Strategy and Development Review | Volume: 11 / N°: 01 (January 2021), p 447 -461

Sentiment Analysis Using R: Applied Study of the Mobile Phone Market

تحليل المشاعر باستخدام R: دراسة تطبيقية لسوق الهاتف المحمول

Balouli Houssame Eddine[1], Chine Lazhar[2]

[1] University Algiers 3 (Algeria), balouli.houssameeddine@univ-alger3.dz
[2]Boumerdes University (Algeria), l.chine@univ-boumerdes.dz

Received:01/09/2020 ***Accepted: 13/12/2020*** ***Published:01/01/2021***

Abstract:

This study aims to analyze the sentiments of social networking platform Twitter users using the programming language R about famous mobile phones brands. Using different packages of the R programming language, 600 tweets were obtained from Twitter users around the world randomly about mobile phones brands: Nokia, Oppo, Samsung, Apple, Huawei and LG (100 tweets per brand). After that, tweets are filtered from special characters, numbers, repeated words, as well as any other type of data. As a first stage, we create and analyze the word cloud for each brand. The final step is to create graphs of the feelings of the tweeters that are divided into five categories: anger, loathing, joy, surprise, and positivity. As a result, sentiment analysis allows us to realize a market segmentation to mobile phone market.

Keywords: Sentiment Analysis; Mobile phone; Twitter; R programming language.

JELClassificationCodes: C81, D85

ملخص

تهدف هذه الدراسة إلى تحليل مشاعر مستخدمي منصة التواصل الاجتماعي تويتر باستخدام لغة البرمجة آر حول موضوع الماركات العالمية المعروفة للهواتف النقالة. باستخدام حزم مختلفة للغة البرمجة آر، تم الحصول على 600 تغريدة لمستخدمي منصة تويتر من مختلف دول العالم بطريقة عشوائية حول أهم الماركات العالمية للهواتف النقالة: نوكيا، أوبو، سامسونغ، آبل، هواوي وآل جي (100 تغريدة لكل

[1] **Corresponding author**: balouli Houssame Eddine, **e-mail:** balouli.houssame.eddine@gmail.com

ماركة). بعد ذلك، يتم تنقية التغريدات من الرموز الخاصة، الأرقام، الكلمات المتكررة، بالإضافة إلى أي نوع آخر من أنواع البيانات. كمرحلة أولى، نقوم بإنشاء سحابة الكلمات الخاصة بكل ماركة وتحليلها. الخطوة النهائية تتمثل في إنشاء مدرجات تكرارية لمشاعر المغردين والتي تم تقسيمها إلى 5 أنواع: الغضب، الاشمئزاز، الفرح، المفاجأة، والإيجابية. كنتيجة للدراسة: من خلال تحليل مشاعر مستخدمي منصة التواصل الاجتماعي تويتر نستطيع القيام بما يسمى تجزئة السوق.

كلمات مفتاحية: تحليل المشاعر،الهاتف النقال،تويتر،لغة البرمجة آر.

تصنيفاتJEL : C81، D85.

1. INTRODUCTION

Social media platforms have become an important source of information because millions around the world use it to exchange ideas, express their psychological state, participate in political life, as well as publish their photos, videos and various daily activities, etc., all for free (Pandey, Rajpoot, & Saraswat, 2017, p. 764)

Among the most important types of data on which social media platforms are available: texts. These texts express the feelings of the users of these platforms, and indicate their personalities, their psychological state, and can even indicate their age, their jobs, their family and social status (Sailunaz & Alhajj, 2019, p. 1).

Twitter is one of the most important social media platforms with more than 340 million monthly active users. The 140-word feature makes its data easy to use and analyze (Ansari, Aziz, Siddiqui, Mehra, & Singh, 2020, p. 1821).

1.1 literature review

Several studies have analyzed the sentiments of Twitter users for various purposes: politics, economics, society, education, sport, art, medicine, and others.

