T-Mobile : Social Media Analytics

Group 03: Fabian Celi, Sharma Rishabh, Alejandra Zambrano

February 6, 2020

Introduction

T-Mobile is a mobile (cell phone) network operator headquartered in Bonn, Germany. Founded in 1990, it is a part of Deutsche Telekom and belongs to the FreeMove alliance. It is a group of mobile phone companies, all owned by Deutsche Telekom, that operate GSM and UMTS networks in Europe and the United States. It offers postpaid and prepaid wireless voice, messaging and data services and wholesale wireless services. The goal of the analysis is to draw a complete twitter landscape of the company and its users. We intend to look at the tweeting behaviour, engagement pattern and sentimental analysis of the tweets written by the company as well as by the users mentioning the company or its brands/services.

We downloaded the data using the Twitter API. For analysing company tweets, we downloaded the tweets from the following official handles: @TMobile, @TMobileTruckPHM, @T_Labs, @TMobileHelp, @JohnLegere, @SievertMike, @MetroByTMobile, @TMobileArena, @TMobilePark, @tmobilecareers, @TMobileBusiness, @TMobileIR, @TMobileNews

For the users' data, we downloaded tweets with the following hashtags and keywords which are either used for tagging TMobile or its various brands and promotions:

#AreYouWithUs, #TmobileTruck, #TMobileCareers, #BeMagenta, #IamMagenta, #TeamMagenta, #TeamMetro,#TMobileBusiness, #TMUS, #TMobileTuesdays #5GThatWorks, #5gflex, #Nationwide5G, #AmericasNetwork, #uncarrier, #magentify, Magenta, Magenta plus,Essentials, TMobile

In both databases, we kept only tweets in English and excluded all retweets as well as duplicates. The users database has no tweets from any of the TMobile official handles mentioned above. Our final databases have 39,570 tweets from TMobile handles and 10,200 tweets from other users.

