

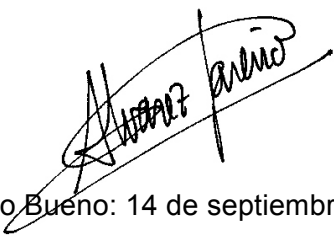


**Análisis de Sentimiento en Twitter de las  
principales Compañías del Sector Asegurador  
Español**

**Trabajo de Fin de Máster**

**Autor: Francisco José Martínez Martínez**

Tutor: Dr. José A. Álvarez Jareño

  
Visto Bueno: 14 de septiembre de 2017

# **Análisis de sentimiento en Twitter de las principales compañías del sector asegurador español**

Francisco José Martínez Martínez

## **Tutor:**

José Antonio Álvarez Jareño

## **Máster en Ciencias Actariales y Financieras**

Universidad de Valencia

## **Resumen:**

En este trabajo se realiza un análisis de sentimiento en Twitter de las principales compañías del sector asegurador español (PCSAE) mediante la utilización del software estadístico R y de léxicos de sentimiento. Obtener y comprender la gran cantidad de información gratuita que circula por Internet puede ser de gran interés para las empresas. Por ejemplo, en la red social Twitter circulan al día millones de opiniones sobre diversos temas, por lo que se decide utilizar esta red social para captar la opinión de los clientes sobre las PCSAE. En primer lugar se extraen los comentarios de Twitter, denominados tuits, para posteriormente prepararlos y limpiarlos para el análisis. Resulta clave la creación de un léxico específico para el sector asegurador (Lexiseg), cuya construcción se realiza partiendo de otros léxicos en castellano ya existentes y la agregación manual de palabras que mejoran los resultados de exactitud. Una vez disponemos de la base datos y del Lexiseg se procede al análisis de sentimiento, en el que podemos obtener diversos resultados para cada una de las PCSAE; polaridad, emociones, tópicos, etc. Finalmente se realiza un análisis y comparación de resultados entre las compañías y se abordan las principales conclusiones obtenidas en el trabajo.

## **Palabras clave:**

Análisis de sentimiento, sector asegurador, Twitter, polaridad, opinión, tuits, léxicos, minería de texto, emoción, tópicos.

## Índice:

|  |       |
|--|-------|
| 1.- Introducción.                                | 1-7   |
| 1.1.- Motivación.                                | 1-2   |
| 1.2.- Objetivos.                                 | 2-3   |
| 1.3.- Estado del arte.                           | 3-5   |
| 1.4.- Metodología.                               | 6-7   |
| 2.- Extracción de datos en Twitter.              | 8-11  |
| 3.- Procesamiento de datos.                      | 12-15 |
| 3.1.- Preparación de la base de datos.           | 12-13 |
| 3.2.- Limpieza de texto.                         | 13-15 |
| 4.- Léxico específico para el sector asegurador. | 16-23 |
| 4.1.- Léxicos en castellano.                     | 16-17 |
| 4.2.- Metodología de la construcción del léxico. | 18-19 |
| 4.3.- Creación del léxico.                       | 20-23 |
| 5.- Polaridad de sentimiento.                    | 24-30 |
| 5.1.- Determinación de la polaridad.             | 24-25 |
| 5.2.- Análisis de sentimiento.                   | 26-30 |
| 6.- Otra información relevante.                  | 31-34 |
| 6.1.- Análisis de emociones.                     | 31-32 |
| 6.2.- Análisis de frecuencias.                   | 32-34 |
| 7.- Análisis y comparación de resultados.        | 35-39 |
| 8.- Conclusiones.                                | 40-41 |
| 9.- Bibliografía.                                | 42-45 |
| Anexo I.   | 46-52 |
| Anexo II.  | 53-58 |

# 1.- Introducción.

## 1.1.- Motivación:

Hoy en día vivimos en un mundo totalmente informatizado y globalizado, en el que las personas comparten gran cantidad de información a través de Internet. En la web se generan diariamente millones de datos debido a la utilización masiva de las redes sociales y otros espacios online, como servicios de mensajería, blogs y wikis. En estos espacios los individuos interactúan entre sí, compartiendo sus opiniones, comentarios, reseñas, gustos, preferencias y debatiendo sobre diversos temas.

Toda esta información puede ser de gran valor para las empresas, pues analizar y comprender la opinión de los clientes se presenta como un factor clave a la hora de establecer una estrategia empresarial eficiente y competitiva, que se adapte rápidamente a las nuevas tendencias en un entorno cambiante. Hasta hace unos años, las empresas no tenían forma de conocer lo que los clientes opinaban sobre ellas, salvo a través de métodos relativamente caros como encuestas o directamente observando las ventas realizadas. En la actualidad, gracias a Internet, esta información está al alcance de cualquier individuo de forma gratuita, pues los usuarios expresan sus opiniones públicamente y son influidos por las opiniones de otras personas a la hora de tomar sus decisiones de compra.

Lo importante, ya no es simplemente obtener dicha información, sino hacerla entendible y manejable para los interesados. En esta línea, la detección de sentimiento en texto tiene cada vez un mayor interés para diversos sectores, por lo que surge la motivación de realizar un Análisis de Sentimiento a través de Twitter, de las principales compañías del sector asegurador en España. Se ha decidido realizar el análisis mediante la red social Twitter debido a la gran cantidad de información que contiene, al gran número de debates que genera, a su carácter gratuito y a la relativa facilidad para extraer sus comentarios de texto (tweets o tuits), que compondrán nuestra base de datos.

El Análisis de Sentimiento, también conocido como Minería de Opinión, se enmarca dentro del Procesamiento del Lenguaje Natural (PLN), y se define como el conjunto de técnicas computacionales que se utilizan para detectar, extraer y evaluar sentimientos, emociones y subjetividad expresados en un texto (Liu 2010). Mediante este análisis un texto puede, por ejemplo, ser clasificado como neutral, positivo o negativo.

Entenderemos por las principales compañías del sector asegurador español, a partir de ahora denominadas PCSAE, a las ocho empresas con mayor cuota de mercado en dicho sector a Diciembre de 2016, ya que estas copan el 60% de las ventas totales del sector. Para evitar problemas de propiedad de marca, serán nombradas como compañías A, B, C, D, E, F, G y H, sin especificar cuál de ellas corresponde a cada empresa.

Las técnicas de Análisis de Sentimiento están mucho más desarrolladas en idioma inglés que en español y un estudio de estas características sobre compañías del sector asegurador no se ha realizado hasta la fecha. Por ello su realización se considera importante para conocer el sentimiento que presentan los clientes hacia estas compañías, así como para hacerse una idea de la visión que tienen los clientes sobre el sector en su conjunto.

## **1.2.- Objetivos:**

El objetivo general de este trabajo es el de analizar el sentimiento que tienen los clientes hacia las PCSAE. Otros objetivos más específicos que se abordan son los siguientes:

- Obtener comentarios de texto (tuits) en Twitter de manera gratuita
- Elaborar un léxico<sup>1</sup> específico para el sector asegurador español mediante la unificación de otros léxicos ya existentes en castellano.
- Conocer la evolución de sentimiento en el tiempo, a través de una puntuación.
- Clasificar los comentarios en neutrales, positivos o negativos.

---

<sup>1</sup> Un léxico de sentimientos es un diccionario que tiene un valor de sentimiento asociado a cada palabra del mismo.

- Clasificar los comentarios por emociones.
- Obtener las palabras más utilizadas (tópicos) al referirse a cada compañía.
- Graficar la relación entre los tópicos para conocer cuáles son los temas que más se comentan.
- Comparar las PCSAE entre sí a través de indicadores para determinar cuáles presentan un mejor sentimiento en Twitter.

### **1.3.- Estado del arte:**

En este epígrafe se presentan los trabajos más relevantes en el ámbito del Análisis de Sentimiento hasta la fecha. El reto de clasificar el sentimiento en textos se ha abordado de diferentes formas a lo largo de los años; destacando la utilización de técnicas de aprendizaje de maquina supervisado como Support Vector Machines (SVM), Naive Bayes y Maximum Entropy o de aprendizaje de maquina no supervisado. Otro modo de abordar el análisis es mediante la creación manual o semiautomática de recursos informáticos como léxicos y el uso de técnicas lingüísticas basadas en el lenguaje y su estructura. Por otra parte, en las investigaciones más recientes, se han comenzado a utilizar otras técnicas más avanzadas como Latent Semantic Analysis (LSA) o Deep Learning.

El término Análisis de Sentimiento fue usado por primera vez en la obra de Sanjiv & Chen (2001) y Tong (2001) en la predicción de juicios para analizar el comportamiento de mercado. Unos años más tarde, Pang, Lee, Vaithyanathan y Turney, estudiaron el problema de la clasificación por polaridad. En la obra Pang, Lee & Vaithyanathan (2002), se presentó el enfoque de aprendizaje supervisado mediante el uso de técnicas de aprendizaje de máquina (también conocido como aprendizaje automático) a través de los algoritmos Naive Bayes, Maximum Entropy y SVM. Por otra parte, en Turney (2002), se utilizó el denominado enfoque semántico, a través de un clasificador no supervisado. Dicho clasificador determinaba la naturaleza positiva o negativa de un documento basado en la orientación semántica de términos que pueden estar

representados por el algoritmo de PMI-IR (Pointwise Mutual Information- Information Retrieval), basado en la frecuencia de co-ocurrencia de los términos.

Estos estudios únicamente consideraban el aprendizaje a partir de ejemplos con una polaridad positiva o negativa, ignorando las opiniones que muestran un sentimiento neutral. No obstante, existen estudios como el de Koppel et al. (2006), que muestran la importancia que tiene el uso de ejemplos neutrales en el proceso de aprendizaje, ya que demuestran una mejor distinción entre polaridad positiva y negativa si se hace uso de estos ejemplos.