Sharma and Ghose (2020) studied the sentiments of Twitter users in India from January 2019 to March 2019 on the occasion of the general elections in India. The results obtained were the same as the real results for the elections (Sharma & Ghose, 2020).

In the same context of voting and elections, Terán and Mancera (2019) proposed an improvement to the Voting Advice Applications for the case of Ecuadorian national elections in order to contribute to the mitigation of the bias of electoral candidates by analyzing sentiment twitter users in Ecuador (Terán & Mancera, 2019).

Regarding critical events such as disasters and wars, we can analyze Twitter users' sentiment in order to improve the decision making process during crises.

Ruz & Henríquez & Mascareño (2020) studied Twitter users sentiments related to two events: 2010 Chilean earthquake and 2017 Catalan independence referendum. The objective is the comparison between several classifiers regarding the search language (Ruz, Henríquez, & Mascareño, 2020).

Similarly, Öztürk & Ayvaz (2018) analyzed Twitter users' sentiments about the Syrian refugee crisis in two languages: Turkish and English. The comparison between the sentiment of Turkish users and the sentiment of English users shows that the first are positive while the second are neutral (Öztürk & Ayvaz, 2018).

All the previous studies used the open source programming language R to extract the tweets. The difference is in the model used in the analysis of the extracted tweets; there are those who used ready-made R packages, and there are those who compare several models and choose the best, while others analyze only the impact of tweets (positive, negative, neutral) on specific topic.

In our case, we used many R packages to first extract tweets about mobile phone brands randomly (English language only) and analyze the impact by classify it in 5 categories: Anger, loathing, Joy, Surprise, and Positivity. In the next section we present the followed approache from the extraction of tweets to the sentiment analysis.

3. METHOD & RESULTS & DISCUSSION

3.1Method

The first step before the extraction of tweets is the creation of connection to twitter server. For this, we must have authentication keys that can be obtained by registering on the development site (https://developer.twitter.com/en). The process is not very complex :

Fig.1. Authentication keys

```
consumer_key <- "dJj2xh6pg6nG3goFh6MywnASy"
consumer_secret <- "oB2dKlsQEa40eyhBXvEnSuqse4M3LAVmFeSeAsHoOTBBgoiqVf"
access_token <- "1013074431550291968-dACoYr3pt1NG1BBkIxoyzCp4Sh0f8R"
access_secret <- "g4ZyGnjGk8v35khMswg5fPzg3a9KwZIpSj8HCdPYwZFPW"
```

Source: obtained after registering on the development site

The next step is the installation and the loading of the necessary packages: “twitter” package (Gentry, Gentry, RSQLite, & Artistic, 2016), “tm” package (Feinerer, Hornik, & Feinerer, 2015), “Tidyverse” package (Wickham & Wickham, 2017), “wourdcloud” package (Fellows, Fellows, & Rcpp, 2018), and “syuzhet” package (Jockers, 2017):

Fig.2. Necessary packages

```
library(twitteR)
library(tm)
library(tidyverse)
library(syuzhet)
library(wordcloud)
```

Source: script screenshot

We specify after that the string consumer_key, consumer_secret, access_token, and access_secret in the setup_twitter_oauth () command:

Fig.3. Access to Twitter server

```
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secre
t)

[1] "Using direct authentication"
```

Source: script screenshot

The message “Using direct authentication” should appear in the console, indicating that the operation is running smoothly. Now, we are ready for the extraction of tweets.

