Proyecto Final: Clasificación de tweets basado en el sentimiento.

Antonio Godoy, Diego Mendoza, Jesús Campos

Resumen—En los últimos años, el análisis de sentimiento ha sido una de las áreas que más fuerza ha tomado, debido a la producción masiva de comentarios o publicaciones por parte de usuarios en redes sociales, blogs, y otros sitios web. Asimismo, ha representado una fuente importante de información para las grandes empresas y organizaciones, puesto que representa el punto de partida para conocer lo que los consumidores estan buscando, así como para conocer su reputación frente a ellos. Este documento busca abordar una comparación entre el Galaxy Note 8, el iPhone X, y el Pixel 2, a través de un proyecto de clasificación de publicaciones en Twitter (a partir de ahora tweets), basándose en un criterio de sentimiento y emociones.

Index Terms—R, RStudio, Aprendizaje máquina, Clasificador Bayesiano, Análisis de Sentimientos, Minería de Opinion, Minería de Datos, Twitter.

1. Introducción

En este documento se muestran los resultados obtenidos en el proyecto final elaborado para el curso de "Aprendizaje de Máquina", impartida por el Dr. Ulises Orozco Rosas en CETYS Universidad Campus Tijuana. El objetivo principal de éste reporte es el de dar a conocer el desenlace de la aplicación del proyecto final, que consiste en la clasificación de tweets sobre el iPhone X, el Galaxy Note 8, y el Pixel 2, de acuerdo con la aplicación de un análisis de sentimiento. Asimismo, se muestran los requerimientos y su implementación en el lenguaje de programación R.

2. Descripción del Problema

El problema a tratar específicamente es la clasificación de tweets en la red social Twitter, de acuerdo a un criterio basado en el sentimiento. El proyecto se enfoca en el "data mining"dentro de Twitter. La idea principal es la de buscar tweets que contengan información sobre el iPhone X, el Galaxy Note 8, y el Pixel 2, y clasificarlos de acuerdo a tres categorías de sentimiento, positivo, neutral, y negativo. El modelo de clasificación de tweets se haría de acuerdo a las palabras contenidas en el mismo. Cada palabra tendrá una categoría de subjetividad asignada (positivo, negativo o neutral), dándo como resultado un valor numérico representando su grado de positividad o negatividad. Asimismo, se tendrá la posibilidad de agrupar tweets de acuedo a la emoción que los caracteriza. La máquina, dado un conjunto de tweets y librerías de apoyo para el análisis de los mismos, agrupará dichos tweets en las categorías ya mencionadas.

3. Objetivos

Los objetivos planteados para este proyecto se muestran a continuación:

- Crear una aplicación de Twitter y autenticarlo mediante código fuente en el lenguaje de programación R.
- Extraer tweets desde la red social de Twitter.
- Depurar el contenido de los tweets, eliminando caracteres extraños y palabras irrelevantes en el análisis.
- Obtener las palabras con mayor frecuencia de aparición en los tweets.
- Representar gráficamente las palabras con mayor frecuencia (nube de palabras, gráficas de barras, etc.).
- Ligar un conjunto de datos que contengan las palabras positivas y negativas, con las cuales se le dará valor a los tweets.
- Crear un método/función que consiga asociar el contenido de los tweets con las palabras del conjunto de datos correspondiente.
- Normalizar los valores o resultados obtenidos de los tweets a una "calificación" estandarizada.
- Ligar la "calificación" con la clasificación de sentimientos (Positivo, neutro, negativo).
- Clasificar los tweets de acuerdo al sentimiento predominante, utilizando una función apoyada en el clasificador Bayesiano.
- Clasificar los tweets de acuerdo a la emoción predominante (miedo, enojo, alegría, etc.), utilizando el clasificador mencionado.
- De acuerdo a una cierta palabra, o a un grupo de palabras, y a las categorías mencionadas anteriormente, proponer soluciones a problemas encontrados por las palabras negativas. Por ejemplo, sugerir una solución a la opinión pública que cree que el iPhone X es muy caro, o que ventajas/desventajas presenta sobre otros dispositivos.

4. Marco Teórico

- **API (Application Programming Interface):** Una API es un marco que facilita la creación de servicios HTTP disponibles para una amplia variedad de clientes, entre los que se incluyen exploradores y dispositivos móviles.
- **Análisis de Sentimiento:** Es el proceso donde la máquina por medio de algoritmos de clasificación y regresión

Antonio Godoy, Diego Mendoza, Jesús Campos, son estudiantes de Licenciatura en Ingeniería en Ciencias Computacionales en CETYS Universidad Campus Tijuana.

(para este caso), determina si una frase contiene alguna opinión, positiva, negativa o neutral.

- **Minería de Opinión:** Serie de aplicaciones de técnicas del procesamiento del lenguaje natural, lingüística computacional y minería de textos, que tienen como objetivo la extracción de información subjetiva a partir de contenidos generados por los usuarios, ejemplo:Facebook, Twitter,etc.
- **Minería de Datos:** Es el conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos, de manera automática o semiautomática, con el objetivo de encontrar patrones repetitivos, tendencias o reglas que expliquen el comportamiento de los datos en un determinado contexto.
- **Nube de Palabras:** Es la representación visual de las palabras e oraciones que conforman un texto, donde el tamaño es mayor para las palabras buscadas o aparecidas con mas frecuencia.
- **Procesamiento de Lenguaje Natural (NLP)**: es el campo que combina las tecnologías de la ciencia computacional (como la inteligencia artificial, el aprendizaje automático o la inferencia estadística) con la lingüística aplicada, con el objetivo de hacer posible la comprensión y el procesamiento asistidos por ordenador de información expresada en lenguaje humano para determinadas tareas, como la traducción automática, los sistemas de diálogo interactivos, el análisis de opiniones, etc.
- **Clasificador Bayesiano:** El algoritmo de clasificación Naive Bayes es un clasificador probabilístico. Se basa en modelos de probabilidades que incorporan fuertes suposiciones de independencia. Las suposiciones de independencia a menos no tienen ningún efecto sobre la realidad. Por lo tanto, se consideran ingenuas (naive en inglés).

5. Metodología de la Investigación

Como se mencionó con anterioridad, el problema radica en la clasificación de tweets encontrados de acuerdo a un análisis de sentimiento, obteniendo como resultado tres agrupaciones de tweets, positivos, neutrales, y negativos, asimismo, se realizará una clasifición de acuerdo a la emoción que predomine en dicho tweet. Los pasos a seguir en nuestra investigación y a través de los cuales se obtendrá este resultado, se muestran a continuación:

