset

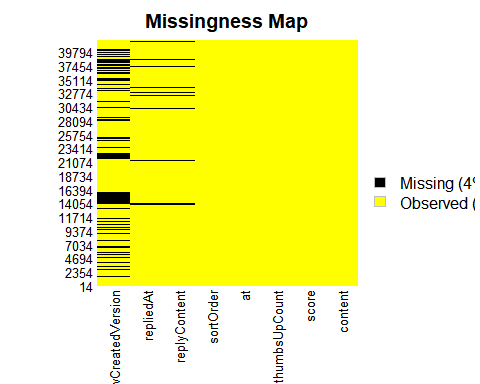
kable(head(reviews\_copy, 2))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| content | score | thumbsUpCount | reviewCreatedVersion | at | replyContent | repliedAt | sortOrder |
| This App is not matured. It won’t run unless you have active data plan. A payment App should run let alone a network provider app. I have issue registering my PORTED SIM as it always telling me it’s not in active state. It’s very annoying. More so. This App is supposed to allow one add another number in the case I have 2 Airtel line as other network apps allowed. Work on it. | 1 | 206 | 1.1.4 | 2020-06-30 21:45:05 | NA | NA | most\_relevant |
| Now it works, now it don’t. Your app fails when the need is dire. And worst, access denied until update is executed is a policy that sucks. And the fact that you would dare dictate on this one makes air-tel(l) a tale-case of polluted minds. | 1 | 13 | 1.1.4 | 2020-07-01 04:13:01 | Hi Sphinx. may we ask you to please contact us at [myairtel@africa.airtel.com](mailto:myairtel@africa.airtel.com) and explain the situation in detail? We’d like to improve our app in every aspect. | 2020-07-07 10:32:02 | most\_relevant |

# 5. Missingness Map

Some of the Columns are missing some observations for obvious reasons e.g The company doesn’t reply to every single review and thus the column will miss some data.

missmap(reviews\_copy, col = c("Black", "Yellow"))



## 6. Some Basic CleanUp and Processing

Some Column Names like Peoples’ names are not Important to this analysis and will be removed privacy reasons. 🔏

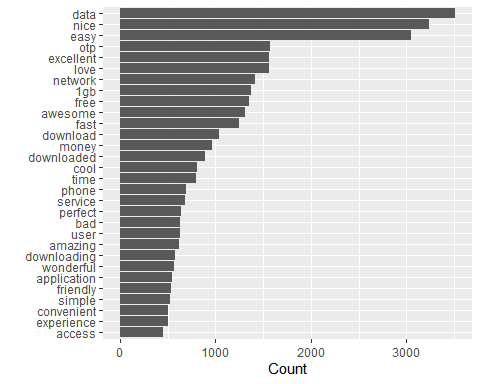
## This will come in handy when am modelling  
pattern <- DGT %R% optional(DGT)  
  
reviews\_processed <- reviews\_copy %>%   
 # na.omit(reviewCreatedVersion) %>%   
 mutate(version\_extracted = str\_extract(reviewCreatedVersion, pattern = pattern)) %>%  
 mutate(version\_nmbr = as.numeric(version\_extracted)) %>%   
 mutate(year = year(at),  
 month = month(at, label = TRUE),   
 week\_day = wday(at, label = TRUE))

## 7. What are the Most Common Used Words in the Reviews?

Top 30 most common words in the reviews

*Stop Words* and also Words like “App”, “Airtel” are filtered out as they don’t bring much value to this analysis and way too common 🔪

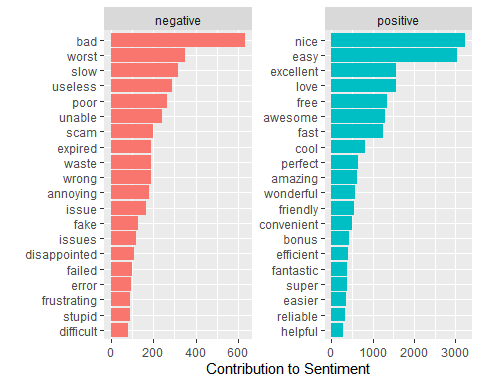
reviews\_processed %>%   
 unnest\_tokens(word, content) %>%   
 anti\_join(stop\_words, by="word") %>%   
 filter(!word %in% c("app", "airtel")) %>%   
 count(word, sort = TRUE) %>%   
 head(30) %>%   
 mutate(word = fct\_reorder(word, n)) %>%   
 ggplot(aes(word, n)) +  
 geom\_col() +  
 coord\_flip() +  
 labs(x="", y="Count")