The searchTwitter () function is used to load tweets online. In our example, we specify seven keywords. We limit the number of messages extracted to n = 100, and from 01/01/2018. We are interested in English language documents only :

Fig.4. Loading tweets script

```
tweets_Apple <- searchTwitter("@Apple",n = 100, since = '2018-01-01', lang =
"en")
tweets_Samsung <- searchTwitter("@SamsungMobile",n= 100, since = '2018-01-01'
, lang = "en")
tweets_Huawei <- searchTwitter("@Huawei",n= 100, since = '2018-01-01', lang =
"en")
tweets_Oppo <- searchTwitter("@oppo",n= 100, since = '2018-01-01', lang = "en
")
tweets_Sony <- searchTwitter("@Sony",n= 100, since = '2018-01-01', lang = "en
")
tweets_LG <- searchTwitter("@LGUSAMobile",n= 100, since = '2018-01-01', lang
= "en")
tweets_Nokia <- searchTwitter("@nokia",n= 100, since = '2018-01-01', lang = "
en")
```

Source: script screenshot

3.1Results

We can show some examples of the extracted tweets using 'print' function :

Fig.5. Loading tweets examples

```
print(tweets_Apple[[1]])
[1] "QaisarahAmanda: RT @Apple: Guitar, mic or monitor: plug your favourite d
evices into the new iPad Pro with USB-C."
print(tweets_Samsung[[1]])
[1] "folol9744: RT @anthonykudaev: I want to buy @SamsungMobile #S9.. What gu
ys u think about it <U+0001F61C> or continue using #apple"
print(tweets_Huawei[[1]])
[1] "dataeconomy: Congrats to Sun Yafang who was named in the #Power200, The
World's Most Influential Data Economy Leaders @Huawei 1⊠ https://t.co/jXA5VTa
DxR"
```

Source: script screenshot

We need to clean the extracted tweets by saving only texts. These changes are necessary for the next stage of analysis:

The cleaning process starts with the extraction of texts only:

Fig.6. extraction of texts from tweets

```
Apple_text <- sapply(tweets_Apple, function(x) x$getText())
Samsung_text <- sapply(tweets_Samsung, function(x) x$getText())
Huawei_text <- sapply(tweets_Huawei, function(x) x$getText())
Oppo_text <- sapply(tweets_Oppo, function(x) x$getText())
Sony_text <- sapply(tweets_Sony, function(x) x$getText())
LG_text <- sapply(tweets_LG, function(x) x$getText())
Nokia_text <- sapply(tweets_Nokia, function(x) x$getText())
```

Source: script screenshot

After that, we create a corpus for each type. Then, we remove punctuation marks using 'removePunctuation' function:

Fig.7. remove punctuation marks

```
Apple_clean <- tm_map(Apple_corpus, removePunctuation)
Samsung_clean <- tm_map(Samsung_corpus, removePunctuation)
Huawei_clean <- tm_map(Huawei_corpus, removePunctuation)
Oppo_clean <- tm_map(Oppo_corpus, removePunctuation)
Sony_clean <- tm_map(Sony_corpus, removePunctuation)
LG_clean <- tm_map(LG_corpus, removePunctuation)
Nokia_clean <- tm_map(Nokia_corpus, removePunctuation)
```

Source: script screenshot

In addition to removing punctuation marks, me clean out data also from numbers using ‘removeNumbers’ function:

Fig.8. remove numbers

```
Apple_clean <- tm_map(Apple_clean, removeNumbers)
Samsung_clean <- tm_map(Samsung_clean, removeNumbers)
Huawei_clean <- tm_map(Huawei_clean, removeNumbers)
Oppo_clean <- tm_map(Oppo_clean, removeNumbers)
Sony_clean <- tm_map(Sony_clean, removeNumbers)
LG_clean <- tm_map(LG_clean, removeNumbers)
Nokia_clean <- tm_map(Nokia_clean, removeNumbers)
```

Source: script screenshot

In the same way, we remove many English words: pronouns, transition words, the verb ‘to be’ in the past, present and future, and other English words that do not help us in the analysis:

Fig.9. English language cleaning

```
Apple_clean <- tm_map(Apple_clean, removeWords, stopwords("english"))
Samsung_clean <- tm_map(Samsung_clean, removeWords, stopwords("english"))
Huawei_clean <- tm_map(Huawei_clean, removeWords, stopwords("english"))
Oppo_clean <- tm_map(Oppo_clean, removeWords, stopwords("english"))
Sony_clean <- tm_map(Sony_clean, removeWords, stopwords("english"))
LG_clean <- tm_map(LG_clean, removeWords, stopwords("english"))
Nokia_clean <- tm_map(Nokia_clean, removeWords, stopwords("english"))
```

Source: script screenshot

We can show the list of the removed words:

Fig.10. Removed English Words

```
stopwords()
  [1] "i"          "me"         "my"           "myself"     "we"
  [6] "our"        "ours"       "ourselves"    "you"        "your"
 [11] "yours"      "yourself"   "yourselves"   "he"         "him"
 [16] "his"        "himself"    "she"          "her"        "hers"
 [21] "herself"    "it"         "its"          "itself"     "they"
 [26] "them"       "their"      "theirs"       "themselves" "what"
 [31] "which"      "who"        "whom"         "this"       "that"
 [36] "these"      "those"      "am"           "is"         "are"
 [41] "was"        "were"       "be"           "been"       "being"
 [46] "have"       "has"        "had"          "having"     "do"
 [51] "does"       "did"        "doing"        "would"      "should"
 [56] "could"      "ought"      "i'm"          "you're"     "he's"
 [61] "she's"      "it's"       "we're"        "they're"    "i've"
 [66] "you've"     "we've"      "they've"      "i'd"        "you'd"
 [71] "he'd"       "she'd"      "we'd"         "they'd"     "i'll"
 [76] "you'll"     "he'll"      "she'll"       "we'll"      "they'll"
 [81] "isn't"      "aren't"     "wasn't"       "weren't"    "hasn't"
 [86] "haven't"    "hadn't"     "doesn't"      "don't"      "didn't"
 [91] "won't"      "wouldn't"   "shan't"       "shouldn't"  "can't"
 [96] "cannot"     "couldn't"   "mustn't"      "let's"      "that's"
[101] "who's"      "what's"     "here's"       "there's"    "when's"
[106] "where's"    "why's"      "how's"        "a"          "an"
```

Source: script screenshot (Rstudio output)

Once our data become clean, we move to the visualization of the word cloud. The word cloud is a graphical representation of the most important words in a text, in which the word size is directly proportional to its frequency in the text (Jin, 2017, p. 788). We start with Apple word cloud:

Fig.11. Script & Apple Word Cloud

```
Apple <- wordcloud(Apple_clean, random.order = F, max.words = 50, scale = c(3, 0.5),
colors = brewer.pal(8, 'Dark2'))
```

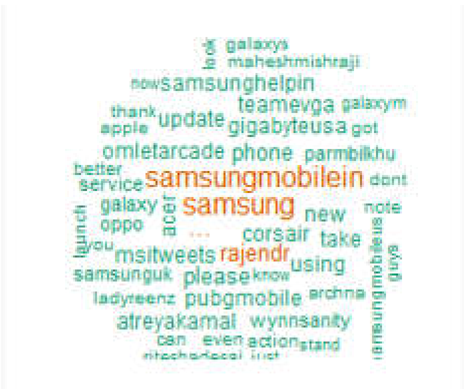
Source: script screenshot (Rstudio output)

Among the most important words in the word cloud is the word ‘**IPad**’, which represents Apple's most popular device. In addition, we find the word ‘**AppStore**’ which used to download Apps and Games. On the other hand, we find the word ‘**screen**’, which may refers to the quality of the screen of Apple phones. There is other significant word ‘**IOS**’ that represents Apple’s operating system.