Entre las pocas herramientas para Análisis de Sentimiento de textos en castellano destaca Sentitext, presentado en la obra Ortiz et al. (2010), y que ha mostrado tener un buen desempeño para determinar la polaridad del sentimiento en diversos ejemplos. Sentitext evalúa el texto asignando una cantidad de "estrellas" que gradúan el sentimiento en una escala que asigna cero "estrellas" cuando el comentario es muy negativo, cinco "estrellas" cuando es neutro y diez cuando el comentario es muy positivo. Se basa en el uso de un diccionario de palabras y reglas del lenguaje y utiliza un algoritmo que no implementa ninguna técnica de aprendizaje automático.

Más tarde, en Fernández et al. (2011), se utilizaron los corpus EmotiBlog Kyoto, Emotiblog Phones y JRC para determinar el beneficio que implicaba la utilización de los mismos en la determinación de la intensidad y emoción de las opiniones. Posteriormente, en Saralegi Urizar (2012), se planteó una solución supervisada que comprendía el tratamiento de emoticonos, la negación y tareas de lematización y etiquetado. En el mismo año, Trilla (2012), presentó una clasificación de texto basado en Multinomial Naive Bayes para procesar mensajes de Twitter. En la obra Martínez Cámara (2012), se utilizó el algoritmo SVM para determinar la polaridad de una serie de tuits en castellano. Paralelamente, en Fernández Anta (2012), se compararon los rendimientos de varios clasificadores supervisados.

Un año más tarde, en Ferran & Hurtado (2013) se aplicó el algoritmo SVM a través de la libreríaSVM que se integra a WEKA, realizando un Análisis de Sentimiento a nivel

global y de entidad de los tuit, clasificándolos por tópicos, y finalmente obteniendo la tendencia política de los usuarios. Ese mismo año, en Vilares et al. (2013), se planteó una aproximación híbrida, que combina conocimiento lingüístico obtenido con técnicas de aprendizaje automático que posteriormente entrenaba un clasificador supervisado.

Un artículo interesante que examina el funcionamiento de los clasificadores en el Análisis de Sentimiento de tuits en castellano, es el de Grigori Sidorov et al. (2013). En este artículo exploran diferentes configuraciones para observar cómo cada una de ellas afecta a la precisión de los algoritmos de aprendizaje automático. Concluyen que el hecho de entrenar el sistema con tuits de un dominio diferente al que posteriormente se utilizará, empeora significativamente la precisión de los resultados, llegando a disminuir la prueba de exactitud del 85,8% al 28% con SVM.

En España, diversos grupos de investigación han presentado sus algoritmos sobre el Análisis de Sentimiento en Twitter en idioma castellano, dando así un impulso a esta línea de investigación. Saralegi & San Vicente (2013) consiguieron en el Taller sobre Análisis de Sentimientos en SEPLN (TASS2013) los mejores resultados en la tarea de Análisis de Sentimiento a nivel global de tuit. El método de aprendizaje supervisado que presentaron usa un clasificador SVM que construyen con la herramienta WEKA.

Uno de los factores clave a la hora de llevar a cabo un Análisis de Sentimiento es el léxico utilizado. Centrándonos únicamente en este campo podemos encontrar principalmente estudios orientados a generar diccionarios de palabras en las que éstas se encuentran anotadas con su correspondiente polaridad. En Rao et al. (2009), se realiza un estudio en el que tratan el problema de detectar la polaridad de las palabras como un problema semi-supervisado, para lo que utilizan como recurso la base de datos léxica en inglés SentiWordNet, y el diccionario de sinónimos de OpenOffice. SentiWordNet fue desarrollado para la comunidad científica (Esuli & Sebastiani 2006) y es un recurso lingüístico de gran utilidad, aunque con cierta complejidad de uso, debido a la necesidad de realizar una desambiguación de la palabra antes de poder ser utilizada. Debido al gran número de personas hispanohablantes en todo el mundo, existen estudios como Brooke et al. (2009), Mohammad (2010), Pérez Rosas et al.



(2012), Saralegi & San Vicente (2013) o Cruz et al. (2014) para el desarrollo de léxicos de polaridad en castellano.

#### **1.4.- Metodología:**

Para realizar un Análisis de Sentimiento en Twitter debemos disponer de la fuente de datos de la que sustraer la información que necesitamos. Por ello, el primer paso, descrito en el epígrafe 2, será el de extraer los tuits de Twitter de forma periódica, de modo que obtengamos una base de datos estable en el tiempo y con la mayor cantidad de comentarios posible. Este proceso nos permitirá disponer de 8 bases de datos (una para cada PCSAE) con los tuits que hacen mención a cada una de las empresas en idioma castellano. La base de datos obtenida incluye adicionalmente diversas variables relacionadas con los tuits, en concreto; si ha sido marcado como favorito o no, la fecha de creación, el alias de quien lo ha escrito en Twitter, el número de veces que ha sido retuiteado, la longitud, la latitud y otras variables menos relevantes.

Obtenida la base de datos, debemos prepararla para el análisis (epígrafe 3.1) suprimiendo tuits no deseables. En primer lugar se eliminarán aquellos comentarios que no sean relevantes para el objetivo del análisis, en concreto tuits duplicados, que proporcionan información redundante, y que pueden producirse por simple spam masivo o bien porque a la hora de recolectar los datos se han almacenado doblemente. Por otra parte se suprimirán los tuits vertidos por las propias cuentas oficiales de las PCSAE, pues no aportan ningún sentimiento de los clientes hacia las empresas, considerándose simple publicidad o spam.

Una vez queda preparada la base de datos nos centramos en la variable de texto que contiene los tuits. En este sentido, se realizará una limpieza de texto (epígrafe 3.2) eliminando símbolos extraños, links, números, signos de puntuación y convirtiendo todo el texto a minúscula.

Con el texto limpio, debemos elegir que léxico utilizamos para clasificar cada comentario como positivo, negativo o neutral. Como comentamos anteriormente, este

campo está mucho más desarrollado en idioma inglés que en castellano, aunque actualmente existen varios léxicos en castellano. En el epígrafe 4 se define todo el proceso para la creación del léxico que se ha utilizado partiendo de dos léxicos ya existentes, de modo que finalmente obtengamos un léxico específico para el sector asegurador, compuesto por un listado de palabras clasificadas como positivas o negativas que nos ayudarán a determinar la polaridad de cada comentario.

En el epígrafe 5, utilizando la base de datos y el léxico, se determinará la polaridad del texto utilizando el enfoque semántico, caracterizado por el uso de léxicos con orientación semántica de polaridad u opinión. El proceso se realiza mediante la función *score\_sentiment()*, que separa las palabras de cada comentario y comprueba si coinciden con los términos del léxico, negativos o positivos, de modo que cada comentario recibirá un -1 si contiene una palabra negativa y 1 si contiene una palabra positiva. Finalmente se le asignará una puntuación de polaridad a cada tuit mediante la suma de valores de polaridad y la asignación de un “bonus” en función de los favoritos y retuit que contenga cada comentario.

Complementario al análisis de polaridad, se realizará un análisis de emociones y de frecuencias, y se estudiará la evolución de las puntuaciones de las PCSAE en el tiempo y se compararán entre sí.

## 2.- Extracción de datos en Twitter.

En primer lugar debemos extraer los tuits que conformarán la base de datos a analizar. La extracción de datos, como se ha comentado anteriormente, se realizará en la red social Twitter, a través de la librería “twitterR” (Jeff Gentry 2015) del software estadístico R.

Twitter es un término inglés que en nuestro idioma significa “gorjear” o “trinar”. Es una de las redes sociales más populares y utilizadas en la actualidad, debido a su facilidad de uso, rápido acceso, carácter gratuito y simplicidad en su sistema de registro. Cuenta con más de 4,5 millones de usuarios activos en España y más de 300 millones en todo el mundo, convirtiéndose así en una de las mayores fuentes de información en tiempo real.

Consiste en una red de microblogging que permite escribir y leer comentarios en Internet que no superen los 140 caracteres, denominados tuits. Cuando un usuario publica un mensaje en su página de Twitter, es enviado automáticamente a todos sus seguidores o followers, es decir, a todos los usuarios que han escogido la opción de recibirlos. Dicho mensaje también puede ser visto de forma inmediata en el perfil del usuario. El microblogging es una variante de los blogs y su principal diferencia radica en la brevedad de sus mensajes y en su facilidad de publicación, ya que pueden enviarse desde el móvil, ordenador o dispositivos con software de mensajería instantánea.

La librería “twitterR” de R permite trabajar a través de la Application Programming Interfaces (API) de Twitter. Las APIs son un conjunto de comandos, funciones y protocolos informáticos que permiten a los desarrolladores interactuar con el sistema operativo o con otro programa. Twitter ofrece tres APIs aplicables a necesidades diferentes; Streaming API, REST API y Search AP. El Streaming API proporciona un subconjunto de tuits prácticamente en tiempo real, pudiéndose obtener estos mediante una muestra aleatoria o un filtrado por palabras clave o usuarios. Por otro lado, REST API ofrece a los desarrolladores el acceso al núcleo de los datos de Twitter y

Search API suministra los tuits que se ajustan a la consulta solicitada y es posible filtrar por cliente utilizado, lenguaje y localización. Esta última API ofrece una información limitada del tuit, con una profundidad en el tiempo de 7 días. También cabe decir que el Search API y el REST API tienen una limitación de 150 peticiones a la hora por usuario o IP. Una vez resumidas las tres tipos de APIs, cabe decir que el grueso de datos utilizados en este trabajo serán extraídos mediante la utilización de REST API y Search API.

A través de la API tratamos de obtener una base de datos para cada PCSAE que contenga los tuits en los que se nombra a dichas compañías a lo largo del tiempo. El proceso de recolección de tuits se irá repitiendo semana a semana con el fin de obtener la mayor cantidad de datos posibles.

Para poder descargar la información de Twitter desde R, es necesario acceder a una cuenta de esta red social. Será necesario registrar una aplicación en Twitter, para lo que se accede a <https://dev.twitter.com/apps> y se pulsa en “Create new app”. Una vez allí se cumplimentan los campos que se solicitan y al finalizar solo queda entrar en la pestaña “Keys and Access Tokens”, pulsar en el botón de “Generate My Access Token and Token Secret” y copiar los credenciales siguientes; “consumer\_key”, “consumer\_secret”, “access\_token” y “access\_secret”.