## 8. What are the Most Common Positive and Negative Words?

Using the **Bing Lexicons**, you get scores for Positive/Negative Words, these are the Top 20 most common -ve and +ve Words

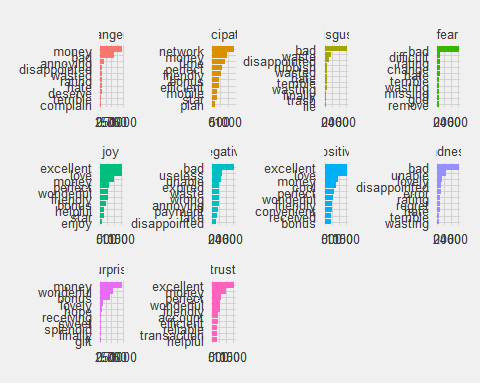
reviews\_processed %>%   
 unnest\_tokens(word, content) %>%   
 inner\_join(get\_sentiments("bing")) %>%   
 anti\_join(stop\_words, by="word") %>%   
 select(word, sentiment) %>%   
 count(word, sentiment, sort = TRUE) %>%   
 ungroup() %>%   
 group\_by(sentiment) %>%   
 top\_n(20) %>%   
 ungroup() %>%   
 mutate(word = fct\_reorder(word, n)) %>%   
 ggplot(aes(word, n, fill = sentiment)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~sentiment, scales = "free") +   
 coord\_flip() +  
 labs(y = "Contribution to Sentiment", x="")



### 8.1 It is important to see which words contribute to your sentiment scores.

What exactly contribute most the different sentiment like anget, disgust, fear etc

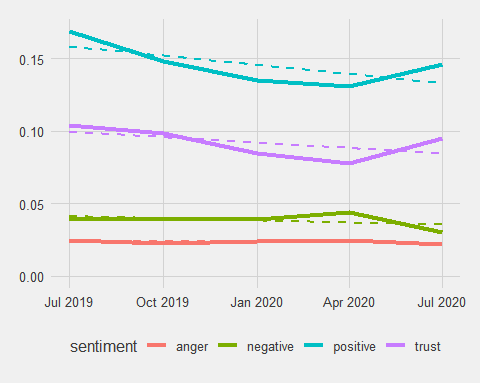
reviews\_processed %>%  
 unnest\_tokens(word, content) %>%   
 anti\_join(stop\_words, by="word") %>%   
 inner\_join(get\_sentiments("nrc")) %>%   
 # Count by word and sentiment  
 count(word, sentiment) %>%  
 # Group by sentiment  
 group\_by(sentiment) %>%  
 # Take the top 10 words for each sentiment  
 top\_n(10) %>%  
 ungroup() %>%  
 mutate(word = reorder(word, n)) %>%  
 # Set up the plot with aes()  
 ggplot(aes(word, n, fill=sentiment)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~ sentiment, scales = "free") +  
 coord\_flip() +  
 theme\_fivethirtyeight()

 Money is the looks to be the biggest driver of the *anger* Sentiment.

### 8.2 Sentiment changes with time

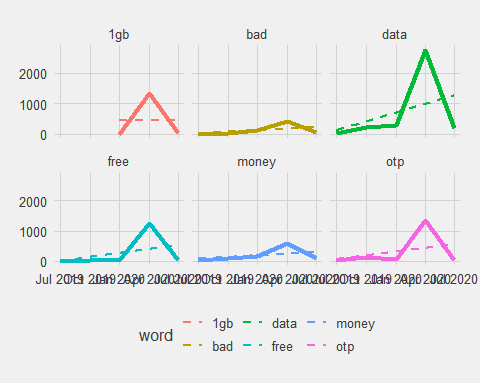
How have the different sentiments faired over the years, Let’s look at Positive, Negative, Trust and Anger

sentiment\_by\_time <- reviews\_processed %>%  
 unnest\_tokens(word, content) %>%   
 anti\_join(stop\_words, by="word") %>%   
 # Define a new column using floor\_date()  
 mutate(date = floor\_date(at, unit = "3 months")) %>%  
 # Group by date  
 group\_by(date) %>%  
 mutate(total\_words = n()) %>%  
 ungroup() %>%  
 # Implement sentiment analysis using the NRC lexicon  
 inner\_join(get\_sentiments("nrc"), by="word")  
  
  
sentiment\_by\_time %>%  
 # Filter for positive and negative words  
 filter(sentiment %in% c("positive", "negative", "trust", "anger")) %>%  
 # Count by date, sentiment, and total\_words  
 count(date, sentiment, total\_words) %>%  
 ungroup() %>%  
 mutate(percent = n / total\_words) %>%  
 # Set up the plot with aes()  
 ggplot(aes(date, percent, color = sentiment))+  
 geom\_line(size = 1.5) +  
 geom\_smooth(method = "lm", se = FALSE, lty = 2) +  
 expand\_limits(y = 0) +  
 theme\_fivethirtyeight()

 Its looking really good for **Airtel** looking at the Graph. *Negative energy* is dropping and the *Trust* in the company is increasing steadily.