We do the same thing to Samsung:

Fig.12. Script & Samsung Word Cloud

```
Samsung <- wordcloud(Samsung_clean, random.order = F, max.words = 50, scale = c(3,
0.5), colors = brewer.pal(8, 'Dark2'))
```



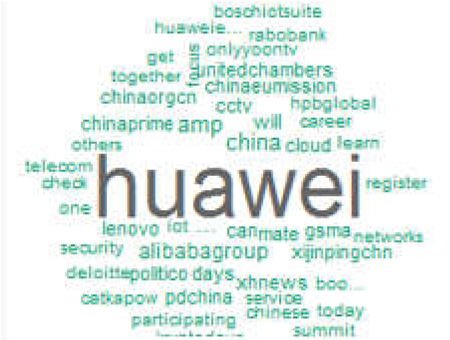
Source: script screenshot (Rstudio output)

Through Samsung word cloud, we notice that there are not many significant words; we have only two: ‘**Galaxys**’ and ‘**Galaxym**’, which refers to Samsung's most popular devices.

Also, we have Huawei word cloud:

Fig.13. Script & Huawei Word Cloud

```
Huawei <- wordcloud(Huawei_clean, random.order = F, max.words = 50, scale = c(3,
0.5), colors = brewer.pal(8, 'Dark2'))
```



Source: script screenshot (Rstudio output)

We have three significant words: '**security**' that may refer to the high level of security that characterizes Huawei devices, or the threat of espionage that European and American countries warn of because of the fifth generation technologies. The second, '**IOT**' that represents a branch of Huawei Company. The third word is '**China**' the Country of origin for Huawei.

After that, we have Oppo word cloud:

Fig.14. Script & Oppo Word Cloud

```
Oppo <- wordcloud(Oppo_clean, random.order = F, max.words = 50, scale = c(3, 0.5),
colors = brewer.pal(8, 'Dark2'))
```

Source: script screenshot (Rstudio output)

Oppo word cloud contain many significant word; '**color**', '**camera**', '**describe**', '**dream**' and '**performances**' that may refers to the high quality of Oppo's devices**.**

Regarding Sony, we have the next word could:

Fig.15. Script & Sony Word Cloud

```
Sony <- wordcloud(Sony_clean, random.order = F, max.words = 50, scale = c(3, 0.5),
colors = brewer.pal(8, 'Dark2'))
```

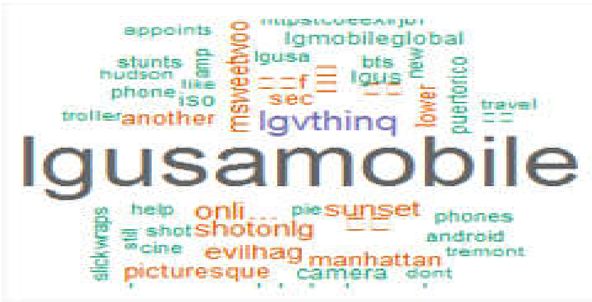
Source: script screenshot (Rstudio output)

In the case of Sony, the word '**playstation**' appear as a significant word (Sony's most popular device).

Moving to LG, in below the word cloud:

Fig.16. Script & LG Word Cloud

```
LG <- wordcloud(LG_clean, random.order = F, max.words = 50, scale = c(3, 0.5), colors
= brewer.pal(8, 'Dark2'))
```



Source: script screenshot (Rstudio output)

Similarly to Oppo, we have many significant words that may refer to screen quality: '**pictures**' '**sunset**', and '**camera**'.

The last word cloud is Nokia word cloud:

Fig.17. Script & Nokia Word Cloud

```
Nokia <- wordcloud(Nokia_clean, random.order = F, max.words = 50, scale = c(3, 0.5),
colors = brewer.pal(8, 'Dark2'))
```

Source: script screenshot (Rstudio output)

‘**Technologies**’, ‘**innovative**’, ‘**discover**’ and ‘**transform**’ are the most significant words about Nokia.

Generally, the word cloud is a primary way to get a general idea on any topic. However, for many complex topics, the results obtained are incomplete. We need to go deeper in the analysis.

In the next part, we present the sentiment analysis.