Data Download Challenges

To maximize our downloads, we first downloaded using the timeline function which didn't use our data allowance. This also allowed us to identify TMobileHelp as the most prolific account on which we could run additional queries to get more tweets. We used all of our FullArchive quota to download further tweets from TMobileHelp going back almost 2 months. Then we used the 30Days function to download the hashtags and keywords tweets appearing in the last 30 days.

```r
################################################
############## Loading libraries ###############
################################################

if(!require("rtweet")) install.packages("rtweet"); library("rtweet")
if(!require("tidyverse")) install.packages("tidyverse"); library("tidyverse")
if (!require("wordcloud")) {
  install.packages("wordcloud",repos="https://cran.rstudio.com/",
                   quiet=TRUE)
  require("wordcloud")
}
for (i in
c('SnowballC','slam','tm','Matrix','tidytext','dplyr','hunspell','purrr'
            ,'textstem','ggplot2','treemapify','treemap')){
  if (!require(i, character.only=TRUE)) install.packages(i, repos =
"http://cran.us.r-project.org")
  require(i, character.only=TRUE)
}

for (i in c("tidytext", "dplyr", "hunspell","topicmodels", "textstem",
"ggplot2", "ldatuning", "utf8", "wordcloud2", "wordcloud",
"RColorBrewer","flexdashboard")){  if (!require(i, character.only=TRUE))
install.packages(i, repos = "http://cran.us.r-project.org")
  require(i, character.only=TRUE)
}

################################################
####### Downloading tweets from twitter #######
################################################

# Authentication

app = 'fceli'
consumer_key = 'CONFIDENTIAL'
consumer_secret = 'CONFIDENTIAL'
access_token = 'CONFIDENTIAL'
access_secret = 'CONFIDENTIAL'

# Creation twitter_token

twitter_token <- create_token(
  app = app,
  consumer_key = consumer_key,
  consumer_secret = consumer_secret,
  access_token = access_token,
  access_secret = access_secret,
  set_renv=FALSE)
```

```
# Searching tweets by users from the company T-Mobile
# We used get_timeline for getting the 3200 latest tweets from each user.

tm_mobile_company=get_timeline(user=c('TMobileTruckPHM','TMobile','T_Labs','T
MobileHelp','JohnLegere','SievertMike',

'MetroByTMobile','TMobileArena','TMobilePark','tmobilecareers','TMobileBusine
ss',
                                      'TMobileIR','TMobileNews'), n = 5000,
include_rts=FALSE)

#Keeping only tweets in english

tm_mobile_company=tm_mobile_company[(tm_mobile_company$lang)=='en',]
save(tm_mobile_company, file = "C:/Users/azambranollano/OneDrive -
IESEG/Alejandra/Social Media Analytics/Group Project/userscompany.RData")

# Loading data previously downloaded from T-Mobile accounts

load('C:/Users/azambranollano/OneDrive - IESEG/Alejandra/Social Media
Analytics/Group Project/Data/handle.RData')
load('C:/Users/azambranollano/OneDrive - IESEG/Alejandra/Social Media
Analytics/Group Project/Data/fcTmob3.RData')
load('C:/Users/azambranollano/OneDrive - IESEG/Alejandra/Social Media
Analytics/Group Project/Data/rishtmobfullarch.RData')
load('C:/Users/azambranollano/OneDrive - IESEG/Alejandra/Social Media
Analytics/Group Project/Data/userscompany.RData')

# Joinig final base from users of the company

tmobile_company=rbind.data.frame(handle,fcTmob3,rishtmobfullarch,tm_mobile_co
mpany)

# Keeping only tweets in english, no retweets and deleting duplicate tweets

tmobile_company=tmobile_company[(tmobile_company$lang)=='en',]
tmobile_company=tmobile_company[!duplicated(tmobile_company$status_id),]
tmobile_company=tmobile_company[!(tmobile_company$is_retweet),]

# Saving the file for later use

save(tmobile_company, file = 'C:/Users/azambranollano/OneDrive -
IESEG/Alejandra/Social Media Analytics/Group Project/tmobile_company.RData')

# Searching tweets by keywords
# We also combine this search, seaching by search_fullarchive, we take care
of the first date downloaded to get more information

tm_mobile3<-search_30day(q='#AreYouWithUs', n=5000,env_name = 'tweets30')
```

```
tm_mobile4<-search_30day(q='#TmobileTruck', n=5000,env_name = 'tweets30')
tm_mobile5<-search_30day(q='#BeMagenta', n=5000,env_name = 'tweets30')
tm_mobile6<-search_30day(q='#TMobile', n=5000,env_name = 'tweets30')
tm_mobile7<-search_30day(q='#TMobileBusiness', n=5000,env_name = 'tweets30')
tm_mobile8<-search_30day(q='#TMUS', n=5000,env_name = 'tweets30')
tm_mobile9<-search_30day(q='#TMobileCareers', n=5000,env_name = 'tweets30')
tm_mobile10<-search_30day(q='#IamMagenta', n=5000,env_name = 'tweets30')
tm_mobile11<-search_30day(q='#TeamMagenta', n=5000,env_name = 'tweets30')
tm_mobile12<-search_30day(q='#TeamMetro', n=5000,env_name = 'tweets30')
tm_mobile13<-search_30day(q='#5GThatWorks', n=5000,env_name = 'tweets30')

a=rbind.data.frame(tm_mobile3,tm_mobile4,tm_mobile5,tm_mobile6,tm_mobile7,tm_
mobile8,

tm_mobile9,tm_mobile10,tm_mobile11,tm_mobile12,tm_mobile13)

# Saving the file for later use

save(a, file = "C:/users/azambranollano/OneDrive - IESEG/Alejandra/Social
Media Analytics/Group Project/actualizada.RData")

# Loading data previously downloaded for combined data sets

load('C:/Users/azambranollano/OneDrive - IESEG/Alejandra/Social Media
Analytics/Group Project/Data/actualizada.RData')
load('C:/Users/azambranollano/OneDrive - IESEG/Alejandra/Social Media
Analytics/Group Project/Data/tm24.RData')
load('C:/Users/azambranollano/OneDrive - IESEG/Alejandra/Social Media
Analytics/Group Project/Data/tdayshashtags2.RData')

tmobile_users=rbind.data.frame(a,tm_mobile24,rishtmobfullarch,rish)
tmobile_users=tmobile_users[(tmobile_users$lang)=='en',]
tmobile_users=tmobile_users[!duplicated(tmobile_users$status_id),]
tmobile_users=tmobile_users[!(tmobile_users$is_retweet),]

# Keeping tweets differents from users of the company

tmobile_users = anti_join(tmobile_users, tmobile_company, by="status_id")
tmobile_users = tmobile_users %>%
  filter(!screen_name %in%
c('TMobileTruckPHM','TMobile','T_Labs','TMobileHelp','JohnLegere','SievertMik
e',

'MetroByTMobile','TMobileArena','TMobilePark','tmobilecareers','TMobileBusine
ss',
                            'TMobileIR','TMobileNews'))
# Saving the file for later use
```

```
save(tmobile_users, file = "C:/Users/azambranollano/OneDrive -
IESEG/Alejandra/Social Media Analytics/Group Project/tmobile_users.RData")
```