### 8.3 How have words been used over time

wrds <- c("otp", "data", "money", "free", "1gb", "bad")  
reviews\_processed %>%  
 unnest\_tokens(word, content) %>%   
 anti\_join(stop\_words, by="word") %>%   
 mutate(date = floor\_date(at, "3 month")) %>%  
 filter(word %in% wrds ) %>%  
 count(date, word) %>%  
 ungroup() %>%  
 ggplot(aes(date, n, color = word)) +  
 # Make facets by word  
 facet\_wrap(~ word) +  
 geom\_line(size = 1.5, show.legend = FALSE) +  
 geom\_smooth(method = "lm", se = FALSE, lty = 2) +  
 expand\_limits(y = 0) +  
 # theme(legend.position = "none") +  
 theme\_fivethirtyeight()

 Words picked at Random from the most common words, Looks like *1gb* and *Free* peaked at in the Same Month. Could have been a promotion or somthing close to that.

### 8.4 What is the Average Rating for a Word

Words count greater than 500 for each individual word

reviews\_processed %>%  
 unnest\_tokens(word, content) %>%   
 anti\_join(stop\_words, by="word") %>%   
 group\_by(word) %>%   
 summarize(avg\_rating = mean(score, na.rm = TRUE),  
 n = n()) %>%  
 filter(n > 500) %>%   
 arrange(avg\_rating)

## # A tibble: 31 x 3  
## word avg\_rating n  
## <chr> <dbl> <int>  
## 1 otp 1.37 1568  
## 2 downloaded 1.77 894  
## 3 bad 1.80 633  
## 4 downloading 1.92 576  
## 5 download 2.17 1038  
## 6 phone 2.31 696  
## 7 1gb 2.37 1378  
## 8 data 2.44 3514  
## 9 free 2.56 1354  
## 10 time 3.06 801  
## # ... with 21 more rows

reviews\_processed %>%  
 unnest\_tokens(word, content) %>%   
 anti\_join(stop\_words, by="word") %>%   
 group\_by(word) %>%   
 summarize(avg\_rating = mean(score, na.rm = TRUE),  
 n = n()) %>%  
 filter(n > 500) %>%   
 arrange(desc(avg\_rating))

## # A tibble: 31 x 3  
## word avg\_rating n  
## <chr> <dbl> <int>  
## 1 excellent 4.90 1565  
## 2 amazing 4.83 619  
## 3 awesome 4.82 1308  
## 4 perfect 4.81 641  
## 5 wonderful 4.81 563  
## 6 convenient 4.80 509  
## 7 simple 4.78 522  
## 8 love 4.77 1559  
## 9 easy 4.75 3049  
## 10 fast 4.73 1248  
## # ... with 21 more rows

*otp(One Time Password)* get a very low average rating 💩. These are must people failing to login ir signup onto the application.

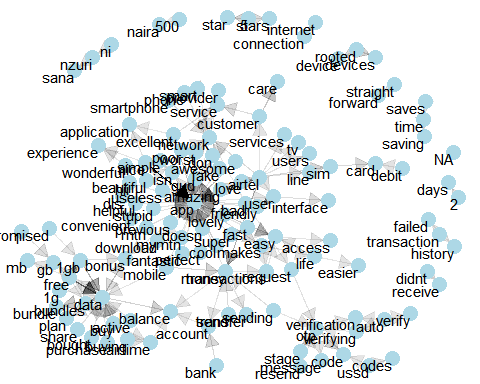
*1gb* and *free* also get a very low average rating. These could be people who didnt recieve their free 1gb 😒

So far we’ve considered words as individual units, and considered their relationships to sentiments. However, many interesting text analyses are based on the relationships between words, e.g examining which words tend to follow others immediately

## 8.5 Visualizing a network of bigrams

Lets visualize all of the relationships among words simultaneously, rather than just the top few at a time.