Una vez que hemos obtenido las claves para acceder a la API de Twitter, instalaremos la librería “twitterR” en R. De este modo, mediante la función `setup_twitter_oauth()`, podremos autenticarnos desde R, extraer los tuits a través de `searchTwitter()` y convertirlos en formato .csv mediante las funciones `twListToDF()` y `write.csv()`.

La función de búsqueda se puede acotar por número de tuits (`n=`) y lengua (`lang =`) entre otras variables, de modo que nos permite realizar la búsqueda, captando los tuits que contengan el nombre (`searchString=`) de cada PCSAE en idioma castellano.

Cabe decir que para evitar generalidades que distorsionen gravemente el análisis, en algunos casos la captación de tuits se realiza mediante la búsqueda de dos palabras. Por ejemplo, si buscamos en Twitter la palabra Zurich, la mayoría de comentarios

obtenidos pueden referirse a la ciudad y esto distorsionaría el análisis, por lo que la búsqueda se realizaría como Zurich Seguros. En otros casos, debido al carácter de grupo de algunas de las PCSAE, se realiza la búsqueda para varios términos cuando existen compañías con una potente imagen dentro del grupo. Por ejemplo, en el caso del Grupo Catalana Occidente, donde Plus Ultra Seguros tiene un gran peso. En la tabla 1 podemos observar, para cada compañía, el número de tuits obtenidos en el intervalo de tiempo que vamos a estudiar (01/04/2017 a 28/08/2017). Cabe destacar en relación al número de tuits, que existen gran cantidad de duplicados que tendremos que eliminar posteriormente.

**Tabla 1.- Resumen datos captados de las PCSAE.**

| Compañía | Nº tuits* |
|----------|-----------|
| <b>A</b> | 7895      |
| <b>B</b> | 7469      |
| <b>C</b> | 1411      |
| <b>D</b> | 2006      |
| <b>E</b> | 37169     |
| <b>F</b> | 5877      |
| <b>G</b> | 4340      |
| <b>H</b> | 2389      |

\* Contiene duplicados.

*Fuente: Elaboración propia.*

Un ejemplo de cómo se captan y almacenan los tuits en código de R es el siguiente:

```
library("twitteR")
api_key <- "y86RrXUyQEZ####"
api_secret <- "Y6ET3VJKSxfOHj5FUmylpxP###"
access_token <- "375386021-eND5ItaSTnxP1b###"
access_token_secret <- "QZKSvU60zdg2Mh1Cd4rhli###"
setup_Twitter_oauth(api_key, api_secret, access_token,
access_token_secret)
ase.tuits <- searchTwitter(searchString="Aseguradora", n=10000,
lang="es")
```

```
aseguradora <- twListToDF(ase.tuits)
write.csv(aseguradora, file="aseguradoraX.csv")
```

Como se ha comentado anteriormente, este proceso se repite semanalmente, de modo que los datos obtenidos cada semana se van acoplando a la base de datos conjunta de cada compañía del siguiente modo:

```
a <- read.csv("aseguradora.csv")
b <- read.csv("aseguradoraX.csv")
c <- rbind(a,b)
write.csv(c, file="aseguradora.csv")
```

## 3.- Procesamiento de datos.

### 3.1.- Preparación de la base de datos:

Una vez almacenados los datos, y para evitar incurrir en errores en el posterior análisis, es necesario efectuar una revisión de los mismos. La actuación se debería llevar fundamentalmente en dos frentes: crear filtros para eliminar posible spam y suprimir información no relevante para el análisis.

El primero de estos dos frentes será abordado mediante la eliminación de los tuits realizados por las propias cuentas oficiales de las PCSAE, que no aportan ningún sentimiento de los clientes hacia las empresas. A priori, esta decisión aumentará nuestros resultados en cuanto a porcentaje de comentarios neutrales se refiere, pues reducirá el número de negativos y sobre todo de positivos. Este hecho no es motivo de preocupación, pues los neutrales simplemente nos van a indicar el nivel de publicidad y comentarios sin polaridad que existe en cada compañía, que a priori debe rondar un mismo porcentaje para todas las compañías estudiadas. Un ejemplo del código en R utilizado para estos casos es el siguiente:

```
datossent <- datossent[!datossent$screenName=="MAPFRE", ]
datossent <- datossent[!datossent$screenName=="MAPFRE_Atiende", ]
datossent <- datossent[!datossent$screenName=="MAPFRE_ES", ]
datossent <- datossent[!datossent$screenName=="MAPFRE_MX", ]
```

La segunda cuestión tiene que ver con la supresión de información no relevante para el análisis. En este sentido, como comentamos en el epígrafe 2, la base de datos se va actualizando semanalmente con nuevos tuits, lo que puede provocar que existan tuits duplicados, ya que la API de Twitter, que en principio proporciona la información de los últimos 7 días, no cumple este intervalo temporal con exactitud. Otra forma de que se creen tuits duplicados es mediante el envío masivo de spam de forma automática. La forma de abordar este tema es mediante la eliminación de los tuits duplicados. En R eliminamos los duplicados de la siguiente forma:

```
datossent <- datossent[!duplicated(datossent$text), ]
```

El número de comentarios para cada compañía queda reflejado en la tabla 2.

**Tabla 2.- Resumen datos de las PCSAE.**

| Compañía | Nº tuits* | Nº tuits | Suprimidos | % Suprimidos |
|----------|-----------|----------|------------|--------------|
| <b>A</b> | 7895      | 1835     | 6060       | 76,8%        |
| <b>B</b> | 7469      | 1975     | 5494       | 73,6%        |
| <b>C</b> | 1411      | 731      | 680        | 48,2%        |
| <b>D</b> | 2006      | 835      | 1171       | 58,4%        |
| <b>E</b> | 37169     | 17214    | 19955      | 53,7%        |
| <b>F</b> | 5877      | 2935     | 2942       | 50,1%        |
| <b>G</b> | 4340      | 1447     | 2893       | 66,7%        |
| <b>H</b> | 2389      | 1145     | 1244       | 52,1%        |

\* Contiene duplicados.

*Fuente: Elaboración propia.*

Una vez limpiados los datos de tuits corporativos y de duplicados, se procederá a la limpieza específica del texto a analizar.

### 3.2.- Limpieza de texto:

A continuación nos centramos en la variable de texto que contiene los tuits. La limpieza de los tuits se realizará mediante la función *gsub()*, que busca por patrones de texto dentro de cada uno de los elementos de un vector columna, para luego reemplazarlos por otro texto que precisemos. A continuación tenemos un ejemplo de cómo funciona:

```
texto <- gsub(pattern="Vlc", replacement="Valencia", x = texto)
```

Las instrucciones que da este código a R son; trabaja sobre la vector columna que contiene el texto, llamado "texto", busca "Vlc", reemplázalo por "Valencia" y sobrescribe el resultado sobre la misma columna "texto".



En nuestro caso nos interesa eliminar diferentes símbolos y caracteres no relevantes para el análisis del texto, por lo que en el argumento *pattern*= escribiremos dichos símbolos o caracteres y en el argumento *replacement*= escribiremos " ", indicando que deje vacío el espacio donde existía dicho valor. Los patrones que queremos eliminar de nuestro texto son varios y se definen a continuación, junto a otras transformaciones.

- Links a páginas web (http/s): En Twitter es habitual que los usuarios hagan referencia a enlaces de páginas web. Estos enlaces no aportan información al análisis de sentimiento y pueden provocar problemas debido a que contienen muchos caracteres diferentes. Su eliminación se realiza de dos formas para cerciorarnos de su supresión completa:

```
texto = gsub("(f|ht)tp(s?):/(.*)[.][a-z]+", "", texto)
texto = gsub("http\\w+", "", texto)
```

- Nombres de usuarios (@): En Twitter, cada usuario dispone de un alias, precedido del símbolo @, por ejemplo @NombreUsuario. Cuando alguien escribe un tuit, puede mencionar a otros usuarios con este alias, ya sea porque el tema esta relacionados con ellos o porque el tuit va dirigido a ellos. Sin embargo, símbolos como este pueden provocar problemas, dado que es un elemento no gramatical característico de este medio. Por otra parte, el nombre en caracteres de texto del usuario mencionado tampoco es relevante a la hora de clasificar los comentarios por polaridad. El código para su eliminación sería el siguiente:

```
texto = gsub("@\\w+", "", texto)
```

- Retuits (RT) a otros usuarios: El RT es una funcionalidad de Twitter que sirve para re-publicar un comentario realizado por otro usuario. De modo, que cuando se hace un RT el comentario sigue la estructura; "RT @NombreUsuario: tuit...". Dicho esto, observamos que para el análisis del texto no nos interesa ni la palabra RT ni el nombre de usuario que lo precede, por lo que se suprimen ambos:

```
texto <- gsub("(RT|via) ((?:\\b\\W*@\\w+)+)", "", texto)
```

- **Hashtags (#):** Los hashtags son términos incluidos en Twitter que los usuarios preceden del símbolo # con el objetivo de etiquetar sus mensajes. Al hacer click sobre un hashtag el usuario es redireccionado al conjunto de tuits que contienen la misma etiqueta. Un hashtag puede referirse a eventos muy específicos (#Forinvest2017), ser utilizado como medio para enfatizar una palabra contenida en el tuit (Accidentes S.A. es la #peor aseguradora de España) o para resumir las principales conclusiones de un tuit (#sonlopeor). En este caso, sí que interesa el texto que precede del símbolo #, por lo que únicamente el símbolo # es borrado. Con el siguiente código, no solo eliminamos el hashtag, sino también otros signos de puntuación irrelevantes para el análisis; !"#\$%&'()\*+,-./:;<=>?@[\\]^\_`{|}~.

```
texto = gsub('[:punct:]', "", texto)
```

- **Números y otros:** También eliminamos caracteres numéricos, caracteres de control y espacios innecesarios:

```
texto = gsub("[[:digit:]]", "", texto)
```

```
texto = gsub('[:cntrl:]', "", texto)
```

```
texto = gsub('\\d+', '', texto)
```

```
texto = gsub("[ \\t]{2,}", "", texto)
```

- **Mayúsculas:** Para estandarizar todo el texto se transforman todas las palabras a minúsculas:

```
texto = tolower(texto)
```

Una vez tenemos limpio el texto, se puede comenzar a clasificar por polaridad, para lo que primeramente necesitaremos tener creado el léxico específico para el sector asegurador.

