Sentiment Analysis Using Google Play store Reviews Data

Objective

This report aims to provide insights into the research we undertook to understand the customer's expectations towards mobile messaging applications and help our client find insights that will help them launch their new messaging application in May 2021.

Background

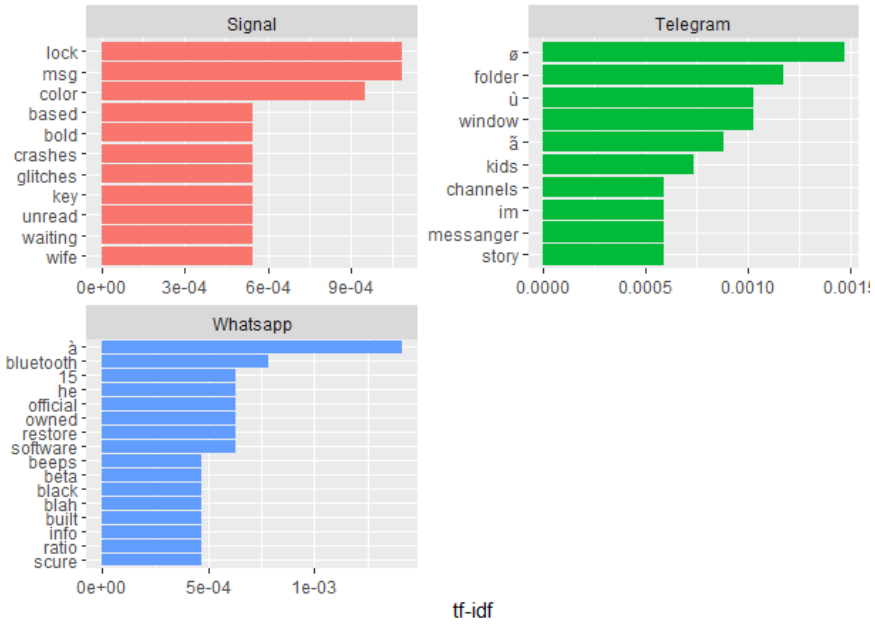
- The recent privacy rules violation issue surrounding WhatsApp has raised concern on all the well-established messaging applications like WhatsApp, Facebook paving way for new applications like signal. To take advantage of this situation, our client wants to launch their messaging application which is designed using the blockchain technology into the market.
- With this in mind, we want to analyze the available data to understand what users are expecting from these applications.
- We selected Google play store reviews as the source of our data and scrapped reviews using google_play_scrapper API.