- **Conexión con Twitter:** Lo primero que se busca es lograr establecer una conexión entre la aplicación de Twitter y el ambiente de RStudio a través de librerías, realizando el proceso de autenticación solicitado por la red social.
- **Búsqueda paramétrica:** Una vez lograda la conexión entre R y Twitter, será necesario realizar una búsqueda paramétrica, en donde los parámetros a recibir serán las palabras o temas de interés, así como una cantidad determinada de tweets. Sobre éstas palabras se buscarán tweets en la red social.
- **Extracción de tweets:** De acuerdo a los parámetros especificados en el paso anterior, mediante las librerías pertinentes se hará uso de una función para extraer los tweets solicitados desde la red social y se almacenarán en una variable de especificada en R.
- **Depuración de tweets:** Una vez extraídos los tweets, será necesario limpiarlos. En este proceso se busca eliminar todo tipo de palabras que no tenga importancia, conocidas como "stop words". Asimismo, se eliminarán palabras que no sean de nuestro interés en la búsqueda, o bien, que sean reduntantes, tales como la misma palabra buscada, o palabras relacionadas de manera obvia a ella.
- **Diccionario de palabras:** Previo al análisis y valoración de los tweets extraídos, será necesario conocer una lista o diccionario de palabras sobre las cuales será evaluado. Estas palabras podrán ser extraídas a través de alguna librería para R, de algún repositorio de internet, o bien, formulada por parte del usuario.
- **Valoración de tweets:** A través de una lectura del tweet, se hará un mapeado entre las palabras leídas en la publicación y aquellas que coincidan de los diccionarios. Una vez terminada la lectura de la publicación, se obtendrá una valoración general y estandarizada del mismo.
- **Determinar subjetividad:** Al obtener su valoración, se buscará conocer el grado de subjetividad de tweet, este grado estará dado por una categoría, a la cual pertenecerá el tweet. Dicha categoría tiene como opciones los tweets muy positivos, positivos, neutrales, negativos, y muy negativos.
- **Clasificación de tweets:** Una vez realizdo lo anterior, los grupos de tweets se verán reducidos a tres categorías, en las cuales el tweet podrá ser ubicado. Estas categorías serán las siguientes: "Tweets Positivos", "Tweets Neutros", "Tweets Negativos". Cada categoría contendrá un rango de valores estimado a través de los cuales la máquina podrá predecir a cuál pertenecerá el tweet. De la misma manera que en la valoración de tweets, los intervalos de las categorías estarán en función de un valor determinado por la puntuación del tweet.
- **Graficación de resultados:** Al haber conseguido clasificar los tweets, se buscará representar los resultados de la clasificación a través de gráficos que permitan analizar la relación de los dispositivos con las salidas.
- **Presentación de palabras representativas de cada clasificación:** Al haber clasificado correctamente los tweets, se buscará que la máquina extraiga las palabras más frecuentes en la búsqueda, así como las más significativas dentro de cada categoría de emociones, con la finalidad de que el usuario conozca la relación entre dichas palabras y el sentimiento u opiniónes de los usuarios de la red social.

6. Descripción de la Implementación

6.1. Creación de Aplicación y Autenticación

El primer paso antes de la codificación, es la creación de una aplicación en Twitter. Se crea una cuenta de usuario de Twitter (en caso de no tener una) y se accede a la página web *www.app.twitter.com*, se iniciará sesión con la cuenta creada, y se creará una *app*. Una vez creada la app de Twitter, se generarán las siguientes claves, *api key*, *api secet*, *access*

token, y access token secret, las cuales serán utilizadas para la autorización. La autenticación requerirá que el usuario ingrese en consola una clave dinámica generada por twitter.

6.2. Extracción de Tweets

Una vez autenticada la aplicación, el siguiente paso es extraer los tweets utilizando la función *searchTwitter()*. Esta función lleva como argumentos la palabra a buscar, el número de tweets a recabar, el lenguaje, el tipo de tweet a recabar (popular, reciente, etc.), un rango de fechas, entre otros. No obstante, puesto que nuestro objetivo es el de recabar información sobre el Iphone X y el Galaxy Note 8 para compararlos, bastará con hacer una búsqueda para cada uno de esos productos, desde mediados de 2017.

6.3. Limpieza de Tweets

Al extraer los tweets, éstos traen consigo demasiados caracteres (como los emoticones,los *hashtags*, enlaces, etc), así como palabras irrelevantes a nuestra búsqueda. Es por eso que mediante una función creada llamada *Clean()*, se depurará el texto de cada uno de los tweets sobre ambos dispositivos, eliminando cualquier palabra y caracter innecesario.

6.4. Adjuntar los Conjuntos de Datos de Palabras

Al mismo tiempo, teniendo listos los tweets (únicamente con palabras significativas), es posible comenzar con el proceso de puntuación. El primer paso para lograrlo es introduciendo un listado de palabras buenas y malas, sobre las cuales será puntuado el tweet. Las listas utilizadas en este proyecto se encuentran referenciadas en la sección de bibliografías.

6.5. Creación de las Funciones de Polaridad y Emociones

Al lograr vincular los listados de palabras con el programa, es posible comenzar con la clasificación de tweets. Sin embargo, para llegar a este paso es necesario crear funciones que se encarguen de asignar una puntuación a cada uno de los tweets encontrados.

6.6. Clasificación de Tweets

Una vez creadas las funciones de polaridad y de emociones, el siguiente paso es clasificar los tweets. Esta clasificación, a su vez, se logrará mediante otra función que logre vincular las puntuaciones de cada tweet con la polaridad o emoción uw mejor se ajuste.

6.7. Graficación de Resultados

Al haber clasificado los tweets, el último paso consta de mostrar gráficamente los resultados, de manera que éstos sean fáciles de entender e interpretar por las personas. Por lo tanto, para la elaboración de esta tarea, se utilizarán gráficos simples, tales como las gráficas de barras, las gráficas de pastel, entre otros.

6.8. Creación de Nube de Palabras

Una vez que los tweets han sido depurados y que solo contienen palabras significativas, es posible utilizarlos para realizar una nube de palabras con la ayuda de la libreria *wordcloud*. Esta librería permite crear una nube de palabras que contiene las palabras con mayor frecuencia de aparición en los tweets ya clasificados, sobre cada dispositivo, dando una pauta para conocer qué es lo que opinan los usuarios de Twitter acerca de ellos.

7. Conclusión

De acuerdo a lo mencionado a lo largo de este documento, se puede inferir que el Análisis de Sentimiento y la Minería de Datos son herramientas muy poderosas actualmente. A través de la implementación de ellas en el lenguaje de programación R, el equipo ha sido capaz de analizar las opiniones de los usuarios de twitter sobre los distintos dispositivos propuestos (Apple iPhone X, Samsung Galaxy Note 8 y Google Pixel 2). Esto ha servido para darnos una idea de la perspectiva de las personas sobre cada uno de estos terminales, así como para proponer soluciones que consigan erradicar cualquier sentimiento negativo o de rechazo hacia un dispositivo en específico. Cabe destacar la importancia y la extensa área de aplicación que tienen estas herramientas, pues pueden ser utilizadas en casi cualquier rubro, siempre y cuando exista un tráfico de datos considerable para realizar el análisis. Estas herramientas, además, permiten llegar a soluciones de una manera efectiva y rápida.

Es importante considerar que el curso de aprendizaje de máquina fue totalmente beneficioso para el desarrollo de este proyecto, pues gracias al contenido de la materia le fue posible al equipo comprender la estructura del problema, así como la metodología a seguir para lograr el objetivo, que era el análisis de sentimiento en los tweets relacionados al iPhone X, Note 8 y Pixel 2. Se espera que se le de un seguimiento a estas herramientas y lograr, a través de ellas, formular soluciones a problemas de gran importancia para nuestra generación, así como aprovechar al máximo la gran cantidad de datos que tenemos actualmente.De esta manera, no solo se aplicarán los conocimientos aprendidos en el curso de aprendizaje de máquina, sino que también se llevarán a cabo en proyectos aplicados en situaciones reales y con datos reales.