Pre-process Data

In order to eliminate other undesired elements from the tweets and put all of them in a clean format, we performed some pre-processing operations as follows:

1. Remove numbers, punctuation and white spaces.
2. Remove stopwords: Stopwords are a set of commonly used words. We removed stopwords in order to concentrate on the important words.In this step, we also included some words that are used often when talking about T-Mobile, including the name of T-Mobile itself.
3. Convert to lower case: In order to standardize the tweets, we convert all of them in lowercase to ensure equal treatment of all words.
4. Lemmatization: for grouping together the different forms of a word, so they can be analyzed as a single item.

Pre-processing Challenges

Building our custom stopwords dictionary was an iterative process which we came back to time and again for adding more words while running our topic analysis model.

```
##### Loading files

#Tweets generated from the company are stored as "tmobile_company.RData"

load("C:/Users/azambranollano/OneDrive - IESEG/Alejandra/Social Media
Analytics/Group Project/tmobile_company.RData")

#Tweets generated from different users  are stored as "tmobile_users.RData"

load("C:/Users/azambranollano/OneDrive - IESEG/Alejandra/Social Media
Analytics/Group Project/tmobile_users.RData")

################################################
################ Cleaning Data ################
################################################

# Subsetting the data

tm_comments <- tmobile_company[,2:6]

hash_comments <- tmobile_users[(tmobile_users$is_retweet==FALSE),
c("status_id", "created_at", "screen_name", "source", "reply_to_screen_name",
"text", "followers_count")]

# Removing punctuations and numbers
```

```r
tm_comments <- mutate(tm_comments, text = gsub(x = text, pattern = "[0-
9]+|[[:punct:]]|\\(.*\\)|[']|[']", replacement = ""))
hash_comments <- mutate(hash_comments, text = gsub(x = text, pattern = "[0-
9]+|[[:punct:]]|\\(.*\\)|[']|[']", replacement = ""))

hash_comments <- hash_comments %>%
  mutate(source2 = ifelse(source == "Twitter for Android", "Android",
                          ifelse(source == "Twitter for iPhone", "iPhone",
                          ifelse(source == "Twitter Web App", "WebApp",
"Other"))))

# Tokenize, remove stop words and lemmatize

# Add more custom stop words

custom_stop_words <- tribble(
  ~word,  ~lexicon,
  # Add custom stop words
  "tmobile", "CUSTOM",
  "johnlegere",  "CUSTOM",
  "magenta", "CUSTOM",
  "telekomwall", "CUSTOM",
  "its", "CUSTOM",
  "were", "CUSTOM",
  "tmobilecareers", "CUSTOM",
  "areyouwithus", "CUSTOM",
  "bemagenta", "CUSTOM",
  "amp", "CUSTOM",
  "closerthanever", "CUSTOM",
  "youre", "CUSTOM",
  "tmobiletuesdys", "CUSTOM",
  "hey", "CUSTOM",
  "rondarousey", "CUSTOM",
  "sideshowjohn", "CUSTOM",
  "metropcs", "CUSTOM",
  "goldenknights", "CUSTOM",
  "tmobiletuesdays", "CUSTOM",
  "atomtickets", "CUSTOM",
  "goldenknights", "CUSTOM"
)

stop_words2 <- stop_words %>%
  bind_rows(custom_stop_words)

tm_lemm <- tm_comments %>%
  unnest_tokens(output = "word",
                input = text,
                token = "words",
```

```
                drop = FALSE, to_lower = TRUE) %>%
  anti_join(stop_words2)%>%
  mutate(word = lemmatize_words(word))

hash_lemm <- hash_comments %>%
  unnest_tokens(output = "word",
                input = text,
                token = "words",
                drop = FALSE, to_lower = TRUE) %>%
  anti_join(stop_words2)%>%
  mutate(word = lemmatize_words(word))
```

Sentiment Analysis

To determine the sentiment of the tweets from the company and users, we used mainly "AFINN" and "NCR" dictionaries. However, in order to analyse the consistency of the results we ran additionally "loughran" and "bing" dictionaries as well. The general sentiment from company and users tweets is positive.