The first step is to obtain sentiment scores from the cleaning data using ‘get_nrc_sentiment’ function:

Fig.18. Sentiment scores

```
Apple_sentiment <- get_nrc_sentiment(Apple_text)
Samsung_sentiment <- get_nrc_sentiment(Samsung_text)
Huawei_sentiment <- get_nrc_sentiment(Huawei_text)
Oppo_sentiment <- get_nrc_sentiment(Oppo_text)
Sony_sentiment <- get_nrc_sentiment(Sony_text)
LG_sentiment <- get_nrc_sentiment(LG_text)
Nokia_sentiment <- get_nrc_sentiment(Nokia_text)
```

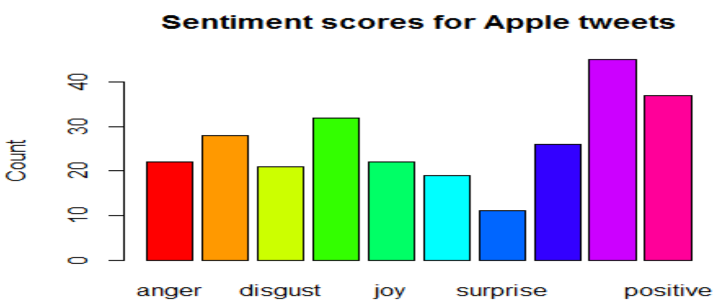
Source: script screenshot

‘Syuzhet’ package classify words in 5 categories: anger, loathing, joy, surprise, and positivity (dictionary that classify english words in 5 categories). There are other packages (dictionaries) such as ‘SentimentAnalysis’ package (Feuerriegel, Proellochs, & Feuerriegel, 2018), and ‘sentimentr’ package (Rinker, 2017).

The next step is the creation of bar plot of sentiment scores. Regarding Apple, in below the result :

Fig.19. Bar plot of Apple Sentiment scores

```
barplot(colSums(Apple_sentiment), ylab = 'Count', col = rainbow(10), main = '
Sentiment scores for Apple tweets')
```



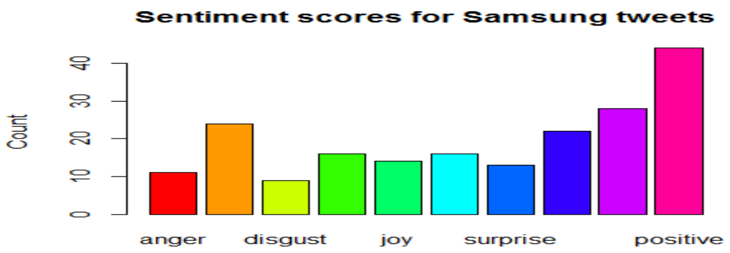
Source: script screenshot (Rstudio output)

Through the figure, we notice that most of the users ’sentiments are positive, with anger growing.

In the case of Samsung, we have the next bar plot:

Fig.20. Bar plot of Samsung Sentiment scores

```
barplot(colSums(Samsung_sentiment), ylab = 'Count', col = rainbow(10), main =
'Sentiment scores for Samsung tweets')
```



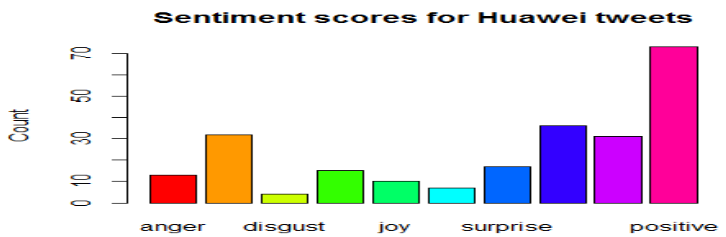
Source: script screenshot (Rstudio output)

Users ’sentiments for Samsung are completely positive, while the rest of the sentiments are stable.