8. Resultados de la implementación

A continuación, se muestran los resultados obtenidos tras la implementación del proyecto.

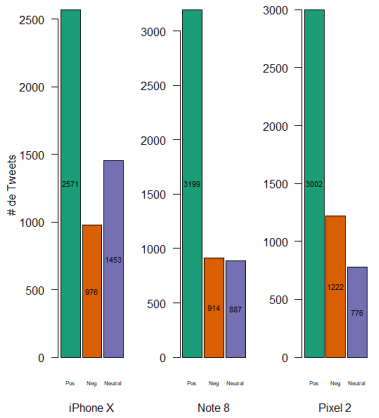

Figura 1. Tweets categorizados por subjetividad en cada dispositivo.

De acuerdo a la clasificación de tweets por subjetividad en cada dispositivo, se obtuvo la Figura 1. Por los números indicados en las gráficas, se puede observar cómo es que de un total de 5000 tweets recabados por cada dispositivo, el iPhone X resultó ser el que más tweets neutros obtuvo, sin embargo, fue el que consiguió la menor cantidad de tweets positivos. Por otro lado, el Note 8 obtuvo la menor cantidad de tweets negativos, y fue el dispositivo con mayor número de tweets positivos. Lo que realmente sorprende es que el Pixel 2 haya conseguido superar al iPhone X en cuanto a tweets positivos, pues fue el que mayor consiguió, aunque también consiguió la mayor cantidad de tweets negativos y la menor cantidad de tweets neutros.

Comentado lo anterior, se puede inferir que, gracias a que las marcas de Apple y Samsung son muy reconocidas por sus smartphones, las críticas hacia ellos han sido un poco más regulares, inclinándose hacia el lado positivo, caso contrario con el Pixel 2, que acaba de introducirse a esta rama y cuyos tweets negativos representan más de la tercera parte de los positivos, siendo muy irregular.

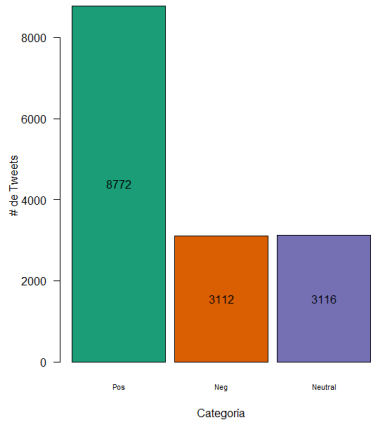

Figura 2. Tweets categorizados por subjetividad de todos los dispositivos.

Con respecto a los datos recabados por la Figura 1, es posible construir un gráfico que represente de manera general la totalidad de tweets recibidos por categoría de subjetividad. Como era de esperarse, los tweets positivos resultaron ser la mayor parte de los totales recabados, sin embargo, los tweets negativos se encuentran a la par con los tweets neutrales. Esta elevada cantidad de tweets negativos puede deberse a diferentes factores, tales como el costo del iPhone X, Note 8, y Pixel 2, las explosiones de las baterías en smartphones de Samsung, o la inexperiencia de Google en la industria de la telefonía con el Pixel 2.

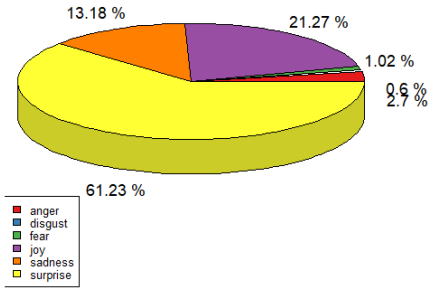

Figura 3. Distribución de las emociones en el iPhone X.

En la Figura 3, se busca analizar la distribución de los tweets en las diferentes categorías de emociones. Se puede inferir, por obviedad, que la gran mayoría de los tweets recaen en la emoción de *sorpresa*, pues el el iPhone X llegó para revolucionar en Apple con todas sus novedades, además, se trataba de nada más y nada menos que del celular con el que la compañía festejaría el décimo aniversario del iPhone. Cabe destacar que existe una parte sumamente considerable que se categoriza triste con el iPhone X. Esto puede deberse a su elevado costo de lanzamiento, a su poca disponibilidad, o bien, a su diseño.

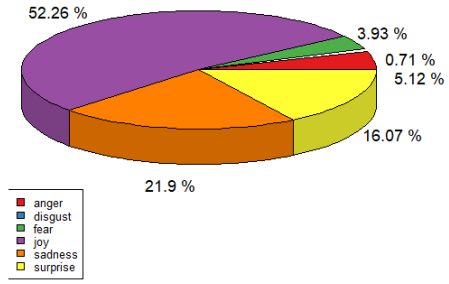

Figura 4. Distribución de las emociones en el Note 8.

En cuanto al Galaxy Note 8, la gran mayoría de los tweets se encuentran concentrados en la emoción de *alegría*, mientras que casi la cuarta parte del total recabado se categorizó como *triste*. Los principales puntos a considerar para la tristeza de los usuarios puede ser el alto costo impuesto por Samsung al dispositivo, la fragilidad de la pantalla, o bien, por el diseño presentado. Por otro lado, la alegría de los usuarios podría radicar en la gran potencia y funcionalidad que permite el dispositivo, así como en el gran tamaño de la pantalla presentado en este modelo.

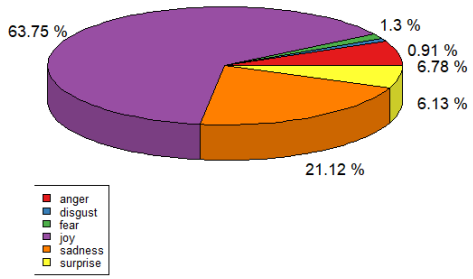

Figura 5. Distribución de las emociones en el Pixel 2.

Por último, el Google Pixel 2 muestra resultados alentadores, pues, al ser un dispositivo relativamente nuevo en este rubro, más de la mitad de los tweets recabados se muestran alegres con él. Esto puede significar que el dispositivo va por buen camino, no obstante, hay detalles por mejorar, pues se encuentra que el 21 % de los tweets se muestran tristes. Las posibles respuestas a esta gran cantidad de tweets podrían ser el alto precio con el que el Pixel 2 se lanzó al mercado, o bien, el diseño parecido al de otros dispositivos y la falta de innovación.

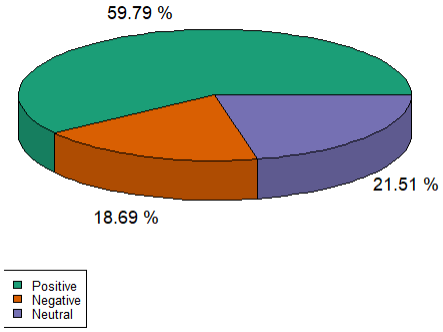

Figura 6. Distribución de las emociones en todos los dispositivos.