## 4.- Léxico específico sector asegurador:

Se decide crear un léxico específico para el sector asegurador español a partir de otros léxicos existentes en idioma castellano. Se construye como específico basándonos en el trabajo Grigori Sidorov et al. (2013), en el que llegan a la conclusión de que el hecho de entrenar el sistema con tuits de un ámbito diferente al que posteriormente se aplicará, empeora significativamente la precisión de los resultados. En este caso, aunque no utilizamos un enfoque de aprendizaje automático, se cree que la mejor opción es la de crear un léxico con palabras positivas y negativas que capte de manera holgada las palabras típicas del sector que estudiamos, ya que estas pueden ser muy significativas para la precisión del análisis. Por otro lado, la omisión de palabras de otros ámbitos nos permite captar mejor los comentarios neutrales, evitando que sean clasificados como positivos o negativos cuando no lo son.

### 4.1.- Léxicos en castellano:

Antes de explicar cómo se ha creado nuestro léxico específico para el sector asegurador vamos a mostrar cuáles son los léxicos que sirven de base para el mismo mediante una revisión bibliográfica de estos. Los léxicos de los que partimos son el construido en Learning Sentiment Lexicons in Spanish (Pérez Rosas et al. 2012), el ElhPolar (Saralegi & San Vicente 2013) y el creado en el trabajo Building Layered, Multilingual Sentiment Lexicons at Synset and Lemma Levels (Cruz et al. 2014). Estos tres léxicos, a los que llamaremos “Léxicon”, “ElhPolar” y “Senticon” respectivamente, están realizados en idioma castellano y son de carácter gratuito a fines de investigación.

Léxicon: El léxico obtenido en Learning Sentiment Lexicons in Spanish (Pérez Rosas et al. 2012) contiene un léxico de sentimiento castellano basado en las anotaciones de sentimiento manual del diccionario OpinionFinder (Wiebe et al. 2005). Puesto que la subjetividad y la polaridad son cualidades que han demostrado ser más fuertemente expresadas en inglés (Wiebe & Mihalcea 2006), transfieren las anotaciones manuales a

la English WordNet mediante la aplicación de SentiWordNet (Esuli & Sebastiani 2006). El criterio que aplican para seleccionar la polaridad final es el de coincidir la opinión asignada manualmente al sentido presente en SentiWordNet. Finalmente traducen al idioma castellano con la ayuda de Spanish Wordnet.

ElhPolar: En cuanto al léxico de polaridad ElhPolar para el idioma castellano fue creado a partir de diferentes fuentes, e incluye palabras negativas y positivas. En "Elhuyar en TASS 2013" (Saralegi & San Vicente 2013) puede encontrarse una descripción detallada del contenido, así como la forma en que se construyó el léxico, no obstante se van a explicar los aspectos más importantes del mismo. Para la creación del ElhPolar utilizan dos fuentes diferentes:

1. Un léxico de polaridad inglés ya existente (Wilson et al. 2005) que incluía palabras positivas y negativas y que fue traducido al castellano. Debido a la traducción, algunas palabras mostraban polaridad ambigua, por lo que fueron resueltas y verificadas manualmente por dos anotadores, adoptando un proceso semiautomático para maximizar el resultado del léxico final.
2. Como segunda fuente se utilizan las palabras más asociadas con una cierta polaridad, extraídas automáticamente del corpus de entrenamiento. Además, se añaden palabras de uso coloquial de la fuente [www.ual.es/EQUAL-ARENA/Documentos/coloquio.pdf](http://www.ual.es/EQUAL-ARENA/Documentos/coloquio.pdf), así como en [www.dictionaryjerga.com](http://www.dictionaryjerga.com), que incluye vocabulario coloquial editado por los usuarios.

Senticon: Por último, el léxico construido en el trabajo Cruz et al. (2014) utiliza la WordNet-Affect (Strapparava et al. 2006) como fuente de semillas positivas y negativas. Los clasificadores fueron entrenados a partir de las distintas fuentes de información usando dos algoritmos de clasificación distintos (Rocchio y SVM), que fueron combinados en una etapa de meta-aprendizaje, obteniéndose finalmente tres clasificadores regresionales capaces de inducir valores de positividad, negatividad y objetividad en el intervalo [0, 1]. El cálculo global de la polaridad trata de refinar en su conjunto los valores de positividad y negatividad asignados a cada palabra, a partir de distintos tipos de relaciones entre ellos.

#### 4.2.- Metodología de la construcción del léxico:

El método para crear nuestro léxico específico para el sector asegurador constará de los siguientes pasos:

1. Se selecciona un conjunto de tuits de prueba para evaluar los distintos léxicos. Este conjunto de prueba se selecciona de forma aleatoria entre todas las entidades estudiadas y en todos los intervalos temporales, consiguiéndose una base de datos de 2620 tuits.
2. Se clasifica cada comentario del conjunto de prueba manualmente como negativo, neutral o positivo.
3. Se aplica el procedimiento de detección de polaridad con cada léxico obteniendo una matriz de confusión.
4. Se realizan pruebas de precisión y exactitud con los léxicos existentes para seleccionar cual será el léxico o conjunto de léxicos base para la siguiente etapa.
5. Se comprueba de forma manual que resultados obtenidos son incoherentes y se añaden o eliminan determinadas palabras siempre que las pruebas de precisión y exactitud mejoren con los cambios.

A la hora de comprobar la precisión y exactitud de los resultados obtenidos con cada léxico utilizado se llevarán a cabo diferentes pruebas. En primer lugar se construirá una matriz de confusión, que se define como una herramienta visual que se utiliza en el aprendizaje supervisado, donde cada columna que posee representa el número de predicciones de cada clase, mientras que cada fila representa las instancias de la clase real. La matriz de confusión que obtenemos tiene la siguiente forma:

*Sea:*

*VN = Verdadero negativo.*

*FN (n) = Falso negativo clasificado como neutral.*

*FN (P) = Falso negativo clasificado como positivo.*

*Vn = Verdadero neutral.*

*Fn (N) = Falso neutral clasificado como negativo.*

*Fn (P) = Falso neutral clasificado como positivo.*

*VP = Verdadero positivo.*

*FP (N) = Falso positivo clasificado como negativo.*

*FP (n) = Falso positivo clasificado como neutral.*

**Tabla 3.- Matriz de confusión.**

| Empresa X | Clasificado como: |         |          |
|-----------|-------------------|---------|----------|
| Real:     | Negativo          | Neutral | Positivo |
| Negativo  | VN                | Fn (N)  | FP (N)   |
| Neutral   | FN (n)            | Vn      | FP (n)   |
| Positivo  | FN (P)            | Fn (P)  | VP       |

Fuente: Elaboración propia.

Uno de los principales beneficios de las matrices de confusión es que facilitan ver si el sistema se confunde entre clases. Una vez obtenida la matriz de confusión se calculan los siguientes ratios que nos mostrarán la exactitud y precisión del análisis con cada léxico evaluado:

Accuracy o exactitud: Es el porcentaje de valores verdaderos (clasificados correctamente) respecto al total. Hace referencia a la conformidad de un valor medido con su valor verdadero, es decir, se refiere a cuán cerca del valor real se encuentra el valor medido. En términos estadísticos, la exactitud está relacionada con el sesgo de una estimación.

$$Accuracy = \frac{VN + Vn + VP}{Total}$$

Precisión: La precisión es el ratio entre el número de valores verdaderos de una determinada polaridad entre el total de valores que son realmente de dicha polaridad, ya sean clasificados correctamente o no por el procedimiento utilizado. Por ejemplo, para los positivos, será el porcentaje de positivos que han sido correctamente clasificados como positivos dentro de todos los positivos.

$$P = \frac{VP}{VP + FP(N) + FP(n)} ; N = \frac{VN}{VN + FN(n) + FN(P)} ; n = \frac{Vn}{Vn + Fn(N) + Fn(P)}$$

### 4.3.- Creación del léxico:

Evaluamos los tres léxicos nombrados en el epígrafe 4.1 con la metodología explicada en el epígrafe 4.2 y obtenemos los siguientes resultados.

**Tabla 4.- Comparación matrices de confusión.**

| Léxico            | <i>Clasificado como:</i> |                |                 |
|-------------------|--------------------------|----------------|-----------------|
| <i>Realmente:</i> | <b>Negativo</b>          | <b>Neutral</b> | <b>Positivo</b> |
| <b>Negativo</b>   | 18                       | 91             | 28              |
| <b>Neutral</b>    | 134                      | 1910           | 424             |
| <b>Positivo</b>   | 0                        | 11             | 4               |

| EhIPolar          | <i>Clasificado como:</i> |                |                 |
|-------------------|--------------------------|----------------|-----------------|
| <i>Realmente:</i> | <b>Negativo</b>          | <b>Neutral</b> | <b>Positivo</b> |
| <b>Negativo</b>   | 71                       | 49             | 17              |
| <b>Neutral</b>    | 267                      | 1450           | 751             |
| <b>Positivo</b>   | 1                        | 4              | 10              |

| Senticon          | <i>Clasificado como:</i> |                |                 |
|-------------------|--------------------------|----------------|-----------------|
| <i>Realmente:</i> | <b>Negativo</b>          | <b>Neutral</b> | <b>Positivo</b> |
| <b>Negativo</b>   | 9                        | 64             | 64              |
| <b>Neutral</b>    | 13                       | 1654           | 801             |
| <b>Positivo</b>   | 0                        | 3              | 12              |