library(igraph)  
library(ggraph)  
library(widyr)  
  
set.seed(12345)  
  
bigrams\_ratings <- reviews\_processed %>%  
 unnest\_tokens(bigrams, content, token = "ngrams", n = 2) %>%   
 select(bigrams, everything())  
 # sample\_n(10) %>%   
 # pull(bigrams)  
  
bigrams\_ratings\_separated <- bigrams\_ratings %>%   
 separate(bigrams, c("word1", "word2", sep = " ")) %>%   
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word) %>%   
 count(word1, word2, sort = TRUE)  
  
bigram\_graph <- bigrams\_ratings\_separated %>%   
 filter(n > 20) %>%   
 graph\_from\_data\_frame()  
  
  
a <- grid::arrow(type = "closed", length = unit(.15, "inches"))  
  
ggraph(bigram\_graph, layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = n), show.legend = FALSE,  
 arrow = a, end\_cap = circle(.07, 'inches')) +  
 geom\_node\_point(color = "lightblue", size = 5) +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1) +  
 theme\_void()

 *App* is one of the common centers of nodes which is often followed by words like amazing, lovely, cool, beautiful etc

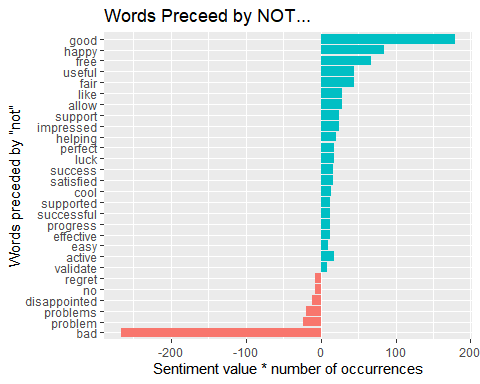
*Data* is also another common center of the nodes and its often followed by active, bought, purchase, bonus etc

We also see pairs or triplets along the outside that form common short phrases (“ni nzuri sana”, “500 naira”, or “internet connection”).

## 8.6 Words preceded by Not, No, Never, Without

By performing sentiment analysis on the bigram data, we can examine how often sentiment-associated words are preceded by “not” or other negating words like “no”, “Never” and “Without”

negation\_words <- c("not", "no", "never", "without")  
AFINN <- get\_sentiments("afinn")  
bigrams\_ratings %>%  
 separate(bigrams, into = c("word1", "word2"), sep = " ") %>%   
 filter(word1 %in% negation\_words) %>%   
 inner\_join(AFINN, by = c(word2 = "word")) %>%  
 count(word1, word2, value, sort = TRUE) %>%   
 mutate(contribution = n \* value) %>%  
 arrange(desc(abs(contribution))) %>%  
 head(30) %>%   
 mutate(word2 = reorder(word2, contribution)) %>%  
 ggplot(aes(word2, n \* value, fill = n \* value > 0)) +  
 geom\_col(show.legend = FALSE) +  
 xlab("Words preceded by \"not\"") +  
 ylab("Sentiment value \* number of occurrences") +  
 coord\_flip() +  
 labs(title = "Words Preceed by NOT...")



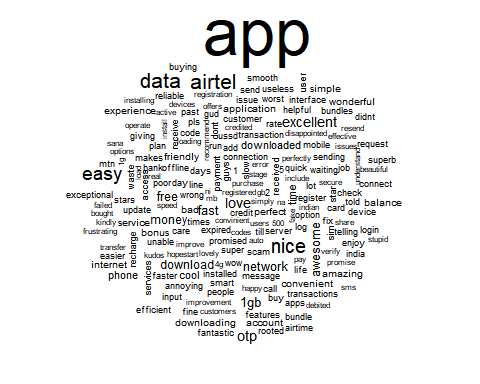
# facet\_wrap(~word1, ncol = 2)

The bigrams “not good” and “not happy” were overwhelmingly the largest causes of misidentification, making the text seem much more positive than it is. But we can see phrases like “no problem” and “not bad” sometimes suggest text is more negative than it is.

## 9. Word Cloud

Text analysis is never complete without a word cloud. 😄

library(wordcloud)  
  
reviews\_processed %>%  
 unnest\_tokens(word, content) %>%   
 anti\_join(stop\_words, by="word") %>%   
 count(word) %>%  
 with(wordcloud(word, n, max.words = 200))



### Future Work

1. A Sentiment Model to Predict a Rating Based the content in the Review.
2. An Interactive Web Application to bring the Analysis to Life.

### Helpul Links

* (R for Data Science)[<https://r4ds.had.co.nz/>]
* (Text Mining in R)[<https://www.tidytextmining.com/>]
* (Sentiment Anlysis in R from DataCamp)[<https://campus.datacamp.com/>]