En la Figura 6, se muestra la cantidad total de tweets positivos, negativos y neutros conseguidos entre los tres dispositivos. De aquí se pueden concluir varias cosas, por ejemplo, que los tres dispositivos, en general, han logrado satisfacer las necesidades de los clientes, pues casi el 60 % de los tweets fueron positivos. Asimismo, aunque se recabaron casi 19 % de tweets negativos sobre los totales, no es una cantidad tan alarmante, considerando que casi el 22 % de los tweets conseguidos corresponden a una subjetividad neutral.

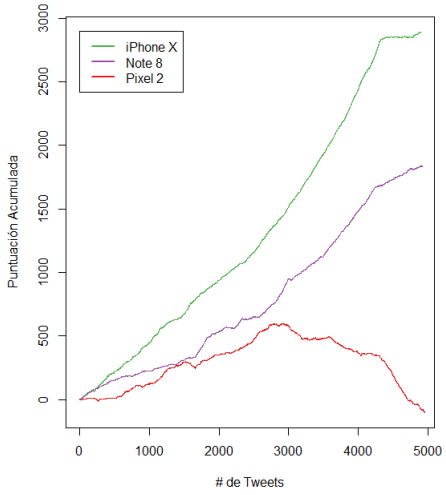

Figura 7. Puntuación histórica acumulada por cada dispositivo.

La Figura 7, a comparación de las analizadas anteriormente, busca explicar la trayectoria de la puntuación obtenida por los dispositivos con la llegada de cada tweet. Tal y como se puede observar, el iPhone X prácticamente no tuvo ninguna alteración, su trayectoria fue casi completamente ascendente. El Galaxy Note 8, tendió a subir conforme llegaban los tweets, aunque se alcanzan a apreciar algunas caídas por tweets con puntuaciones negativas. En cuanto al Pixel 2, el caso es completamente distinto. Desde los inicios la línea va mostrando inconsistencias, llega a su punto máximo al haber recorrido aproximadamente 2900 tweets, pero a partir de ahí, su trayectoria va en declive. Esto es un aspecto interesante considerando que fue el segundo dispositivo con más tweets positivos, aunque, claramente esto no representa qué puntuaciones positivas obtuvo, pues como se puede observar en la Figura 7, las puntuaciones de los tweets negativos fueron mayores a las obtenidas por los positivos, aun tratándose de menos tweets.

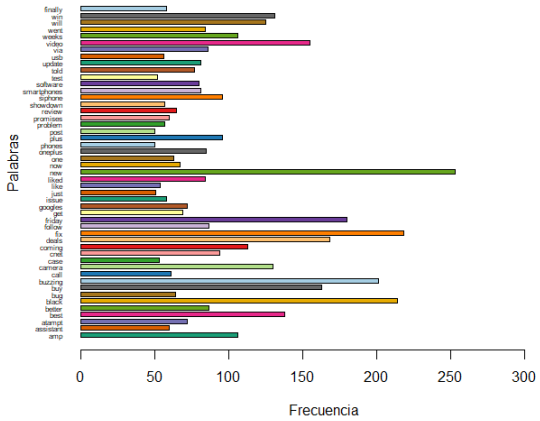

Figura 8. Lista de palabras encontradas con mayor frecuencia en los tweets (con una aparición mínima de 50 veces).

En la Figura 8 se encuentran las palabras con mayor

frecuencia de aparición en los tweets obtenidos. Es importante destacar de este grupo las palabras más repetidas, tales como *new*, *fix*, *buzzing* o *video*. Estas palabras nos permiten inferir los intereses o sensaciones de los consumidores con respecto al dispositivo. Se podría considerar que los usuarios se emocionaron en gran medida cuando alguno de estos dispositivos salió a la venta. Además, una de las características principales de estos tres smartphones en particular, es la gran actualización que recibieron para su cámara, por lo que no es una coincidencia que *video* y *camera* se encuentren en este grupo.

Figura 9. Nube de palabras categorizada por emociones del iPhone X.

Tomando en consideración las palabras con mayor frecuencia de aparición en los tweets, es posible encontrar las más mencionadas con respecto a un dispositivo en específico. En este caso, la Figura 9 muestra las palabras con mayor frecuencia categorizadas por la emoción a la que pertenecen para el iPhone X. Tomando las más populares de cada grupo es posible darse una idea general de lo que se piensa del dispositivo. Por ejemplo, podría decirse que la mayoría está triste por la dificultad de obtener el dispositivo, o que se encuentra impresionado por lo asombroso que es el diseño, tal vez a los usuarios no les pareció suficiente la reducción de los marcos y no les gustan los que tiene este modelo, entre otras cosas.

Figura 10. Nube de palabras categorizada por emociones del Note 8.

En cuanto al Note 8, se puede observar que las palabras que representan disgusto son mucho menores que las contenidas por el iPhone X, sin embargo, sorprende bastante que fueron repetidas con mucha frecuencia. Esto permite que, a simple vista, se pueda conocer que a los usuarios no les gustó la duración de la batería o el consumo de la misma, así como lo propenso que es a que se dañe. Sin embargo, se pueden resaltar las palabras de sorpresa, pues alagan al dispositivo, dando a entender que es un smartphone de gama alta y que es de los mejores en el mercado.

Figura 11. Nube de palabras categorizada por emociones del Pixel 2.

Por último, el Google Pixel 2 sorprende nuevamente, ahora con la frecuencia de aparición de palabras que denotan sorpresa. Estas palabras describen al dispositivo como un celular increíble con un software y diseño de calidad. No obstante, consideran como enojo la calidad de sonido presentada por el dispositivo y temen de la calidad final que pueda presentar, así como la inseguridad que sienten sobre adquirirlo.

Figura 12. Nube de palabras categorizada por emociones de todos los dispositivos.

En la Figura 12 se concentran las palabras más frecuentemente encontradas en todos los tweets en general, sin considerar categorizaciones por dispositivo. Esto permite inferir cosas interesantes, pues es el punto de intersección entre los tres smartphones analizados. en este caso, se podría pensar que los usuarios se encuentran sorprendidos con

respecto a los diseños presentados, así como a la autonomía de la batería y al tamaño de la pantalla, sin embargo, puede entenderse por *buzzing* a la gran espera que han tenido por los dispositivos, sintiendo ese miedo de que no sea de su agrado, o bien, que se tenga el temor de fallos de audio relacionados con zumbidos.