Challenges

Evaluating whether certain steps such as stemming and spellcheck should be done or not. When we implemented stemming and then spellcheck, many words were wrongly changed to a different meaning that change the sentiments as well. Therefore, we decided not used those steps.

Determining sentiment

We ran different sentiment dictionaries and were able to identify the sentiment per the whole datasets: company and users. Furthermore, with the "AFINN" parametrization, we were qble to determine how positive or negative each tweet is based on the words that it contained.

Results

Company tweets

The main sentiments (above 7500 words) are positive, trust and anticipation. In positive sentiment, the words with more than 2000 repetitions are love, happy and gift. In trust sentiment, the words with more than 2000 repetitions are team and assist. Finally, in anticipation sentiment, the words with more than 2000 repetitions are time and start. The distribution of sentiment in days and hours does not follow any clear tendency; it is because mostly all tweets from the company are considered positive. However, between 5 to 10 am the are not many tweets.

User tweets

The main sentiments (above 2000 words) are positive, surprise, fear and anticipation. In positive sentiment, the words with more than 100 repetitions are love, hope, happy and

excite. In surprise sentiment, the word with more repetitions is chance. In happy sentiment, the word with more repetitions is love. Finally, in anticipation sentiment, the words with more than 100 repetitions are hope and happy. The highest point for positive tweets is from 0 to 4 am during the first day of the week and between 12 to 15 pm during the second day of the week. For negative tweets, it does not show a concentration in hours or weekdays.

```
#########   Sentiment of "tm_lemm" - tweets from the company  #########

TMCSentiment1 <- inner_join(tm_lemm,get_sentiments("bing"))
TMCSentiment2 <- inner_join(tm_lemm,get_sentiments("afinn"))
TMCSentiment3 <- inner_join(tm_lemm,get_sentiments("loughran"))
TMCSentiment4 <- inner_join(tm_lemm,get_sentiments("nrc"))

# Most positive/negative words

summarySentiment1 <- TMCSentiment1 %>%  count(word,sentiment,sort=TRUE) %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  arrange(n) %>%
  as.data.frame(stringsAsFactors=FALSE)

ggplot(summarySentiment1, aes(x = sentiment, y = n,fill=factor(sentiment))) +
  geom_col() +
  coord_flip() +
  labs(
    title = "Sentiment Counts in bing",
    x = "Counts",
    y = "Sentiment"
  )

summarySentiment2 <- TMCSentiment2 %>%  count(word,value,sort=TRUE) %>%
  group_by(value) %>%
  top_n(10) %>%
  arrange(n) %>%
  as.data.frame(stringsAsFactors=FALSE)

ggplot(summarySentiment2, aes(x = value, y = n,fill=factor(value))) +
  geom_col() +
  coold_flip() +
  labs(
    title = "Sentiment Counts in afinn",
    x = "Counts",
    y = "Sentiment"
  )

summarySentiment3 <- TMCSentiment3 %>%  count(word,sentiment,sort=TRUE) %>%
  group_by(sentiment) %>%
  top_n(10) %>%
```

```r
  arrange(n) %>%
  as.data.frame(stringsAsFactors=FALSE)

ggplot(summarySentiment3, aes(x = sentiment, y = n,fill=factor(sentiment))) +
  geom_col() +
  coord_flip() +
  labs(
    title = "Sentiment Counts in loughran",
    x = "Counts",
    y = "Sentiment"
  )


summarySentiment4 <- TMCSentiment4 %>%  count(word,sentiment,sort=TRUE) %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  arrange(n) %>%
  as.data.frame(stringsAsFactors=FALSE)

ggplot(summarySentiment4, aes(x = sentiment, y = n, fill=factor(sentiment)))
+
  geom_col() +