Then, we move to Huawei:

Fig.21. Bar plot of Huawei Sentiment scores

```
barplot(colSums(Huawei_sentiment), ylab = 'Count', col = rainbow(10), main =
'Sentiment scores for Huawei tweets')
```



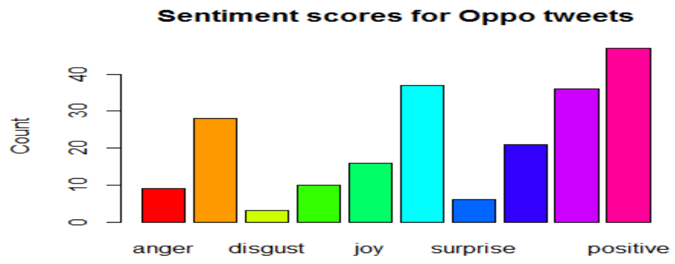
Source: script screenshot (Rstudio output)

Users ’sentiments for Huawei are similar to them of Samsung with lower levels of anger and disgust. These sentiments are perfect, and they really reflect the spread of this brand in the market.

We do the same for Oppo:

Fig.22. Bar plot of Oppo Sentiment scores

```
barplot(colSums(Oppo_sentiment), ylab = 'Count', col = rainbow(10), main = 'S
entiment scores for Oppo tweets')
```



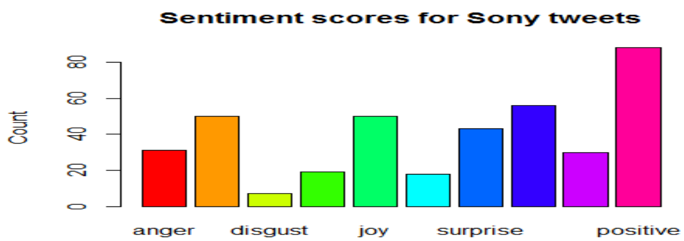
Source: script screenshot (Rstudio output)

Users 'sentiments for Oppo are positive with great enjoyment. While sentiments of anger and disgust are slight and do not affect.

Sony is the next:

Fig.23. Bar plot of Sony Sentiment scores

```
barplot(colSums(Sony_sentiment), ylab = 'Count', col = rainbow(10), main = 'S
entiment scores for Sony tweets')
```

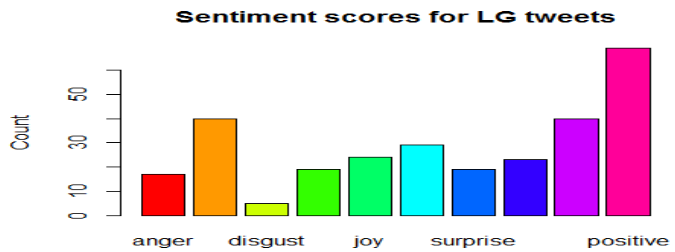


Source: script screenshot (Rstudio output)

Users 'sentiments are a mixture of anger, joy, surprise and positive with a slight advance for positive sentiments. We can say the same thing to LG and Nokia. But, Sony joy sentiments are the most obvious.

Fig.24. Bar plot of LG Sentiment scores

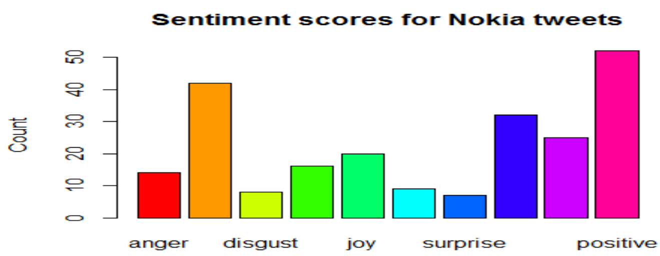
```
barplot(colSums(LG_sentiment), ylab = 'Count', col = rainbow(10), main = 'Sen
timent scores for LG tweets')
```



Source: script screenshot (Rstudio output)

Fig.25. Bar plot of Nokia Sentiment scores

```
barplot(colSums(Nokia_sentiment), ylab = 'Count', col = rainbow(10), main = '
Sentiment scores for Nokia tweets')
```



Source: script screenshot (Rstudio output)

3.3Discussion

Most of the users 'sentiments are positive for all brands, with the superiority of Asian brands due to the presence of feelings of enjoyment and surprise in one hand, and the presence of anger and disgust feelings regarding European and American brands in the other hand.