9. Requerimientos

Para la correcta implementación del proyecto en el lenguaje de programación R, se necesitarán las siguientes librerías y archivos listados a continuación:

- **ROAuth:** Librería utilizada para realizar una autenticación con la aplicación de Twitter.
- **twitteR:** Librería requerida para extraer los tweets.
- **base64enc:** Librería para la codificación en base64.
- **httr:** Librería necesaria para realizar una conexión vía URL y HTTPS.
- **devtools:** Librería que permite instalar librerías de R desde un repositorio de GitHub, entre otras funciones.
- **tm:** Librería enfocada en la implementación de *text mining*, requerida para realizar la limpieza de los tweets.
- **wordcloud:** Librería que permite crear una nube de palabras con mayor frecuencia de aparición en un conjunto de datos.
- **RColorBrewer:** Librería que permite utilizar paletas de colores en la graficación de nubes de palabras.
- **RTextTools:** Librería utilizada como base para la función creada *createMatrix()* (*create_matrix* de *RTextTools*).
- **Opinion/sentiment Lexicon:** Obtenido de *https://www.cs.uic.edu/Ĩiub/FBS/sentiment-analysis.html#lexicon*. (Link de descarga)
- **Funciones bayesianas:** Funciones implementadas en el lenguaje de programación R, necesarias para realizar la clasificación de tweets de acuerdo a la emoción correspondiente, así como su clasificación de acuerdo a la polaridad.
- **Relación de palabras con emociones y con subjetividad:** Base de datos que contiene palabras relacionadas con una emoción y subjetividad específica, requerida por las funciones bayesianas descritas anteriormente. Últimos dos requerimientos obtenidos de: *https://cran.r-project.org/src/contrib/Archive/sentiment/*. (Link de descarga)