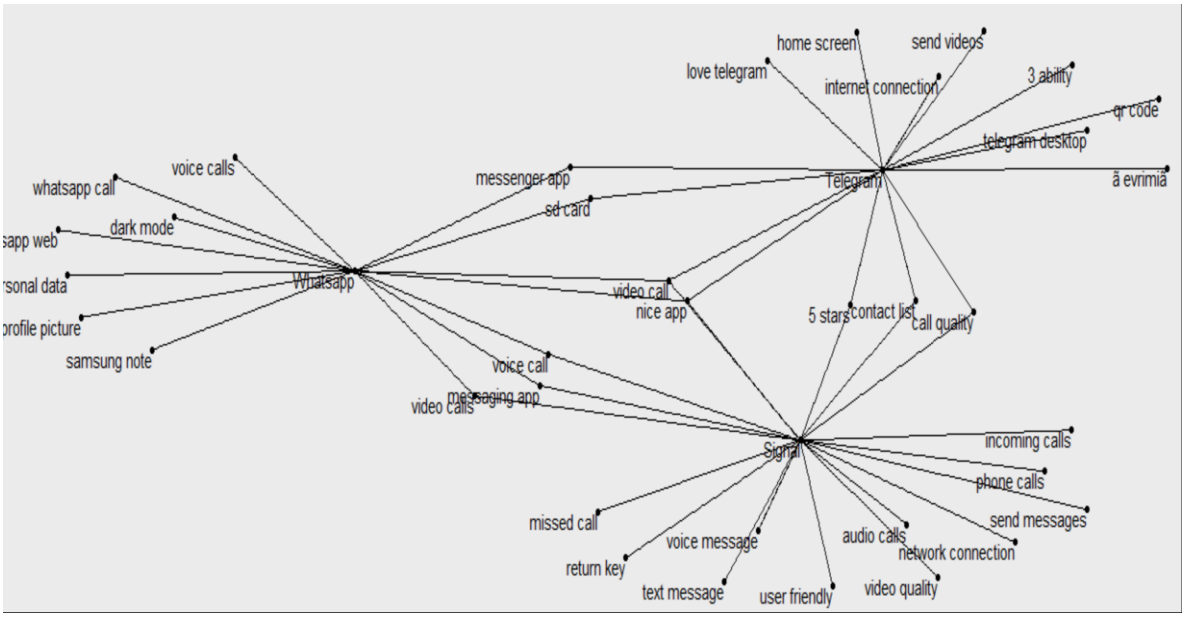
Methodology

- As Google Play store is an Android platform we are mainly focusing on three android messaging applications which are WhatsApp, Telegram and Signal.
- The play store reviews are collected along with a mandatory rating score that ranges from 1 to 5, where 1 indicates strongly satisfied, and 5 indicate that they strongly dissatisfied. To get a balanced data set for our analysis we divided these ratings into three categories: Positive sentiment, which contains 4 & 5 stars ratings, Negative sentiment which includes 1 & 2-star ratings and Neutral sentiment with 3-star rating reviews. We extracted 60 reviews from each of these three categories for each of the above mentioned three applications.
- We are focusing more on the recent information, so we used the filters most relevant and newest reviews. The data that we are analyzing belongs to only the US Market.

Findings

- We discovered the following findings from our analysis
    - Using the TF-IDF, we found that lock, folder, Bluetooth, crashes, and glitches are important words. Following are the top 10 important words that are used in reviews by customers in each application





Bigram

- We also designed a bigram to understand the relationship between the words from our reviews. Following are the findings from the bigram we generated
  - It looks like WhatsApp customers are mainly talking about its profile picture, WhatsApp web, video and audio calling features. We can also see WhatsApp has a lot of Samsung Customers.
  - Telegram users seem to talk more about the QRcode, video sharing, call quality features, and primary features.
  - Being new to this field signal customers are mainly focusing on the messaging, video, audio and network quality. Even though we don't see any new features here, It looks like signal is more user friendly.

Recommendations

- Based on our findings, we can see that voice calling, video calling, and messaging are the basic features that all three applications provide. The Initial release of our application should include these features without fail.
- Before releasing any new functionalities, our unique selling proposition ensure that our required features work without any issues.
- Make sure to have the web application ready as well, as we see customers are talking more about that.
- As we can we that most of the Samsung users are using WhatsApp, we should come up with some campaigns or tie-ups with Samsung to attract that significant customer segment.
- The data we are using is from Google App Store which is for android users only so to get insights that are specially related to apple users as apple has a significant market we need to do a separate analysis

Action Steps

- Launching separate marketing Campaigns for both Apple and Android users
- Making sure that the basic features work in our initial release of May without any issues.
- Finding out which features from other applications the users prefer to get in next releases in the next 3 to 6 months.
- Coming up with our own features which are not offered by other applications to attract customers from other applications and release by the end of this year.

HULT
INTERNATIONAL
BUSINESS SCHOOL

Conclusion

As the number of mobile users and penetration of the internet in society is increasing day by day, the messaging application market is bound to grow more in the future. As we can see most of the customers from our competitors are mainly using those applications because of the basic functionalities like Messaging, Audio & Video calling, so we can be sure that the customers are not expecting something out of the box to download such applications. If we can provide these basic functionalities without any issues we will get success in this category of applications according to our analysis.

Vishal Lingineni
Insights Analyst
Feb 2021

Appendix

R-code:

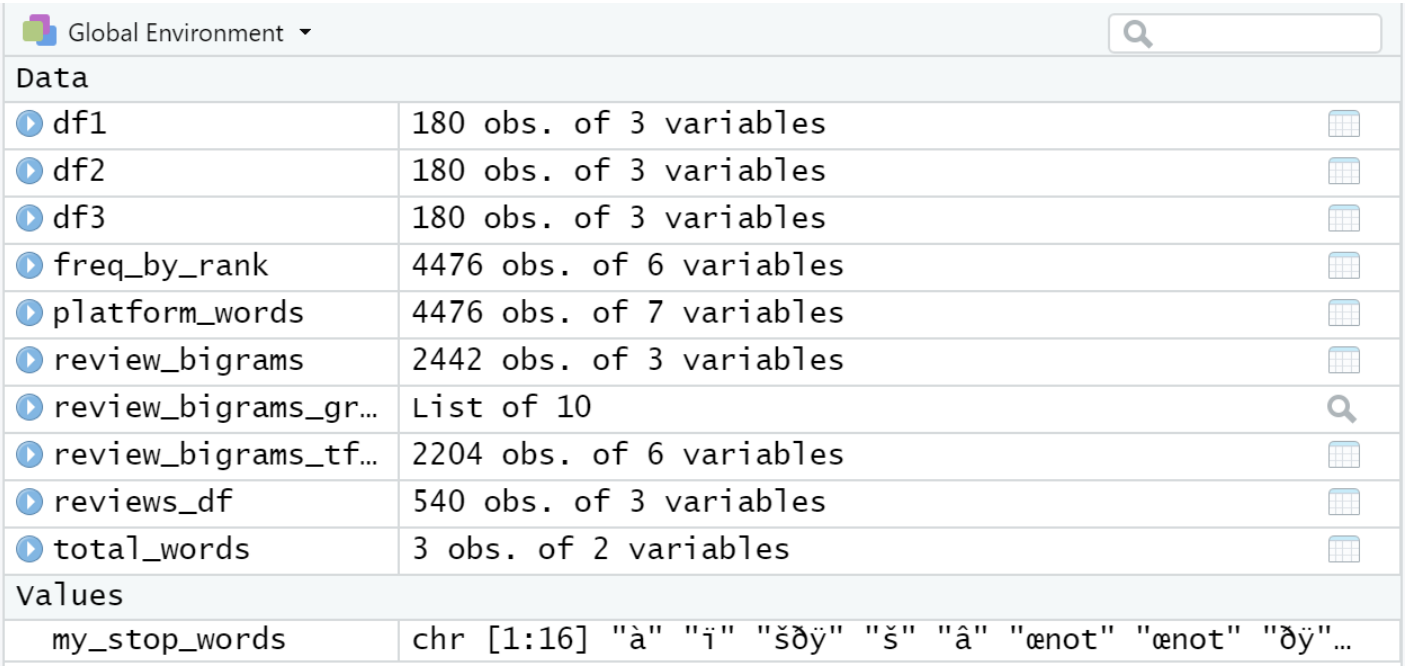
```
# ------------------------------------ Importing Dataset ------------------------------------------------------------#
library(readxl)
df1 <- read_excel("C:/Users/Lingi/OneDrive - Education First/MBAN Courses/NLP/Individual Assignment/Files/Whatsapp_reviews.xlsx")
df2 <- read_excel("C:/Users/Lingi/OneDrive - Education First/MBAN Courses/NLP/Individual Assignment/Files/Telegram_reviews.xlsx")
df3 <- read_excel("C:/Users/Lingi/OneDrive - Education First/MBAN Courses/NLP/Individual Assignment/Files/Signal_reviews.xlsx")
df1 <- df1[,c("content","thumbsUpCount")]
df1$platform <- c('Whatsapp')
df2 <- df2[,c("content","thumbsUpCount")]
df2$platform <- c('Telegram')
df3 <- df3[,c("content","thumbsUpCount")]
df3$platform <- c('Signal')

# Getting Revieew number column to the beginning
reviews_df <- rbind(df1,df2,df3)
reviews_df <- reviews_df[c('platform',"content","thumbsUpCount")]

# Renaming the content column name as text
colnames(reviews_df) <- c("platform","text","thumbsUpCount")

# Summarizing all datframe to check whether all the columns are correctly assigned
str(reviews_df)

# ------------------------------------------ Tokenizing ------------------------------------------------------------#

# Tokenizing the reviews and also find the number of times each token appeared in each platform
library(dplyr)
library(tidytext)
platform_words <- reviews_df %>%
  unnest_tokens(word, text) %>%
```

```
                count(platform, word, sort = TRUE) %>%
                ungroup()

# finding the total number of tokens in each platform
total_words <- platform_words %>%
  group_by(platform) %>%
  summarize(total = sum(n))

# Joining both tables
platform_words <- left_join(platform_words, total_words)

# ---------------------------------- Visualizing the distribution of n/total ----------------------------------------#
library(ggplot2)
ggplot(platform_words, aes(n/total, fill = platform)) +
  geom_histogram(show.legend = FALSE) +
  xlim(NA, 0.008) +
  facet_wrap(~platform, ncol = 1, scales = "free_y")



#------------------------------------------------------- Zipf’s Law ---------------------------------------------------#

freq_by_rank <- platform_words %>%
  group_by(platform) %>%
  mutate(rank = row_number(),
         `term frequency` = n/total)
freq_by_rank

#rank column here tells us the rank of each word within the frequency table
#visualizing Zip's law
freq_by_rank %>%
  ggplot(aes(rank, `term frequency`, color = platform)) +
  geom_abline(intercept = -0.62, slope = -1.1, color = "gray50", linetype = 2) +
  geom_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +
  scale_x_log10() +
  scale_y_log10()

#------------------------------------------------------- TF-IDF -------------------------------------------------------#