  coord_flip() +
  labs(
    title = "Sentiment Counts in NCR",
    x = "Counts",
    y = "Sentiment"
  )

# Sentiment per tweet
TMCTokenized2 <- tm_lemm
sent_tweet<- TMCTokenized2 %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(text, status_id) %>%
  summarize(sentiment = sum(value)) %>%
  arrange(sentiment)

tmobile_company3<-tmobile_company
tmobile_company3<- tmobile_company3 %>% left_join(sent_tweet, by='status_id')
%>%
  arrange(sentiment)
colnames(tmobile_company3)
sum(is.na(tmobile_company3$sentiment))
tmobile_company3 <-tmobile_company3%>%drop_na(sentiment)


save(tmobile_company3, file = "C:/Users/azambranollano/OneDrive -
IESEG/Alejandra/Social Media Analytics/Group Project/tmobile_company2.Rdata")
```

```r
# NCR - analysis per sentiment
as.data.frame(table(get_sentiments("nrc")$sentiment)) %>%
  arrange(desc(Freq))

trust <- get_sentiments("nrc") %>%
  filter(sentiment == "trust")
TMC_trust<-TMCTokenized2 %>%
  inner_join(trust) %>%
  count(word, sort = TRUE)
TMC_trust2<-TMC_trust
TMC_trust2["sentiment" <- "trust"]

surprise <- get_sentiments("nrc") %>%
  filter(sentiment == "surprise")
TMC_surprise<-TMCTokenized2 %>%
  inner_join(surprise) %>%
  count(word, sort = TRUE)
TMC_surprise2<-TMC_surprise
TMC_surprise2["sentiment"]<- "surprise"

sadness <- get_sentiments("nrc") %>%
  filter(sentiment == "sadness")
TMC_sadness<-TMCTokenized2 %>%
  inner_join(sadness) %>%
  count(word, sort = TRUE)
TMC_sadness2<-TMC_sadness
TMC_sadness2["sentiment"]<- "sadness"

positive <- get_sentiments("nrc") %>%
  filter(sentiment == "positive")
TMC_positive<-TMCTokenized2 %>%
  inner_join(positive) %>%
  count(word, sort = TRUE)
TMC_positive2<-TMC_positive
TMC_positive2["sentiment"]<- "positive"

negative <- get_sentiments("nrc") %>%
  filter(sentiment == "negative")
TMC_negative<-TMCTokenized2 %>%
  inner_join(negative) %>%
  count(word, sort = TRUE)
TMC_negative2<-TMC_negative
TMC_negative2["sentiment"]<- "negative"

joy <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")
TMC_joy<-TMCTokenized2 %>%
  inner_join(joy) %>%
  count(word, sort = TRUE)
```

```r
TMC_joy2<-TMC_joy
TMC_joy2["sentiment"]<- "joy"

fear <- get_sentiments("nrc") %>%
  filter(sentiment == "fear")
TMC_fear<-TMCTokenized2 %>%
  inner_join(fear) %>%
  count(word, sort = TRUE)
TMC_fear2<-TMC_fear
TMC_fear2["sentiment"]<- "fear"

disgust <- get_sentiments("nrc") %>%
  filter(sentiment == "disgust")
TMC_disgust<-TMCTokenized2 %>%
  inner_join(disgust) %>%
  count(word, sort = TRUE)
TMC_disgust2<-TMC_disgust
TMC_disgust2["sentiment"]<- "disgust"

anticipation <- get_sentiments("nrc") %>%
  filter(sentiment == "anticipation")
TMC_anticipation<-TMCTokenized2 %>%
  inner_join(anticipation) %>%
  count(word, sort = TRUE)
TMC_anticipation2<-TMC_anticipation
TMC_anticipation2["sentiment"]<- "anticipation"

anger <- get_sentiments("nrc") %>%
  filter(sentiment == "anger")
TMC_anger<-TMCTokenized2 %>%
  inner_join(anger) %>%
  count(word, sort = TRUE)
TMC_anger2<-TMC_anger
TMC_anger2["sentiment"]<- "anger"

company_sentiment <- rbind(TMC_trust2,TMC_surprise2,TMC_sadness2,
                           TMC_positive2,TMC_negative2,TMC_joy2,TMC_fear2,
                           TMC_disgust2,TMC_anticipation2,TMC_anger2)


company_sentiment["source"] <- "T-Mobile Tweets"


# Cloud per sentiment NCR
set.seed(1234)
wordcloud(TMC-anger$word,TMC_anger$n,min.freq = 1, colors=brewer.pal(8,
"Dark2"),random.order=FALSE, rot.per=0.35, shape = 'star', max.words=300)
wordcloud2(TMC_anticipation, color = "random-light", backgroundColor =
"white",size = 1, shape = 'cardioid')