10. Código de Implementación en R

Para fines prácticos y de estética del documento, se han omitido los comentarios del código. De cualquier manera, en el repositorio del proyecto (listado en la sección de Enlace del Video de Resultados) se encuentra el código de manera íntegra.

```r
#-----------------------------------INSTALAR LIBRERIAS-----------------------------------#
# install.packages("ROAuth")
# install.packages("twitteR")
# install.packages("base64enc")
# install.packages("httr")
# install.packages("devtools")
# install.packages("tm")
# install.packages("wordcloud")
# install.packages("RColorBrewer")
# install.packages("RTextTools")

#-----------------------------------CARGAR LIBRERIAS-----------------------------------#
library(ROAuth)
library(twitteR)
library(base64enc)
library(httr)
library(devtools)
library(tm)
library(wordcloud)
library(RColorBrewer)
library(RTextTools)

#-----------------------------------AUTENTICACION DE TWITTER-----------------------------------#
api_key = "Y5fbQA5lJodmk0E4q4c1DYbXD"
api_secret = "f4OQTcDmCg56EiHEaSZ7Zo3POHrEn2T0QHj0KbqBmZyRWmNkVL"
access_token = "919998032938012672-6GP05oGCs1SZ6cwP9QwXw8VWulKDXDe"
access_token_secret = "YWOmWI4B3UTzOeYMVyipwVtiQvzIqygfO2cN2ySd1RZk4"
request_url = 'https://api.twitter.com/oauth/request_token'
access_url = 'https://api.twitter.com/oauth/access_token'
auth_url = 'https://api.twitter.com/oauth/authorize'

setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)

credential = OAuthFactory$new(consumerKey = api_key,consumerSecret =
        api_secret,requestURL = request_url,accessURL = access_url,authURL =
        auth_url)

credential$handshake(cainfo = system.file("CurlSSL","cacert.pem",package ="
        RCurl"))

#-----------------------------------EXTRACCION DE TWEETS-----------------------------------#
iphoneTweets = searchTwitter("iPhone X",n=5000,lang="en",since = "2017-06-01"
        )
noteTweets = searchTwitter("Galaxy Note 8",n=5000,lang="en",since = "
        2017-06-01")
pixelTweets = searchTwitter("Google+Pixel 2",n=5000,lang="en",since = "
        2017-06-01")

iphoneTweets.df = twListToDF(iphoneTweets)
noteTweets.df = twListToDF(noteTweets)

#-----------------------------------LIMPIEZA DE TWEETS-----------------------------------#
tweets.clean = function(tweetsList){
  cleanedTweets = sapply(tweetsList,function(row) iconv(row, "latin1", "ASCII
        ", sub=""))
  cleanedTweets = gsub("(f|ht)tp(s?)://(.*)[.][a-z]+", "", cleanedTweets)
  tweetsCorpus = Corpus(VectorSource(cleanedTweets[names(cleanedTweets)]))
  tweetsCorpus = tm_map(tweetsCorpus, function(tweet) gsub("http[^[:space:]]*
        ", "", tweet))
  tweetsCorpus = tm_map(tweetsCorpus, function(tweet) gsub("\xed[^[:space:]]*
        ", "", tweet))
  tweetsCorpus = tm_map(tweetsCorpus, function(tweet) gsub("/[^[:space:]]*",
        "", tweet))
  tweetsCorpus = tm_map(tweetsCorpus, function(tweet) gsub("@[^[:space:]]*",
        "", tweet))
  tweetsCorpus = tm_map(tweetsCorpus, removePunctuation)
  tweetsCorpus = tm_map(tweetsCorpus, content_transformer(tolower))
  tweetsCorpus = tm_map(tweetsCorpus, removeWords, c(stopwords("english"),"\n
        ","rt"))
  tweetsCorpus = tm_map(tweetsCorpus, removeNumbers)
  searchedWords = c("iphone x","iphonex","iphone","apple","ios","galaxy","
        galaxynote","note","note 8","samsung","android","google","pixel","
        pixel 2","pixel2","phone","smartphone","cellphone","giveaway","
        international","tweet")

  tweetsCorpus = tm_map(tweetsCorpus, removeWords, searchedWords)
  tweetsCorpus = tm_map(tweetsCorpus, stripWhitespace)
  cleanedTweets = data.frame(text = sapply(tweetsCorpus, as.character),
        stringsAsFactors = FALSE)
  return(cleanedTweets)
}
iphoneCleanedTweets = tweets.clean(iphoneTweets.text)
noteCleanedTweets = tweets.clean(noteTweets.text)
pixelCleanedTweets = tweets.clean(pixelTweets.text)

#-----------------------------------ADJUNTAR CONJUNTO DE DATOS DE PALABRAS-----------------------------------#
posWords = scan('./posWords.txt', what='character', comment.char = ';')
negWords = scan('./negWords.txt', what='character', comment.char = ';')

#-----------FUNCIONES DE PUNTUACION POR PALABRAS POSITIVAS Y NEGATIVAS-----------#
getScores = function(tweets, pos.words, neg.words){
  results.df = data.frame(matrix(nrow=0,ncol=5))

  for(i in 1:length(tweets)){
    tweet = tweets[i]
    words = strsplit(tweet,' ')
    words = unlist(words)
    words = words[!words %in% c("","  ","NA")]
    pos.match  = match(words, posWords)
    pos.match = !is.na(pos.match)
    neg.match = match(words,negWords)
    neg.match = !is.na(neg.match)
    totalPos = sum(pos.match)
    totalNeg = sum(neg.match)
    score = totalPos - totalNeg

    if(score <= -3){
      category = "very negative"
    }
    else if(score > -3 && score < 0){
      category = "negative"
    }
    else if(score == 0){
      category = "neutral"
    }
    else if(score > 0 && score <= 2){
      category = "positive"
    }
    else{
      category = "very positive"
    }
    if(length(words) != 0L){
      results.df[i,]= c(tweet,totalPos,totalNeg,score,category)
    }
  }
  colnames(results.df) = c("tweet","pos","neg","score","category")
  return(results.df)
}
```

```
iphone.scores = getScores(iphoneCleanedTweets$text, posWords, negWords)
note.scores = getScores(noteCleanedTweets$text, posWords, negWords)
pixel.scores = getScores(pixelCleanedTweets$text, posWords, negWords)

#----------FUNCIONES DE POLARIDAD Y EMOCIONES----------#
createMatrix = function(textColumns,language="english", minDocFreq=1,
    minWordLength=3,weighting=weightTf){
  control = list(language=language,minDocFreq=minDocFreq,minWordLength=
    minWordLength,weighting=weighting)
  trainingColumn = apply(as.matrix(textColumns),1,paste,collapse=" ")
  trainingColumn = sapply(as.vector(trainingColumn,mode="character"),iconv,to
    ="UTF8",sub="byte")
  corpus = Corpus(VectorSource(trainingColumn),readerControl=list(language=
    language))
  matrix = DocumentTermMatrix(corpus,control=control)
  return(matrix)
}

bayesianEmotions = function(tweets,prior=1.0){
  matrix = createMatrix(tweets)
  lexicon = read.csv(file = "emotions.csv",header=FALSE,sep=",")
  counts = list(anger=length(which(lexicon[,2]=="anger")),disgust=length(
    which(lexicon[,2]=="disgust")),fear=length(which(lexicon[,2]=="fear"))
    ,joy=length(which(lexicon[,2]=="joy")),sadness=length(which(lexicon
    [,2]=="sadness")),surprise=length(which(lexicon[,2]=="surprise")),
    total=nrow(lexicon))
  documents = c()

  for(i in 1:nrow(matrix)){
    scores = list(anger=0,disgust=0,fear=0,joy=0,sadness=0,surprise=0)
    doc = matrix[i,]
    words = findFreqTerms(doc,lowfreq=1)
    for(word in words){
      for(key in names(scores)){
        emotions = lexicon[which(lexicon[,2]==key),]
        index = pmatch(word,emotions[,1],nomatch=0)
        if(index>0){
          entry = emotions[index,]
          category = as.character(entry[[2]])
          count = counts[[category]]
          score = 1.0
          score = abs(log(score*prior/count))
          scores[[category]] = scores[[category]]+score
        }
      }
    }

    for(key in names(scores)){
      count = counts[[key]]
      total = counts[["total"]]
      score = abs(log(count/total))
      scores[[key]]  = scores[[key]]+score
    }

    best_fit = names(scores)[which.max(unlist(scores))]
    if(best_fit == "disgust" && as.numeric(unlist(scores[2]))-3.09234 < .01){
      best_fit = NA
    }

    documents = rbind(documents,c(scores$anger,scores$disgust,scores$fear,
      scores$joy,scores$sadness,scores$surprise,best_fit))
  }
  colnames(documents) = c("Anger","Disgust","Fear","Joy","Sadness","Surprise"
    ,"Emotion")
  return(documents)
}

bayesianPolarity = function(tweets,pStrong=0.5,pWeak=1.0,prior=1.0){
  matrix = createMatrix(tweets)
  lexicon = read.csv(file = "subjectivity.csv",header=FALSE,sep=",")
  counts = list(positive=length(which(lexicon[,3]=="positive")),negative=
    length(which(lexicon[,3]=="negative")),total=nrow(lexicon))
  documents = c()

  for(i in 1:nrow(matrix)){
    scores = list(positive=0,negative=0)
    doc = matrix[i,]
    words = findFreqTerms(doc,lowfreq=1)
    for(word in words){
      index = pmatch(word,lexicon[,1],nomatch=0)
      if(index>0){
        entry = lexicon[index,]
        polarity = as.character(entry[[2]])
        category = as.character(entry[[3]])
        count = counts[[category]]
        score = pWeak
        score = abs(log(score*prior/count))
        scores[[category]] = scores[[category]]+score
      }
    }

    for(key in names(scores)){
      count = counts[[key]]
      total = counts[["total"]]
      score = abs(log(count/total))
      scores[[key]] = scores[[key]]+score
    }

    best_fit = names(scores)[which.max(unlist(scores))]
    ratio = as.integer(abs(scores$positive/scores$negative))
    if(!length(ratio)==0){
      if(ratio==1){
        best_fit="neutral"
      }
    }

    documents = rbind(documents,c(scores$positive,scores$negative,abs(
      scores$positive/scores$negative),best_fit))
  }
  colnames(documents) = c("Pos","Neg","Pos/Neg","Polarity")
  return(documents)
}
emotions = c("anger","disgust","fear","joy","sadness","surprise")
iphoneEmotions.class = bayesianEmotions(iphone.scores$tweet)
noteEmotions.class = bayesianEmotions(note.scores$tweet)
pixelEmotions.class = bayesianEmotions(pixel.scores$tweet)

iphoneEmotions = iphoneEmotions.class[,7]
noteEmotions = noteEmotions.class[,7]
pixelEmotions = pixelEmotions.class[,7]

iphoneEmotions[is.na(iphoneEmotions)] = 'unknown'
noteEmotions[is.na(noteEmotions)] = 'unknown'
pixelEmotions[is.na(pixelEmotions)] = 'unknown'

iphonePolarity.class = bayesianPolarity(iphone.scores$tweet)
notePolarity.class = bayesianPolarity(note.scores$tweet)
pixelPolarity.class = bayesianPolarity(pixel.scores$tweet)

iphonePolarity = iphonePolarity.class[,4]
notePolarity = notePolarity.class[,4]
pixelPolarity = pixelPolarity.class[,4]

iphonePolarity = data.frame(text=iphone.scores$tweet, emotion=iphoneEmotions,
    polarity=iphonePolarity, stringAsFactors=FALSE)
notePolarity = data.frame(text=note.scores$tweet, emotion=noteEmotions,
    polarity=notePolarity, stringAsFactors=FALSE)
pixelPolarity = data.frame(text=pixel.scores$tweet, emotion=pixelEmotions,
    polarity=pixelPolarity, stringAsFactors=FALSE)

#----------GRAFICACION DE RESULTADOS----------#

    #----------Subjetividad por dispositivo----------#
iphone.sub = data.frame(Pos = length(which(iphonePolarity.class=="positive"))
    , Neg = length(which(iphonePolarity.class=="negative")), Neutral =
    length(which(iphonePolarity.class=="neutral")))