*Fuente: Elaboración propia.*

**Tabla 5.- Comparación ratios de evaluación 1.**

| Ratio / Léxico       | <b>Léxico</b> | <b>EhIPolar</b> | <b>Senticon</b> |
|----------------------|---------------|-----------------|-----------------|
| <b>Accuracy =</b>    | <b>73,7%</b>  | 58,4%           | 63,9%           |
| <b>Precisión N =</b> | 13,1%         | <b>51,8%</b>    | 6,6%            |
| <b>Precisión n =</b> | <b>77,4%</b>  | 58,8%           | 67%             |
| <b>Precisión P =</b> | 26,7%         | 66,7%           | <b>80%</b>      |

*Fuente: Elaboración propia.*

Observamos que el Léxico es el léxico que más porcentaje de exactitud tiene, aunque este resultado se debe a su gran porcentaje de identificación de neutrales. Lo que realmente nos interesa es detectar los comentarios negativos y positivos, por lo que se decide descartar el léxico nombrado y seguir haciendo comprobaciones con los otros dos, ya que el EhlPolar es el que mejor detecta los negativos (51,8%) y el Senticon detecta el 80% de los positivos, lo que hace pensar que una de sus combinaciones pueda proporcionar resultados muy fiables. Ahora vamos a realizar algunas pruebas con estos dos diccionarios, para comprobar si alguna de las combinaciones de ambos mejora los resultados. Sea P=EhlPolar y S=Senticon:

**Tabla 6.- Comparación ratios de evaluación 2.**

| Léxico / Ratio            | <i>Accuracy</i> | <i>Precisión N</i> | <i>Precisión n</i> | <i>Precisión P</i> |
|---------------------------|-----------------|--------------------|--------------------|--------------------|
| <b>1. posP - negP</b>     | 58,4%           | <b>51,8%</b>       | 58,8%              | 66,7%              |
| <b>2. posP - negS</b>     | 63,7%           | 32,1%              | 65,9%              | 0%                 |
| <b>3. posP - negPyS</b>   | 58,3%           | <b>51,8%</b>       | 58,6%              | 66,7%              |
| <b>4. posS - negS</b>     | 63,9%           | 6,6%               | <b>67%</b>         | <b>80%</b>         |
| <b>5. posS - negP</b>     | <b>64,4%</b>    | 39,4%              | 65,7%              | 73,3%              |
| <b>6. posS - negPyS</b>   | 64,2%           | 40,1%              | 65,5%              | 73,3%              |
| <b>7. posPyS - negPyS</b> | 54,9%           | 35,8%              | 55,8%              | <b>80%</b>         |
| <b>8. posPS - negP</b>    | 55%             | 35%                | 56%                | <b>80%</b>         |
| <b>9. posPyS - negS</b>   | 53,9%           | 5,1%               | 56,4%              | <b>86,7%</b>       |

*Fuente: Elaboración propia.*

En términos de exactitud los mejores resultados son los de las combinaciones 5 y 6 y aunque no destacan como los mejores en precisión en ninguna de las polaridades, son los más equilibrados junto a las combinaciones 1 y 3. Cualquiera de estas combinaciones se podría tomar como base de nuestro léxico, pero se toma la decisión de seleccionar la combinación 6. Se descartan la 1 y la 3 debido a su exactitud menor y la menor precisión en neutrales y positivos, y finalmente se selecciona la 6 en detrimento de la 5 debido que se prefiere el leve porcentaje de mejora en negativos y el hecho de que el conjunto de palabras negativas sea más amplio al contener las



negativas de ambos diccionarios. Esto último nos puede ayudar si nuestra base de datos es más amplia y contiene nuevas palabras.

A partir de esta base, el léxico se va probando de forma manual conforme van entrando nuevos comentarios en nuestra base de datos, de forma que se prueba la coherencia de los comentarios en los clasificados de forma errónea y se van eliminando y añadiendo nuevas palabras siempre que las pruebas de evaluación mejoren. Las matrices de confusión obtenidas son muy numerosas y no es óptimo mostrarlas en este trabajo, por lo que solo se va a mostrar la última, que conseguimos con un léxico que nos proporciona unos resultados de exactitud y precisión que rondan en todos los casos el 90%, muy superiores a los resultados obtenidos con cualquiera de las otras combinaciones. Este léxico específico para el sector asegurador, que a partir de ahora llamaremos Lexiseg, presenta los siguientes resultados en la base de datos de prueba:

**Tabla 7.- Matriz de confusión y ratios de evaluación Lexiseg.**

| Lexiseg         | <i>Clasificado como:</i> |                |                 |
|-----------------|--------------------------|----------------|-----------------|
| <i>Real:</i>    | <b>Negativo</b>          | <b>Neutral</b> | <b>Positivo</b> |
| <b>Negativo</b> | 122                      | 13             | 2               |
| <b>Neutral</b>  | 157                      | 2272           | 39              |
| <b>Positivo</b> | 0                        | 1              | 14              |

Accuracy = 91,9%

Precisión N = 89,1%

Precisión n = 92,1%

Precisión P = 93,3%

*Fuente: Elaboración propia.*

Este es el léxico con el que finalmente realizaremos el análisis de sentimiento de las entidades aseguradoras del sector asegurador en España. El Lexiseg contiene como base la combinación 6 de la tabla 6 y las palabras que manualmente se han añadido al notar que la incursión de las mismas mejora el léxico. Estas palabras añadidas, tanto

negativas como positivas son palabras que si bien no son exclusivas del sector asegurador, se detecta que tienen una gran influencia en el mismo, por lo que su incursión es clave a la hora de obtener unos resultados de exactitud tan altos. A continuación se muestra una tabla con 10 palabras de ejemplo para cada polaridad, ya que en total se añaden 50 palabras positivas y 93 negativas.

**Tabla 8.- Ejemplo palabras añadidas manualmente al léxico.**

| <b>Positivas</b> | <b>Negativas</b> |
|------------------|------------------|
| Recomiendo       | Reclamo          |
| Compartimos      | Denuncio         |
| Solucionadores   | Malditos         |
| Cercanos         | Mentirosos       |
| Orgullosos       | Obligarme        |
| Arregla          | Páguenme         |
| Ayudado          | Robaron          |
| Rápido           | Chapuza          |
| Resuelto         | Ascazo           |
| Eficiente        | Esperando        |

*Fuente: Elaboración propia.*

## 5.- Polaridad de sentimiento.

### 5.1.- Determinación de la polaridad:

Una vez disponemos de la base de datos y del léxico podemos determinar la polaridad de cada tuit referido a cada una de las entidades estudiadas. El proceso se realiza mediante la función creada en Breen (2011) denominada *score\_sentiment()*, que comprueba si las palabras que contiene un tuit coinciden con los términos del léxico, negativos o positivos. De esta forma, se puede determinar el número de palabras positivas y negativas que contiene cada uno de los comentarios. Otras de las variables asociadas a cada tuit son el número de retuits que se han hecho del mismo, así como el número de personas que lo han marcado como favorito.

De este modo podemos calcular la polaridad de los tuits, que vendrá dada por la variable *Score*. Dicha polaridad podrá ser positiva o negativa, existiendo la posibilidad de categorizarse también como neutral, de modo que:

$$Polaridad = \begin{cases} Si\ Score < 0 ; & Negativa \\ Si\ Score = 0 ; & Neutral \\ Si\ Score > 0 ; & Positiva \end{cases}$$

A continuación se muestra el proceso para el cálculo de la variable *Score*. El valor base de esta variable se calcula como el número de palabras positivas menos el número de palabras negativas que contiene.

Sea:

*POS* = Número de palabras positivas.

*NEG* = Número de palabras negativas.

$$Score\ base\ (SB) = POS - NEG$$

En la tabla 9 se muestran 3 ejemplos de cálculo, uno para cada una de las polaridades, del cálculo de la variable *Score* base. A este valor se le aplicará más tarde un bonus en función de los favoritos y retuit que contiene cada comentario para obtener el valor del *Score* final.

**Tabla 9.- Ejemplo determinación de la polaridad y SB.**

| Tuit     | Texto   | Positivas | Negativas | SB | Polaridad |
|----------|---|-----------|-----------|----|-----------|
| <b>1</b> | <b>Imposible</b> contactar con ustedes para un siniestro! Qué <b>horror</b> , que <b>inseguridad</b> da su compañía de seguros. | 0         | 3         | -3 | Negativa  |
| <b>2</b> | Creo que siendo una <u>buena</u> empresa de seguros la publicidad es <b>inadecuada</b> .  | 1         | 1         | 0  | Neutral   |
| <b>3</b> | Gracias, ya fue resuelto de manera <u>oportuna</u> y <u>eficiente</u> .   | 2         | 0         | 2  | Positiva  |

*Fuente: Elaboración propia.*

Se sumarán o restarán al Score el número de favoritos que contiene (dependiendo de su polaridad), pues se considera que si alguien marca un tuit como favorito es porque se siente identificado plenamente con el sentimiento que éste expresa. Por otra parte, se sumarán o restarán la mitad del número de retuit que contenga cada tuit, suponiendo que un individuo que comparte un tuit de otro usuario coincide en cierta manera su opinión o quiere darla a conocer a sus seguidores. Por lo tanto, el cálculo del *Score* final es el siguiente:

Sea:

*FAV* = Número de personas que han guardado el tuit como favorito.