#creating TF_IDF
platform_words <- platform_words %>%
  bind_tf_idf(word, platform, n)
platform_words

platform_words %>%
  select(-total) %>%
  arrange(desc(tf_idf))
# visualizing TF_IDF
platform_words %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  group_by(platform) %>%
  top_n(10) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill = platform)) +
```

```
                geom_col(show.legend = FALSE) +
                labs(x = NULL, y = "tf-idf") +
  facet_wrap(~platform, ncol = 2, scales = "free") +
  coord_flip()

# here we identified some of the top features that are important in each of the platforms



# ------------------------------------------------------ N - Grams ------------------------------------------------#

# Even though we identified that are important for each platform still they dont make senses

# Tokenizing the reviews and also find the number of times each token appeared in each platform
library(dplyr)
library(tidytext)
my_stop_words <- c('à' ,'ï','šðÿ','š','â', 'œnot',
                   'œnot','ðÿ','ƒðÿ','ƒðÿ', 'ðÿ', 'žðÿ','º','ø','øªùˆø ø','ù øªùˆø')
review_bigrams <- reviews_df %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2) %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word1 %in% my_stop_words) %>%
  filter(!word2 %in% my_stop_words) %>%
  filter(!word1 == "NA") %>%
  filter(!word2 == "NA") %>%
  unite(bigram, word1, word2, sep=" ")

review_bigrams

# priotitizing the ngrams usinf tf_idf
review_bigrams_tf_idf <- review_bigrams %>%
  count(platform, bigram) %>%
  bind_tf_idf(bigram, platform, n) %>%
  arrange(desc(tf_idf))

review_bigrams_tf_idf

# visualizing the bigram
# preparing for chart
review_bigrams_graph <- review_bigrams_tf_idf %>%
  filter(n>2) %>%
  graph_from_data_frame()
review_bigrams_graph

ggraph(review_bigrams_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)



# ------------------------------------------------------ Trigram ------------------------------------------------------#
```

```
                fs_ngrams <- reviews_df %>%
  unnest_tokens(trigram, text, token = "ngrams", n=3) %>%
  separate(trigram, c("word1", "word2","word3"), sep = " ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word3 %in% stop_words$word) %>%
  filter(!word1 == "NA") %>%
  filter(!word2 == "NA") %>%
  filter(!word3 == "NA") %>%
  unite(trigram, word1, word2,word3, sep=" ")

fs_ngrams

# priotitizing the ngrams usinf tf_idf
fs_ngrams_tf_idf <- fs_ngrams %>%
  count(platform, trigram) %>%
  bind_tf_idf(trigram, platform, n) %>%
  arrange(desc(tf_idf))

fs_ngrams_tf_idf

# visualizing the bigram
# preparing for chart
fs_ngrams_graph <- fs_ngrams_tf_idf %>%
  filter(n>1) %>%
  graph_from_data_frame()
fs_ngrams_graph

ggraph(fs_ngrams_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)
```