note.sub = data.frame(Pos = length(which(notePolarity.class=="positive")),
    Neg = length(which(notePolarity.class=="negative")), Neutral = length(
    which(notePolarity.class=="neutral")))
pixel.sub = data.frame(Pos = length(which(pixelPolarity.class=="positive")),
    Neg = length(which(pixelPolarity.class=="negative")), Neutral = length
    (which(pixelPolarity.class=="neutral")))

par(fig = c(0.05, 0.35, 0, 0.95), mar = c(5,4,3,0))
barplot(as.numeric(c(iphone.sub$Pos,iphone.sub$Neg,iphone.sub$Neutral)),
    names.arg = names(iphone.sub), space=c(0.1,1), cex.names = 0.5, xlab="
    iPhone X", ylab="# de Tweets"las = 1, col = brewer.pal(3,'Dark2'))
text(c(1.5,2.6,3.7),c(iphone.sub$Pos/2,iphone.sub$Neg/2,iphone.sub$Neutral/2)
    ,c(iphone.sub$Pos,iphone.sub$Neg,iphone.sub$Neutral),cex = 0.7)

par(fig = c(0.35, 0.65, 0, 0.95), mar = c(5,4,3,0), new = TRUE)
barplot(as.numeric(c(note.sub$Pos,note.sub$Neg,note.sub$Neutral)), names.arg
    = names(note.sub), space=c(0.1,0), cex.names = 0.5, xlab="Note 8", las
    = 1, col = brewer.pal(3,'Dark2'))
text(c(0.5,1.6,2.7),c(note.sub$Pos/2,note.sub$Neg/2,note.sub$Neutral/2),c(
    note.sub$Pos,note.sub$Neg,note.sub$Neutral),cex = 0.7)

par(fig = c(0.65, 0.95, 0, 0.95), mar = c(5,4,3,0), new = TRUE)
barplot(as.numeric(c(pixel.sub$Pos,pixel.sub$Neg,pixel.sub$Neutral)), names.
    arg = names(pixel.sub), space=c(0.1,0), cex.names = 0.5, xlab="Pixel 2
    ", las = 1, col = brewer.pal(3,'Dark2'))
text(c(0.5,1.6,2.7),c(pixel.sub$Pos/2,pixel.sub$Neg/2,pixel.sub$Neutral/2),c(
    pixel.sub$Pos,pixel.sub$Neg,pixel.sub$Neutral),cex = 0.7)
title(main = 'Tweets Categorizados por Subjetividad por Dispositivo', outer =
    TRUE, line = -2)

    #----------Subjetividad por todos los dispositivos----------#
mixed.sub = data.frame(Pos = length(which(iphonePolarity.class=="positive"))
    + length(which(notePolarity.class=="positive")) + length(which(
    pixelPolarity.class=="positive")),
                       Neg = length(which(iphonePolarity.class=="negative"))
    + length(which(notePolarity.class=="negative")) + length(which(
    pixelPolarity.class=="negative")),
                       Neutral = length(which(iphonePolarity.class=="neutral
    ")) + length(which(notePolarity.class=="neutral")) + length(which(
    pixelPolarity.class=="neutral")))

barplot(as.numeric(c(mixed.sub$Pos,mixed.sub$Neg,mixed.sub$Neutral)), names.
    arg = names(mixed.sub), space=c(0.1,1), cex.names = 0.7, xlab="
    Categoria", ylab="# de Tweets",las = 1, col = brewer.pal(3,'Dark2'))
text(c(1.5,2.6,3.7),c(mixed.sub$Pos/2,mixed.sub$Neg/2,mixed.sub$Neutral/2),c(
    mixed.sub$Pos,mixed.sub$Neg,mixed.sub$Neutral),cex = 1)
title(main = 'Tweets Categorizados por Subjetividad de Todos los Dispositivos
    ', outer = TRUE, line = -2)

    #----------Emociones por dispositivo----------#
iphoneAnger = sum(length(which(iphoneEmotions == "anger")))
iphoneDisgust = sum(length(which(iphoneEmotions == "disgust")))
iphoneFear = sum(length(which(iphoneEmotions == "fear")))
iphoneJoy =  sum(length(which(iphoneEmotions == "joy")))
iphoneSadness =  sum(length(which(iphoneEmotions == "sadness")))
iphoneSurprise =  sum(length(which(iphoneEmotions == "surprise")))
emotionsIphone = c(iphoneAnger,iphoneDisgust, iphoneFear, iphoneJoy,
    iphoneSadness, iphoneSurprise)
total = sum(iphoneAnger,iphoneDisgust,iphoneFear,iphoneJoy,iphoneSadness,
    iphoneSurprise)
percentages = c(paste(round(iphoneAnger/total*100,2),"%"),paste(round(
    iphoneDisgust/total*100,2),"%"),paste(round(iphoneFear/total*100,2),"%
    "),paste(round(iphoneJoy/total*100,2),"%"),paste(round(iphoneSadness/
    total*100,2),"%"),paste(round(iphoneSurprise/total*100,2),"%"))

pie3D(x=emotionsIphone,labels = percentages, col = brewer.pal(6,'Set1'),main=
    "Distribucion de Emociones en el iPhone X (%)", labelcex = 1.1)
legend("bottomleft", legend = emotions, fill = brewer.pal(6,'Set1'), cex =
    0.55)

noteAnger = sum(length(which(noteEmotions == "anger")))
noteDisgust = sum(length(which(noteEmotions == "disgust")))
noteFear = sum(length(which(noteEmotions == "fear")))
noteJoy =  sum(length(which(noteEmotions == "joy")))
noteSadness =  sum(length(which(noteEmotions == "sadness")))
noteSurprise =  sum(length(which(noteEmotions == "surprise")))
emotionsNote = c(noteAnger,noteDisgust, noteFear, noteJoy, noteSadness,
    noteSurprise)
total = sum(noteAnger,noteDisgust,noteFear,noteJoy,noteSadness,noteSurprise)
percentages = c(paste(round(noteAnger/total*100,2),"%"),paste(round(
    noteDisgust/total*100,2),"%"),paste(round(noteFear/total*100,2),"%"),
    paste(round(noteJoy/total*100,2),"%"),paste(round(noteSadness/total
```

```
        *100,2),"%"),paste(round(noteSurprise/total*100,2),"%"))

pie3D(emotionsNote, labels = percentages, col = brewer.pal(6,'Set1'),main = "
        Distribucion de Emociones en el Note 8 (%)", labelcex = 1.1)
legend("bottomleft", legend = emotions, fill = brewer.pal(6,'Set1'), cex =
        .55)

pixelAnger = sum(length(which(pixelEmotions == "anger")))
pixelDisgust = sum(length(which(pixelEmotions == "disgust")))
pixelFear = sum(length(which(pixelEmotions == "fear")))
pixelJoy =  sum(length(which(pixelEmotions == "joy")))
pixelSadness =  sum(length(which(pixelEmotions == "sadness")))
pixelSurprise =  sum(length(which(pixelEmotions == "surprise")))
emotionsPixel = c(pixelAnger,pixelDisgust, pixelFear, pixelJoy, pixelSadness,
         pixelSurprise)
total = sum(pixelAnger,pixelDisgust,pixelFear,pixelJoy,pixelSadness,
        pixelSurprise)
percentages = c(paste(round(pixelAnger/total*100,2),"%"),paste(round(
        pixelDisgust/total*100,2),"%"),paste(round(pixelFear/total*100,2),"%")
        ,paste(round(pixelJoy/total*100,2),"%"),paste(round(pixelSadness/total
        *100,2),"%"),paste(round(pixelSurprise/total*100,2),"%"))

pie3D(emotionsPixel, labels = percentages, col = brewer.pal(6,'Set1'), main =
         "Distribucion de Emociones en el Pixel 2 (%)", labelcex = 1.1)
legend("bottomleft", legend = emotions, fill = brewer.pal(6,'Set1'), cex =
        .5)

            #----------Emociones por todos los dispositivos----------#
mixPos = sum(length(which(iphonePolarity.class =="positive")),length(which(
        notePolarity.class =="positive")),length(which(notePolarity.class ==""
        positive")))
mixNeg = sum(length(which(iphonePolarity.class =="negative")),length(which(
        notePolarity.class =="negative")),length(which(notePolarity.class ==""
        negative")))
mixNeutral = sum(length(which(iphonePolarity.class =="neutral")),length(which
        (notePolarity.class =="neutral")),length(which(notePolarity.class ==""
        neutral")))
devices = c(mixPos,mixNeg,mixNeutral)
total = sum(devices)
percentages = c(paste(round(mixPos*100/total,2),"%"), paste(round(mixNeg*100/
        total,2),"%"),paste(round(mixNeutral*100/total,2),"%"))

pie3D(devices, labels = percentages, col = brewer.pal(3,'Dark2'), main = "
        Distribucion de Emociones en Todos los Dispositivos (%)", labelcex =
        1.1)
legend("bottomleft",legend = c("Positive","Negative","Neutral"), fill =
        brewer.pal(3,'Dark2'), cex = .8)

            #---Puntuaciones de tweets por todos los dispositivos---#
getAccumulated = function(list){
  accum = c()
  sum = 0
  for(i in 1:length(list)){
    if(!is.na(list[i])){
      sum = sum+as.numeric(list[i])
      accum = c(accum,sum)
    }
  }
  return(accum)
}

iphoneAccum = getAccumulated(iphone.scores$score)
noteAccum = getAccumulated(note.scores$score)
pixelAccum = getAccumulated(pixel.scores$score)

plot(iphoneAccum,type = "l", ylim = c(min(iphoneAccum,noteAccum,pixelAccum),
        max(iphoneAccum,noteAccum,pixelAccum)) ,col = brewer.pal(8,'Set1')[3],
         xlab = "# de Tweets", ylab = "Puntuacion Acumulada", main = "
        Puntuacion Historica Acumulada por Dispositivo")
lines(noteAccum, type = "l", col = brewer.pal(8,'Set1')[4])
lines(pixelAccum, type = "l", col = brewer.pal(8,'Set1')[1])
legend(0,1850,c("iPhone X","Note 8","Pixel 2"),lty=c(1), lwd=c(2.5),col = c(
        brewer.pal(8,'Set1')[3],brewer.pal(8,'Set1')[4],brewer.pal(8,'Set1')
        [1]))

            #----------Nube de palabras por emociones----------#
getTdm = function(tweets,tweetsEmotions){
  wcEmotions = levels(factor(emotions))
  nEmotions = length(wcEmotions)
  wcEmotions.docs = rep('', nEmotions)

  for(i in 1:nEmotions)
  {
    tmp = tweets$text[tweetsEmotions == wcEmotions[i]]
    wcEmotions.docs[i] = paste(tmp, collapse=' ')
  }

  corpus = Corpus(VectorSource(wcEmotions.docs))
  tdm = TermDocumentMatrix(corpus)
  tdm = as.matrix(tdm)
  colnames(tdm) = wcEmotions
  return (tdm)
}

iphoneCorpus = getTdm(iphoneCleanedTweets,iphoneEmotions)
noteCorpus = getTdm(noteCleanedTweets,noteEmotions)
pixelCorpus = getTdm(pixelCleanedTweets,pixelEmotions)

mixedTweets = rbind(iphoneCleanedTweets,noteCleanedTweets,pixelCleanedTweets)
mixedEmotions = rbind(iphoneEmotions,noteEmotions,pixelEmotions)
mixedCorpus = getTdm(mixedTweets,mixedEmotions)

comparison.cloud(iphoneCorpus, random.order = FALSE, max.words = 1000, rot.
        per=.15, colors = brewer.pal(emotions, 'Dark2'), scale = c(4,0.5),
        title.size = 1)
title(main = "Nube de Palabras de Emociones del iPhone X", outer = TRUE, line
         = -0.7)

comparison.cloud(noteCorpus, random.order = FALSE, max.words = 1000, rot.per
        =.15, colors = brewer.pal(emotions, 'Dark2'), scale = c(4,0.5), title.
        size = 1)
title(main = "Nube de Palabras de Emociones del Note 8", outer = TRUE, line =
         -0.7)

comparison.cloud(pixelCorpus, random.order = FALSE, max.words = 1000, rot.per
        =.15, colors = brewer.pal(emotions, 'Dark2'), scale = c(4,0.5), title.
        size = 1)
title(main = "Nube de Palabras de Emociones del Pixel 2", outer = TRUE, line
        = -0.7)

comparison.cloud(mixedCorpus, random.order = FALSE, max.words = 1000, rot.per
        =.15, colors = brewer.pal(emotions, 'Dark2'), scale = c(4,0.5), title.
        size = 1)
title(main = "Nube de Palabras de Emociones de Todos los Dispositivos", outer
         = TRUE, line = -0.7)

            #----------Frecuencia de palabras----------#
termsFreq = rowSums(as.matrix(mixedCorpus))
mostFreq = subset(termsFreq,termsFreq>=50)
df = data.frame(term = names(mostFreq), freq=mostFreq)

par(mai=c(1,1.5,1,1))
barplot(mostFreq, xlim=c(0,round(max(mostFreq)/100,0)*100), col = c(brewer.
        pal(name = 'Dark2',n = 8),brewer.pal(name = 'Paired',n = 12)), horiz =
         TRUE, las=1, cex.names = 0.5, space = c(0.5,0))
par(mar = c(5,7,4,2) + 0.1)
title(main = "Palabras Mas Frecuentes por Todos los Dispositivos", ylab = "
        Palabras", xlab = "Frecuencia")
```