*RET* = Número de personas que han retuiteado el tuit.

$$Score = \begin{cases} Si SB > 0; Score = SB + FAV + RET * 0.5 \\ Si SB = 0; Score = 0 \\ Si SB < 0; Score = SB - FAV - RET * 0.5 \end{cases}$$

**Tabla 10.- Ejemplo determinación de la puntuación Score.**

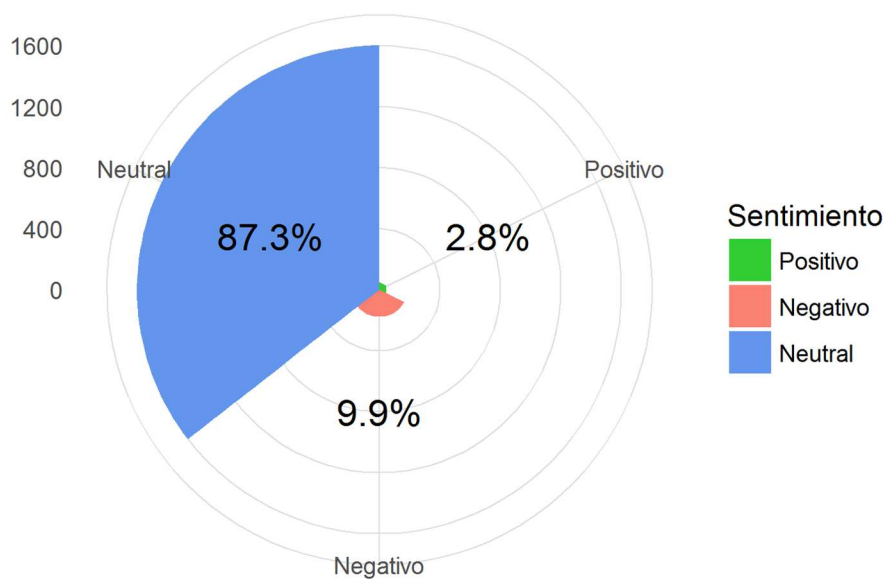
| Tuit     | SB | Retuits | Favoritos | Polaridad | Score                           |
|----------|----|---------|-----------|-----------|---------------------------------|
| <b>1</b> | -3 | 6       | 2         | Negativa  | $-3 - (6/2) - 2 =$<br><b>-8</b> |
| <b>2</b> | 0  | 2       | 1         | Neutral   | <b>0</b>                        |
| <b>3</b> | 2  | 2       | 5         | Positiva  | $2 + (2/2) + 5 =$<br><b>8</b>   |

*Fuente: Elaboración propia.*

## 5.2.- Análisis de sentimiento:

Una vez obtenida esta puntuación se pueden obtener diversos indicadores y gráficos que nos permiten observar el sentimiento que muestran los clientes hacia cada empresa. Por ejemplo, el porcentaje de tuits positivos, negativos y neutros respecto al total, cuantos comentarios negativos se realizan por cada comentario positivo, el *Score* promedio para cada empresa o una calificación creada a partir de estas variables que se explicará posteriormente. Los resultados y gráficos obtenidos en la determinación de la polaridad de sentimiento se van a mostrar tomando como ejemplo la compañía A. Como se ha comentado anteriormente, mediante la función *score\_sentiment()* obtenemos el número de comentarios negativos, positivos y neutrales para cada empresa, así como su porcentaje respecto al total.

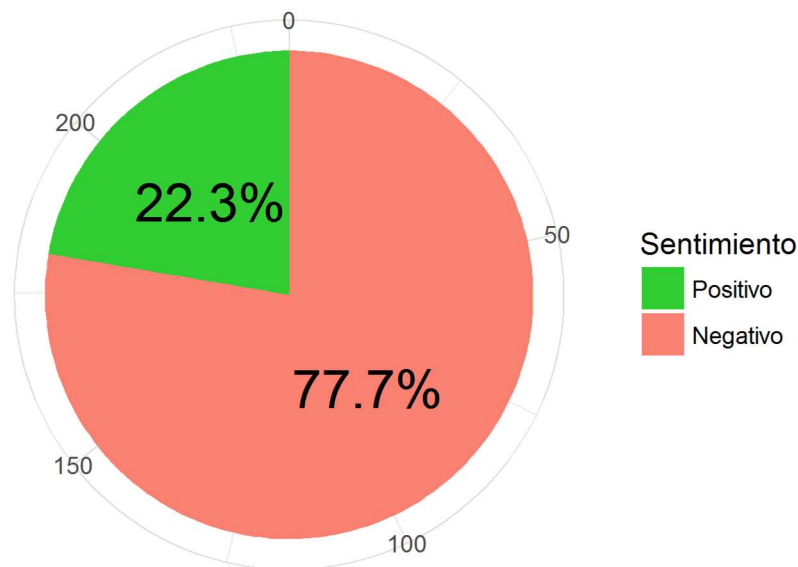
**Gráfico 1.- Polaridad de sentimiento compañía A con neutrales.**



*Fuente: Elaboración propia.*

Como se puede observar en el gráfico 1, destacan los comentarios neutrales muy por encima del resto, hecho que se produce en todas las empresas del sector como veremos posteriormente. Si obtenemos el porcentaje de positivos y negativos en relación al total de tuits no neutrales, se puede ver más rápidamente la dualidad entre positivos y negativos.

Gráfico 2.- Polaridad de sentimiento compañía A.

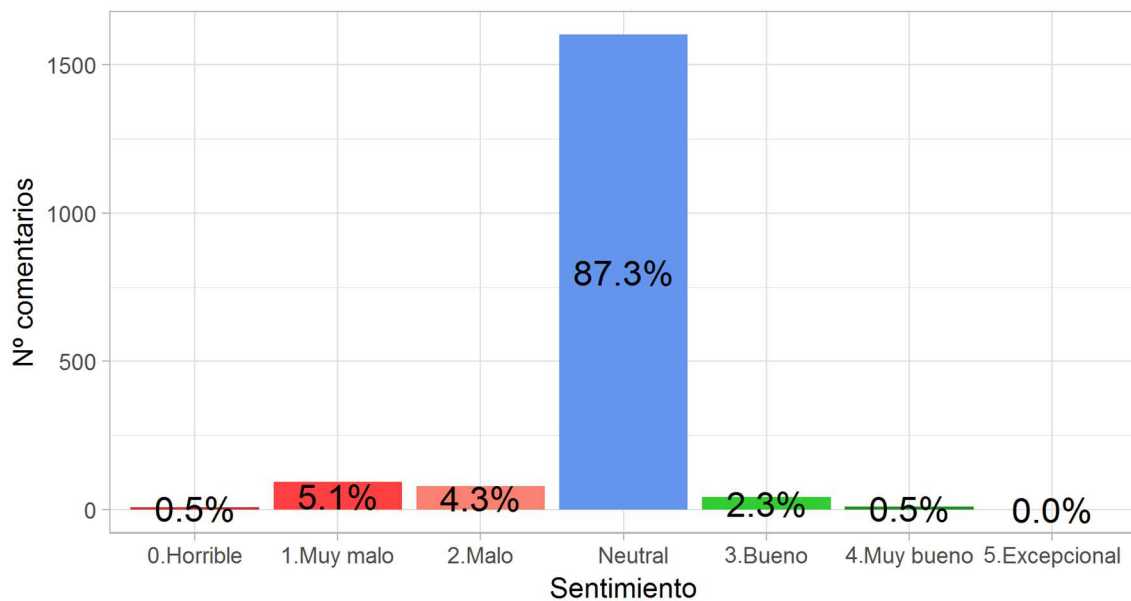


Fuente: Elaboración propia.

En este caso se escriben casi 4 tuits negativos por cada positivo. Dependiendo del *Score*, un tuit positivo o negativo puede serlo con más o menos intensidad, dependiendo de lo alto o bajo que sea el valor del mismo, por lo que se realiza un desglose de las polaridades positivas y negativas que nos permita observar este hecho.

$$\text{Sentimiento} = \begin{cases} \text{Si } \text{Score} \geq 6; & \text{Excepcional} \\ \text{Si } 1 < \text{Score} < 6; & \text{Muy bueno} \\ \text{Si } 0 < \text{Score} \leq 1; & \text{Bueno} \\ \text{Si } \text{Score} = 0; & \text{Neutral} \\ \text{Si } -1 \leq \text{Score} < 0; & \text{Malo} \\ \text{Si } -6 < \text{Score} < -1; & \text{Muy malo} \\ \text{Si } \text{Score} \leq -6; & \text{Horrible} \end{cases}$$

Para la compañía A, la partición es la que se muestra en el gráfico 3. Vemos como más del 50% de los negativos superan la puntuación -1 en valor absoluto, clasificándose como *Muy malos* u *Horribles*. Por otro lado, solo un pequeño porcentaje de los positivos superan el *Score* 1. Esto parece indicar más intensidad de los negativos.

**Gráfico 3.- Polaridad de sentimiento compañía A.**

*Fuente: Elaboración propia.*

Mediante una media ponderada podemos obtener una calificación del 0 al 5 para cada compañía, siendo 0 horrible y 5 excepcional como figura en el gráfico X y omitiendo los neutrales. Obtenido este valor y mediante una simple regla de tres, se puede obtener en relación de 0 a 10 para facilitar su comprensión inmediata. La fórmula para obtener esta calificación es la siguiente:

*Sea (cantidad en tanto por 1):*

*H = Horrible.*

*MM = Muy malo.*

*M = Malo.*

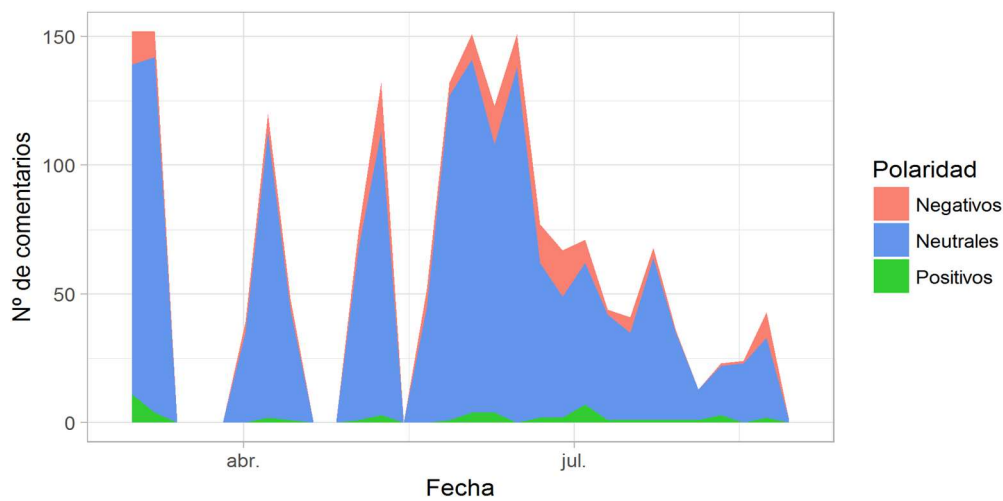
*B = Bueno.*

*MB = Muy bueno.*

*E = Excepcional.*

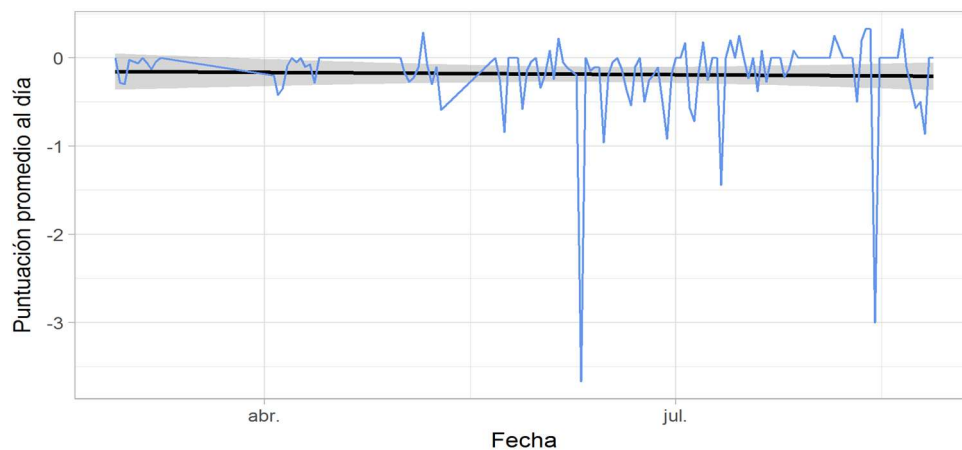
$$Nota = [(H * 0) + (MM * 1) + (M * 2) + (B * 3) + (MB * 4) + (E * 5)] * 2$$

También resulta interesante entender la evolución de la polaridad a lo largo de los meses estudiados, pues esto nos permite conocer en qué momentos del tiempo se producen picos de polaridad, ya sea negativa o positiva. A grandes rasgos, se puede decir que el gráfico 4 mostrará la evolución de la cantidad y la intensidad de los tuits.

**Gráfico 4.- Evolución de la polaridad de sentimiento de la compañía A.**

*Fuente: Elaboración propia.*

En el gráfico 4 podemos notar la evolución del número de comentarios para cada polaridad a lo largo del tiempo, lo que nos posibilita observar en que momentos se producen más tuits positivos o negativos, así como el número de comentarios que se producen. En el caso de la compañía A, se observa un aumento de los tuit negativos a finales de Junio y un pico de positivos a principios de Julio, lo que puede indicar la reacción de los clientes ante una campaña publicitaria en esas fechas o el lanzamiento de un nuevo producto. Por otro lado, vemos como el número de tuits disminuye en los meses vacacionales.

**Gráfico 5.- Evolución de la puntuación media diaria de la compañía A.**

*Fuente: Elaboración propia.*



Por su parte el gráfico 5 nos muestra la evolución del sentimiento desde otra perspectiva. En este caso se muestra la evolución de la variable *Score* media diaria, cuyos picos no tienen por qué coincidir con los del gráfico 4, ya que puede suceder que en unas fechas con muchos comentarios negativos, estos sean de intensidad baja, por lo que no veamos ningún pico en cuanto al *Score* medio. Sin embargo, pueden darse fechas con pocos comentarios negativos con mucha intensidad que nos muestren picos en este gráfico. Para la compañía A vemos 2 picos, uno a finales de Junio, que coincide con el gráfico 4 y otro a finales de Agosto, que puede deberse a lo explicado anteriormente.

A modo de resumen, para cada compañía podemos obtener los siguientes indicadores. Continuando con nuestro ejemplo, tenemos que:

*Sea:*

*NP: Muestra el número de comentarios negativos por cada positivo.*

*Pr: Promedio Score.*

*M: Máximo Score.*

*m: Mínimo Score.*

*Mo: Moda Score.*

*D: Desviación típica Score.*

**Tabla 11.- Tabla resumen compañía A.**

| Compañía A | Nº   | Score | Neg. | Neu.  | Pos.  | NP   | Pr   | M | m   | Mo | D    |
|------------|------|-------|------|-------|-------|------|------|---|-----|----|------|
|            | 1835 | 3,6   | 9,9% | 87,3% | 2,80% | 3,48 | -0,2 | 5 | -31 | 0  | 1,27 |

*Fuente: Elaboración propia.*

La tabla 11 muestra un resumen de los resultados obtenidos, donde también se pueden observar los diferentes estadísticos descriptivos de la variable *Score*.

Los gráficos relativos a las demás compañías se encuentran en el Anexo II debido a que el espacio que los mismos ocuparían dentro del trabajo sería excesivo.

## 6.- Otra información relevante:

### 6.1.- Análisis de emociones:

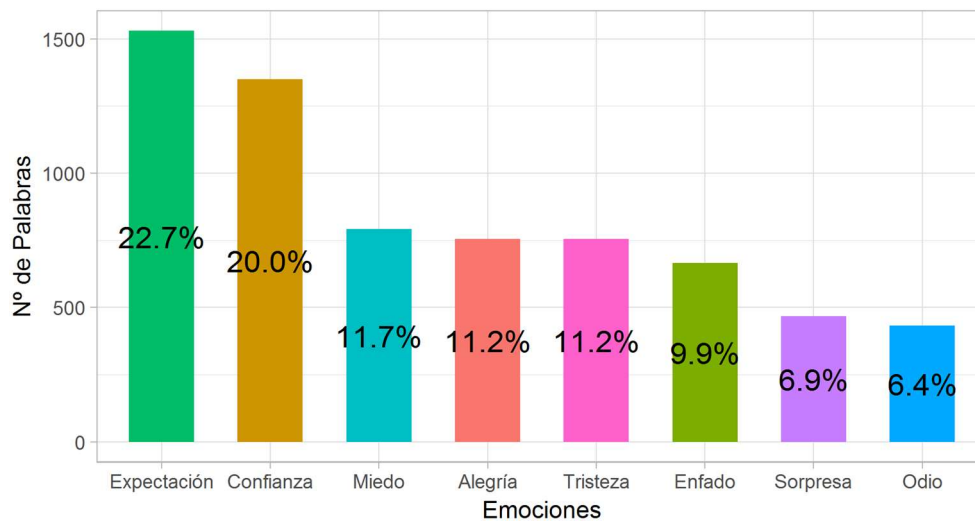
A parte de la polaridad de sentimiento se puede determinar que emoción contienen los tuits. Para el análisis de emociones se ha utilizado la librería de R "syuzhet" (Jockers ML 2015), en concreto la función *get\_nrc\_sentiment()*. Esta librería se utiliza para extraer la emoción del texto mediante la utilización de diversos léxicos de sentimiento que contienen emociones. Los léxicos implementados en las funciones de esta librería son los siguientes:

- "Syuzhet", desarrollado en el Nebraska Literary Lab.
- "AFINN" (Nielsen 2011).
- "Bing" (Hu & Liu 2004).
- "NRC" (Mohammad & Turney 2013)

En nuestro caso, vamos a realizar la clasificación mediante el léxico "NRC", que fue creado por los expertos del Consejo Nacional de Investigación de Canadá. Desarrollado con una amplia gama de aplicaciones, puede utilizarse en multitud de contextos como el análisis de sentimiento, marketing de productos, comportamiento de los consumidores e incluso análisis de campañas políticas. Contiene palabras en inglés, y pueden usarse para analizar textos en inglés, aunque también proporciona traducciones en otros 40 idiomas, incluyendo francés, árabe, chino y español. Cuenta con palabras asociadas a ocho emociones (enfado, miedo, expectación, confianza, sorpresa, tristeza, alegría y odio) anotadas manualmente en Amazon Mechanical.

La función *get\_nrc\_sentiment()* itera sobre un vector de cadenas y devuelve valores de sentimiento y emociones basados en el léxico "NRC". Funciona llamando a este léxico para identificar la presencia de las ocho emociones nombradas anteriormente.

Como vemos en el gráfico 7, para la compañía A destacan los sentimientos de expectación y confianza, mientras que odio, sorpresa y enfado quedan en los últimos lugares, lo que en principio parece positivo.

**Gráfico 6.- Clasificación de emociones compañía A.**

*Fuente: Elaboración propia.*

## 6.2.- Análisis de frecuencias:

Por último, resulta útil conocer que palabras son las más utilizadas por los clientes a la hora de referirse a una determinada compañía, pues esto nos puede indicar también cierto nivel de sentimiento y emoción, así como cuáles son las principales preocupaciones o temas de interés de los mismos.

Una manera muy gráfica de observar las palabras que se repiten más frecuentemente es el de generar una nube de palabras, ya que esta proporciona un rápido resumen visual de la frecuencia de las palabras en un texto.

Antes de realizar la nube de palabras, debemos limpiar el texto con el que vamos a realizarla, para lo que utilizamos la función `tm_map()` de la librería de minería de texto en R “tm” (Ingo Feinerer and Kurt Hornik 2017). Esta función transforma el texto seleccionado en función del argumento que lo acompañe. Estos argumentos eliminan signos de puntuación (`removePunctuation`), convierten cada palabra a minúscula (`content_transformer(tolower)`), suprimen números (`removeNumbers`), espacios en blanco (`stripWhitespace`) y eliminan directamente las palabras que indiquemos (`removeWords`). Centrándonos en `removeWords` nos interesa eliminar las palabras de

uso común, denominadas stopwords, para lo que añadiremos a la función el argumento (`stopwords("spanish")`). Por otra parte es preciso eliminar otras palabras que no aportan nada, como el mismo nombre de la compañía y sus derivaciones (por ejemplo VidaCaixa, Caixa, LaCaixa) o palabras sin sentido (por ejemplo hffhfh, djfj, lodjd). El ejemplo de código en R sería el siguiente:

```
texto <- tm_map(text_corpus, removePunctuation)
texto <- tm_map(texto, content_transformer(tolower))
texto <- tm_map(texto, removeNumbers)
texto <- tm_map(texto, stripWhitespace)
texto <- tm_map(texto, removeWords, stopwords("english"))
texto <- tm_map(texto, removeWords, stopwords("spanish"))
texto <- tm_map(texto, removeWords, c("allianz", "alianzas", "bhd"))
tdm <- TermDocumentMatrix(texto)
m = as.matrix(tdm)
```

Una vez tenemos limpio el texto que contiene todas las palabras y se ha creado la matriz de términos se cuenta el número de veces que aparece cada una en el total del texto mediante la función `sort()` de la librería “base” de R (R Core Team 2017). Por último, creamos una tabla con las frecuencias obtenidas anteriormente para poder representar gráficamente nuestra nube de palabras con la función `wordcloud()` de la librería “wordcloud” (Ian Fellows 2014).