```

```r
wordcloud2(TMC_disgust, color = "random-light", backgroundColor =
"white",size = 1, shape = 'diamond')
wordcloud2(TMC_fear, color = "random-light", backgroundColor = "white",size =
1, shape = 'triangle-forward')
wordcloud2(TMC_joy, color = "random-light", backgroundColor = "white",size =
1, shape = 'triangle')
wordcloud2(TMC_negative, color = "random-light", backgroundColor =
"white",size = 1, shape = 'pentagon')
wordcloud2(TMC_positive, color = "random-light", backgroundColor =
"white",size = 2, shape = 'star')
wordcloud2(TMC_surprise, color = "random-light", backgroundColor =
"white",size = 2, shape = 'circle')
wordcloud2(TMC_trust, color = "random-light", backgroundColor = "white",size
= 1, shape = 'star')

#########   Sentiment of "hash_lemm" - tweets from different users  #########

# Sentiment

hashTokenized1 <- inner_join(hash_lemm,get_sentiments("bing"))
hashTokenized2 <- inner_join(hash_lemm,get_sentiments("afinn"))
hashTokenized3 <- inner_join(hash_lemm,get_sentiments("loughran"))
hashTokenized4 <- inner_join(hash_lemm,get_sentiments("nrc"))

# Most positive/negative words

hashsummSent1 <- hashTokenized1 %>%  count(word,sentiment,sort=TRUE) %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  arrange(n) %>%
  as.data.frame(stringsAsFactors=FALSE)

ggplot(hashsummSent1, aes(x = sentiment, y = n, fill=factor(sentiment))) +
  geom_col() +
  coord_flip() +
  labs(
    title = "Sentiment Counts in bing",
    x = "Counts",
    y = "Sentiment"
  )

hashsummSent2 <- hashTokenized2 %>%  count(word,value,sort=TRUE) %>%
  group_by(value) %>%
  top_n(10) %>%
  arrange(n) %>%
  as.data.frame(stringsAsFactors=FALSE)

ggplot(hashsummSent2, aes(x = value, y = n, fill=factor(value))) +
  geom_col() +
```

```r
  coord_flip() +
  labs(
    title = "Sentiment Counts in afinn",
    x = "Counts",
    y = "Sentiment"
  )

hashsummSent3 <- hashTokenized3 %>%  count(word,sentiment,sort=TRUE) %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  arrange(n) %>%
  as.data.frame(stringsAsFactors=FALSE)

ggplot(hashsummSent3, aes(x = sentiment, y = n, fill=factor(sentiment))) +
  geom_col() +
  coord_flip() +
  labs(
    title = "Sentiment Counts in loughran",
    x = "Counts",
    y = "Sentiment"
  )

hashsummSent4 <- hashTokenized4 %>%  count(word,sentiment,sort=TRUE) %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  arrange(n) %>%
  as.data.frame(stringsAsFactors=FALSE)

ggplot(hashsummSent4, aes(x = sentiment, y = n, fill=factor(sentiment))) +
  geom_col() +
  coord_flip() +
  labs(
    title = "Sentiment Counts in ncr",
    x = "Counts",
    y = "Sentiment"
  )

# Sentiment per tweet
TMCTokenized3 <- hash_lemm
sent_tweet2<- TMCTokenized3 %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(text, status_id) %>%
  summarise(sentiment = sum(value)) %>%
  arrange(sentiment)

tmobile_users2<-tmobile_users
tmobile_users2<- tmobile_users2 %>% left_join(sent_tweet2, by='status_id')
%>%
  arrange(sentiment)
```

```r
colnames(tmobile_users2)
sum(is.na(tmobile_users2$sentiment))
tmobile_users2<-tmobile_users2%>%drop_na(sentiment)


save(tmobile_users2, file = "C:/Users/azambranollano/OneDrive -
IESEG/Alejandra/Social Media Analytics/Group Project/tmobile_users2.Rdata")


# NCR - analysis per sentiment
as.data.frame(table(get_sentiments("nrc")$sentiment)) %>%
  arrange(desc(Freq))

trusthash <- get_sentiments("nrc") %>%
  filter(sentiment == "trust")
hash_trust<-hashTokenized2 %>%
  inner_join(trust) %>%
  count(word, sort = TRUE)
hash_trust2<-hash_trust
hash_trust2["sentiment"]<- "trust"