11. Enlace del Video de Resultados

En el sigiente enlace se encuenta el video de demostración del código de implementación y resultados parciales del proyecto (video). Además, se incluye el enlace al repositorio del proyecto para mayor información. (repositorio).

Referencias

[1] ASP.NET Web API, Msdn.microsoft.com, 2017. [En línea]. Disponible en: https://msdn.microsoft.com/es-es/library/hh833994(v=vs.108).aspx.

[2] Business Intelligence, Sinnexus, 2016. [En línea]. Disponible en: http://www.sinnexus.com/business_intelligence/datamining.aspx

[3] Entendiendo el análisis de sentimiento: qué es y para qué se usa, brandwatch, 2015. [En línea]. Disponible en: https://www.brandwatch.com/es/2015/02/analisis-de-sentimiento/

[4] H. Fujita, M. Ali, A. Selamat, J. Sasaki and M. Kurematsu, Trends in applied knowledge-based systems and data science. Springer, 2016, pp. 269-279.

[5] IBM Knowledge Center, Ibm.com, 2017. [En linea]. Disponible en : https://www.ibm.com/support/knowledgecenter/es/SSEPGG_9.7.0/com.ibm.im.overview.doc/c_naive_bayes_classification.html.

[6] Liu, B., Hu, M. and Cheng, J. Opinion Observer: Analyzing and Comparing Opinions on the Web. Chiba, 2005 [En línea]. Disponible en: https://www.cs.uic.edu/ liub/publications/www05-p536.pdf

[7] Minería de Opiniones, BrainSINS, 2016.[En línea]. Disponible en: https://www.brainsins.com/es/blog/mineria-opiniones/3555/modeler_mainhelp_client_ddita/clementine/svm_howwork.html

[8] Procesamiento del lenguaje natural, Vicomtech.org, 2017. [En linea]. Disponible: http://www.vicomtech.org/t4/e11/procesamiento-del-lenguaje-natural.

[9] Y. Zhao, R and Data Mining: Examples and Case Studies, 1st ed. Elsevier, 2015.