R Outputs:



```
> freq_by_rank
# A tibble: 4,476 x 6
# Groups:   platform [3]
   platform word      n total  rank `term frequency`
   <chr>    <chr> <int> <int> <int>            <dbl>
 1 Signal   the     311  8085     1           0.0385
 2 Signal   to      259  8085     2           0.0320
 3 Telegram the     256  7470     1           0.0343
 4 Signal   i       241  8085     3           0.0298
 5 Telegram to      241  7470     2           0.0323
 6 Telegram i       220  7470     3           0.0295
 7 Whatsapp the     217  6985     1           0.0311
 8 Whatsapp i       206  6985     2           0.0295
 9 Whatsapp to      201  6985     3           0.0288
10 Signal   and     199  8085     4           0.0246
# ... with 4,466 more rows
> #rank column here tells us the rank of each word within the frequency table
> #visualizing Zip's law
> freq_by_rank %>%
+   ggplot(aes(rank, `term frequency`, color = platform)) +
+   geom_abline(intercept = -0.62, slope = -1.1, color = "gray50", linetype = 2) +
+   geom_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +
+   scale_x_log10() +
+   scale_y_log10()
> #creating TF_IDF
> platform_words <- platform_words %>%
+   bind_tf_idf(word, platform, n)
> platform_words
# A tibble: 4,476 x 7
   platform word      n total     tf   idf tf_idf
   <chr>    <chr> <int> <int>  <dbl> <dbl>  <dbl>
 1 Signal   the     311  8085 0.0385     0      0
 2 Signal   to      259  8085 0.0320     0      0
 3 Telegram the     256  7470 0.0343     0      0
 4 Signal   i       241  8085 0.0298     0      0
 5 Telegram to      241  7470 0.0323     0      0
 6 Telegram i       220  7470 0.0295     0      0
 7 Whatsapp the     217  6985 0.0311     0      0
 8 Whatsapp i       206  6985 0.0295     0      0
 9 Whatsapp to      201  6985 0.0288     0      0
10 Signal   and     199  8085 0.0246     0      0
# ... with 4,466 more rows
> platform_words %>%
+   select(-total) %>%
+   arrange(desc(tf_idf))
# A tibble: 4,476 x 6
   platform word          n       tf   idf   tf_idf
   <chr>    <chr>     <int>    <dbl> <dbl>    <dbl>
 1 Telegram ø            10 0.00134   1.10 0.00147
 2 Whatsapp à             9 0.00129   1.10 0.00142
 3 Telegram folder        8 0.00107   1.10 0.00118
 4 Signal   lock          8 0.000989  1.10 0.00109
 5 Signal   msg           8 0.000989  1.10 0.00109
 6 Telegram ù             7 0.000937  1.10 0.00103
 7 Telegram window        7 0.000937  1.10 0.00103
 8 Signal   color         7 0.000866  1.10 0.000951
 9 Telegram ã             6 0.000803  1.10 0.000882
10 Whatsapp bluetooth     5 0.000716  1.10 0.000786
# ... with 4,466 more rows
```

```
> review_bigrams
# A tibble: 2,531 x 3
   platform thumbsUpCount bigram
   <chr>            <dbl> <chr>
 1 Whatsapp          1228 sits idol
 2 Whatsapp          1228 phone's default
 3 Whatsapp          1228 default sms
 4 Whatsapp          1228 sms app
 5 Whatsapp          1228 contact book
 6 Whatsapp          1228 book adding
 7 Whatsapp          1228 adding people
 8 Whatsapp          1228 contacts chatting
 9 Whatsapp          1228 text messaging
10 Whatsapp          1228 messaging app
# ... with 2,521 more rows
> View(review_bigrams)
> View(review_bigrams)
> # priotitizing the ngrams usinf tf_idf
> review_bigrams_tf_idf <- review_bigrams %>%
+   count(platform, bigram) %>%
+   bind_tf_idf(bigram, platform, n) %>%
+   arrange(desc(tf_idf))
> review_bigrams_tf_idf
# A tibble: 2,259 x 6
   platform bigram                n      tf   idf  tf_idf
   <chr>    <chr>             <int>   <dbl> <dbl>   <dbl>
 1 Whatsapp à à                   8 0.0109   1.10 0.0120
 2 Whatsapp voice calls           6 0.00820  1.10 0.00901
 3 Whatsapp whatsapp web          5 0.00683  1.10 0.00750
 4 Telegram ø ù                   5 0.00556  1.10 0.00610
 5 Signal   incoming calls        4 0.00445  1.10 0.00489
 6 Signal   network connection    4 0.00445  1.10 0.00489
 7 Telegram love telegram         4 0.00444  1.10 0.00488
 8 Telegram ù ø                   4 0.00444  1.10 0.00488
 9 Whatsapp personal data         3 0.00410  1.10 0.00450
10 Whatsapp whatsapp call         3 0.00410  1.10 0.00450
# ... with 2,249 more rows

> # visualizing the bigram
> # preparing for chart
> review_bigrams_graph <- review_bigrams_tf_idf %>%
+   filter(n>2) %>%
+   graph_from_data_frame()
> review_bigrams_graph
IGRAPH ef4284b DN-- 44 55 --
+ attr: name (v/c), n (e/n), tf (e/n), idf (e/n), tf_idf (e/n)
+ edges from ef4284b (vertex names):
 [1] Whatsapp->à à                Whatsapp->voice calls
 [3] Whatsapp->whatsapp web       Telegram->ø ù
 [5] Signal  ->incoming calls     Signal  ->network connection
 [7] Telegram->love telegram      Telegram->ù ø
 [9] Whatsapp->personal data      Whatsapp->whatsapp call
[11] Signal  ->audio calls        Signal  ->phone calls
[13] Signal  ->return key         Signal  ->text message
[15] Telegram->3 ability          Telegram->ã evrimiã
+ ... omitted several edges
```