surprisehash <- get_sentiments("nrc") %>%
  filter(sentiment == "surprise")
hash_surprise<-hashTokenized2 %>%
  inner_join(surprise) %>%
  count(word, sort = TRUE)
hash_surprise2<-hash_surprise
hash_surprise2["sentiment"]<- "surprise"

sadnesshash <- get_sentiments("nrc") %>%
  filter(sentiment == "sadness")
hash_sadness<-hashTokenized2 %>%
  inner_join(sadness) %>%
  count(word, sort = TRUE)
hash_sadness2<-hash_sadness
hash_sadness2["sentiment"]<- "sadness"

positivehash <- get_sentiments("nrc") %>%
  filter(sentiment == "positive")
hash_positive<-hashTokenized2 %>%
  inner_join(positive) %>%
  count(word, sort = TRUE)
hash_positive2<-hash_positive
hash_positive2["sentiment"]<- "positive"

negativehash <- get_sentiments("nrc") %>%
  filter(sentiment == "negative")
hash_negative<-hashTokenized2 %>%
  inner_join(negative) %>%
```

```
  count(word, sort = TRUE)
hash_negative2<-hash_negative
hash_negative2["sentiment"]<- "negative"

joyhash <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")
hash_joy<-hashTokenized2 %>%
  inner_join(joy) %>%
  count(word, sort = TRUE)
hash_joy2<-hash_joy
hash_joy2["sentiment"]<- "joy"

fearhash <- get_sentiments("nrc") %>%
  filter(sentiment == "fear")
hash_fear<-hashTokenized2 %>%
  inner_join(fear) %>%
  count(word, sort = TRUE)
hash_fear2<-hash_fear
hash_fear2["sentiment"]<- "fear"

disgusthash <- get_sentiments("nrc") %>%
  filter(sentiment == "disgust")
hash_disgust<-hashTokenized2 %>%
  inner_join(disgust) %>%
  count(word, sort = TRUE)
hash_disgust2<-hash_disgust
hash_disgust2["sentiment"]<- "disgust"

anticipationhash <- get_sentiments("nrc") %>%
  filter(sentiment == "anticipation")
hash_anticipation<-hashTokenized2 %>%
  inner_join(anticipation) %>%
  count(word, sort = TRUE)
hash_anticipation2<-hash_anticipation
hash_anticipation2["sentiment"]<- "anticipation"

angerhash <- get_sentiments("nrc") %>%
  filter(sentiment == "anger")
hash_anger<-hashTokenized2 %>%
  inner_join(anger) %>%
  count(word, sort = TRUE)
hash_anger2<-hash_anger
hash_anger2["sentiment"]<- "anger"

user_sentiment <- rbind(hash_trust2,hash_surprise2,hash_sadness2,
                        hash_positive2,hash_negative2,hash_joy2,hash_fear2,
                        hash_disgust2,hash_anticipation2,hash_anger2)
user_sentiment["source"] <- "Users Tweets"
```

```
# Combining the sentiment file 'NCR' in 1 file
total_sentiment <- rbind(user_sentiment, company_sentiment)

save(total_sentiment, file = "C:/Users/azambranollano/OneDrive -
IESEG/Alejandra/Social Media Analytics/Group Project/total_sentiment.Rdata")


# Cloud per sentiment NCR
set.seed(1234)
wordcloud(hash_anger$word,hash_anger$n,min.freq = 1, colors=brewer.pal(8,
"Dark2"),random.order=FALSE, rot.per=0.35, shape = 'star', max.words=300)
wordcloud2(hash_anticipation, color = "random-light", backgroundColor =
"white",size = 1, shape = 'cardioid')
wordcloud2(hash_disgust, color = "random-light", backgroundColor =
"white",size = 1, shape = 'diamond')
wordcloud2(hash_fear, color = "random-light", backgroundColor = "white",size
= 1, shape = 'triangle-forward')
wordcloud2(hash_joy, color = "random-light", backgroundColor = "white",size =
1, shape = 'triangle')
wordcloud2(hash_negative, color = "random-light", backgroundColor =
"white",size = 1, shape = 'pentagon')
wordcloud2(hash_positive, color = "random-light", backgroundColor =
"white",size = 2, shape = 'star')
wordcloud2(hash_surprise, color = "random-light", backgroundColor =
"white",size = 2, shape = 'circle')
wordcloud2(hash_trust, color = "random-light", backgroundColor = "white",size
= 1, shape = 'star')
```