The results of sentiment analysis confirm the initial impressions of the word cloud analysis but with more precision by the classification of the sentiments on 5 categories: anger, disgust, joy, surprise and positive.

The results of the sentiment analysis allows decision makers, managers, customers, producers and other actors by take into consideration the sentiment analysis to improve their plans and attitudes in the future, in addition to the results of The usual methods of analysis and forecasting.

This work is not intended to publicize one of the brands at the expense of another, but it is a modern way to analyze the feelings of Twitter users about a particular topic.

Regarding deficiencies and the limits of the study, 100 tweets per brand remain insufficient to analyze user sentiments, because the Twitter administration has placed restrictions on accessing tweets after the issue of allegations of Russian interference in the US presidential election.

4. CONCLUSION

As an important source of data, social media such as twitter is a base for many studies and research. Among the most important recent studies is the analysis of user sentiments based on the extracted tweets about several topics.

Using the open source programming language R, we extracted tweets of more than 600 users around the world about the most famous mobile phone brands in the world in order to analyze the sentiments of these users.

The results of the sentiment analysis allows us realize a market segmentation, understand the trend of market, and improve our decision making by understand customer behavior, and guide them well.

5. BIBLIOGRAPHY

1. Ansari, M. Z., Aziz, M. B., Siddiqui, M. O., Mehra, H., & Singh, K. P. (2020). Analysis of political sentiment orientations on twitter. Procedia Computer Science , 167, 1821-1828.
2. Feinerer, I., Hornik, K., & Feinerer, M. I. (2015). Package ‘tm’. Corpus .
3. Fellows, I., Fellows, M. I., & Rcpp, L. (2018). Package ‘wordcloud’.
4. Feuerriegel, S., Proellochs, N., & Feuerriegel, M. S. (2018). Package ‘SentimentAnalysis’. CRAN .
5. Gentry, J., Gentry, M. J., RSQLite, S., & Artistic, R. L. (2016). Package ‘twitteR’. R package version .
6. Jin, Y. (2017). Development of word cloud generator software based on python. Procedia engineering , 174, 788-792.
7. Jockers, M. (2017). Package ‘syuzhet’.
8. Öztürk, N., & Ayvaz, S. (2018). Sentiment analysis on Twitter: A text mining approach to the Syrian refugee crisis. Telematics and Informatics , 35 (1), 136-147.
9. Pandey, A. C., Rajpoot, D. S., & Saraswat, M. (2017). Twitter sentiment analysis using hybrid cuckoo search method. Information Processing & Management , 53 (4), 764-779.

10. Rinker, T. (2017). Package 'sentimentr'. Retrieved .
11. Ruz, G. A., Henríquez, P. A., & Mascareño, A. (2020). Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers. Future Generation Computer Systems , 106, 92-104.
12. Sailunaz, K., & Alhajj, R. (2019). Emotion and sentiment analysis from Twitter text. Journal of Computational Science , 36, 101003.
13. Sharma, A., & Ghose, U. (2020). Sentimental Analysis of Twitter Data with respect to General Elections in India. Procedia Computer Science , 173, 325-334.
14. Terán, L., & Mancera, J. (2019). Dynamic profiles using sentiment analysis and twitter data for voting advice applications. Government Information Quarterly , 36 (3), 520-535.
15. Wickham, H., & Wickham, M. H. (2017). Package tidyverse. Easily Install and Load the 'Tidyverse.