Topic Modeling

For topic modeling of tweets from the company, we used the official Twitter handles of the company as placeholders for documents to see what topics emerge from the different official handles of the company and whether they fall in any particular category. For tweets from the users, we used the platform used by users for analysis of topics.

Challenges

It took multiple iterations with different combinations to determine the ideal grouping as analysing topics per tweet was giving too broad results and no apparent analytical viewpoint.

Determining topics count

After doing the necessary preprocessing (cleaning, tokenization and lemmatization), we ran the ldatuning model to determine the optimum number of topics. The available metrics in the ldatuning model showed 3 as a fair number of topics for tweets from the company and 4 for tweets from users.

Modeling

We used the lda function to determine the topics using the term frequency and keeping the seed set at the same number as the ldatuning function.

Results

We can see the list of top topics below. For detailed analysis of the top topics and their occurance in each of the handles, please refer to the dashboard.

```
################################################
################ Topic Modeling ################
################################################


# Create the DTM

tm_DTM <- tm_lemm %>%
  count(screen_name, word, sort = TRUE) %>%
  cast_dtm(document = screen_name, term = word, value = n, weighting =
tm::weightTf)

hash_DTM_reply <- hash_lemm %>%
  count(source2, word, sort = TRUE) %>%
  cast_dtm(document = source2, term = word, value = n, weighting =
tm::weightTf)

# Find the optimal number of topics using LDA tuning

result <- FindTopicsNumber(
  tm_DTM,
  topics = seq(from = 2, to = 15, by = 1),
  metrics = c("Griffiths2004", "CaoJuan2009", "Arun2010", "Deveaud2014"),
  method = "Gibbs",
  control = list(seed = 77),
  mc.cores = 4L,
  verbose = TRUE
)
FindTopicsNumber_plot(result)

hash_result <- FindTopicsNumber(
  hash_DTM_reply,
  topics = seq(from = 2, to = 15, by = 1),
  metrics = c("Griffiths2004", "CaoJuan2009", "Arun2010", "Deveaud2014"),
  method = "Gibbs",
  control = list(seed = 77),
  mc.cores = 4L,
  verbose = TRUE
)
```

```r
FindTopicsNumber_plot(hash_result)


tweets_lda <- LDA(tm_DTM, k = 3,method="gibbs",control = list(seed = 77))
tweets_lda

save(tweets_lda, file = "C:/Users/azambranollano/OneDrive -
IESEG/Alejandra/Social Media Analytics/Group Project/tmob_LDA.RData")


hashreply_tweets_lda <- LDA(hash_DTM_reply, k = 4,method="gibbs",control =
list(seed = 77))
hashreply_tweets_lda

save(hashreply_tweets_lda, file = "C:/Users/azambranollano/OneDrive -
IESEG/Alejandra/Social Media Analytics/Group Project/users_LDA.RData")

# Isolate the topics stored in "beta"

tweet_topics <- tidy(tweets_lda, matrix = "beta")

hashreply_topics <- tidy(hashreply_tweets_lda, matrix = "beta")

# Get the top terms per topic
top_tweet_terms <- tweet_topics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
top_tweet_terms

top_hashreply_terms <- hashreply_topics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
top_hashreply_terms

# Plot the top terms per topic

top_tweet_terms %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip() +
  scale_x_reordered()

t(topics(tweets_lda,4))
```

```r
top_hashreply_terms %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip() +
  scale_x_reordered()

t(topics(hashreply_tweets_lda,4))


# Which company handles are the most active

days_since <- tmobile_company %>%
  mutate(diff = as.numeric((difftime(max(as.Date.POSIXct(created_at)),
created_at))/(24*60))) %>%
  group_by(screen_name) %>%
  arrange(desc(diff)) %>%
  slice(1) %>%
  select(screen_name, diff)

tmobile_company %>%
  group_by(screen_name) %>%
  count() %>%
  left_join(days_since) %>%
  mutate(monthly_freq = round(n/diff,1))

# Engagement statistics for

tmobile_company %>%
  group_by(screen_name) %>%
  summarise(sum(favorite_count), sum(retweet_count))

# Handling user tweets

tmobile_company %>%
  filter(!is.na(reply_to_status_id)) %>%
  group_by(screen_name) %>%
  count(sort = TRUE)
```