**Figura 1.- Nube de palabras compañía A.**

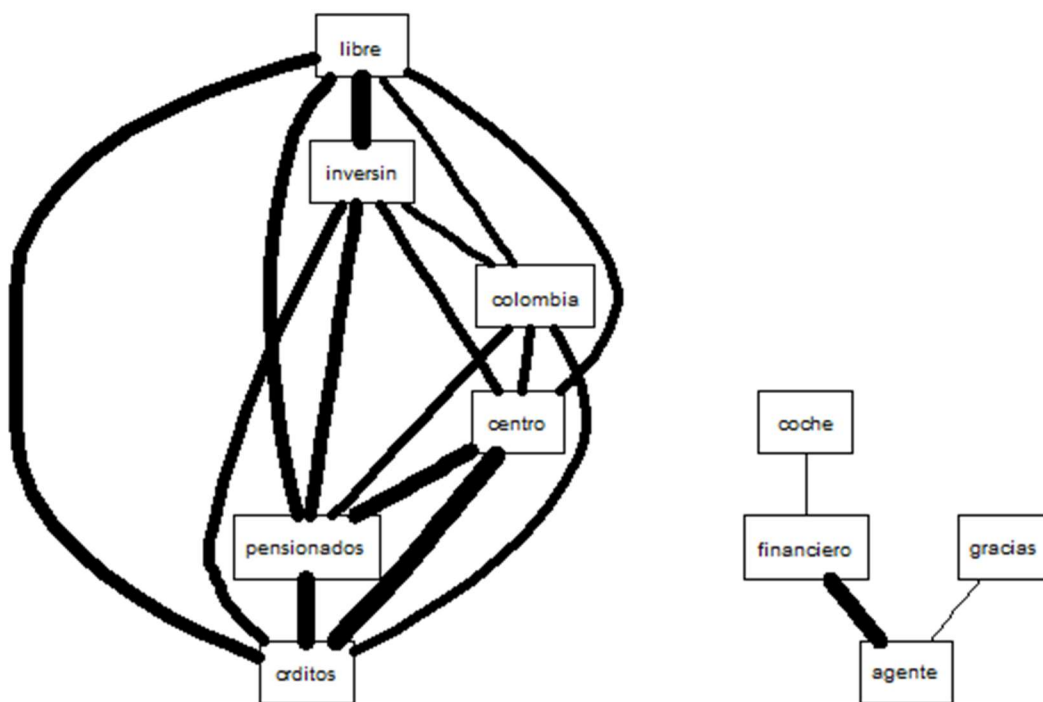


Fuente: Elaboración propia.

Resulta interesante conocer cómo se asocian las principales palabras con las que los clientes se refieren a cada compañía, por lo que se dibuja un diagrama de asociaciones con las 10 más repetidas. Para ello utilizamos el siguiente código en R y obtenemos la figura 2:

```
FiguraX <- plot(tdm, term=df$word, corThreshold=0.01, weighting=T)
```

**Figura 2.- Diagrama de asociaciones compañía A.**



*Fuente: Elaboración propia.*

Cabe decir que cuanto más ancha es la línea que une dos palabras más alta es su correlación. Para la compañía A vemos que las 10 palabras más utilizadas se dividen claramente en dos grupos.

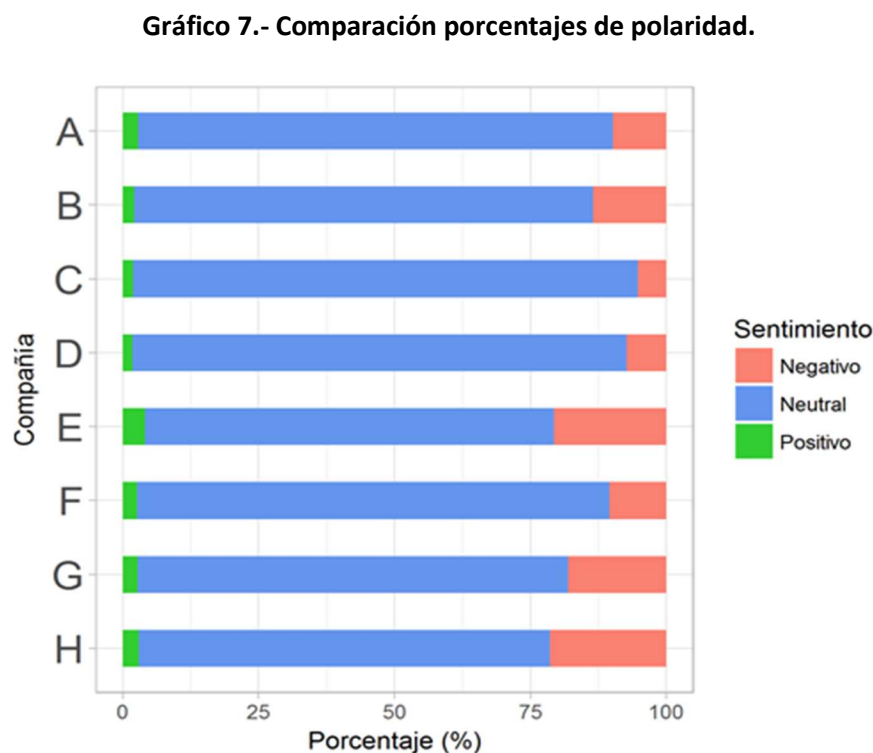
Tanto el gráfico 6 como las figuras 1 y 2 para el resto de compañías se encuentran en el anexo II, al igual que ocurre con el resto de gráficos relacionados con el epígrafe 5.

Por otra parte, cabe decir que el ejemplo de código R utilizado en los epígrafes 5 y 6 se encuentra en el Anexo I.

## 7.- Análisis y comparación de resultados.

En este apartado se van a analizar los resultados obtenidos en el trabajo poniendo el foco en la comparación de polaridad entre las compañías analizadas, ya que un ejemplo de análisis individual es el representado en los epígrafes 5 y 6.

En primer lugar se muestra el gráfico 7, en el que se observa la comparación de los porcentajes de comentarios positivos, negativos y neutros entre las empresas.



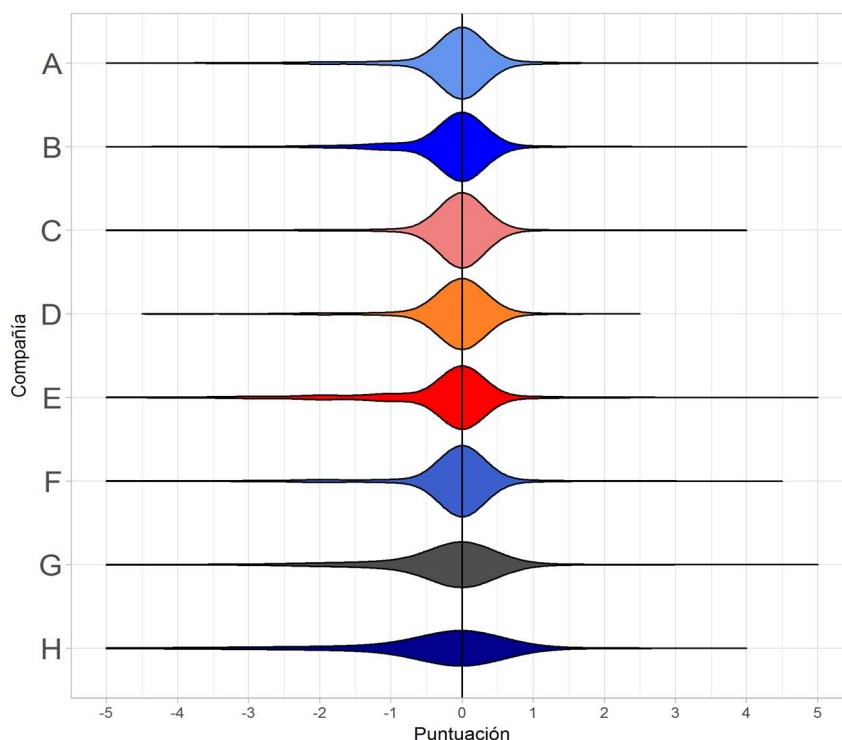
*Fuente: Elaboración propia.*

Queda claro que el porcentaje de neutrales es notablemente mayor que el de positivos y negativos, lo que puede indicar una gran cantidad de publicidad en el sector. La mayoría de estos comentarios no son de clientes finales, sino de mediadores o agencias de publicidad. Notamos que las compañías con mayor porcentaje de comentarios positivos son las compañías A, E, G y H, mientras que las que cuentan con más negativos son las B, E, G y H. El porcentaje tan alto de negativos de las tres últimas hace que los positivos se vean muy neutralizados, mostrando también que en estas tres compañías, a simple vista, se dará el porcentaje menor de comentarios neutrales.

Por otra parte destaca el bajo porcentaje de positivos de las compañías C y D, aunque en el caso de la C también muestra el menor porcentaje de negativos. Los porcentajes exactos se mostrarán más adelante en la tabla 12.

A continuación, el gráfico 8 nos muestra una comparación de las compañías mediante un gráfico de violín, que nos permite observar cómo se distribuye la puntuación para cada compañía.

**Gráfico 8.- Comparación puntuación gráfico de violín.**

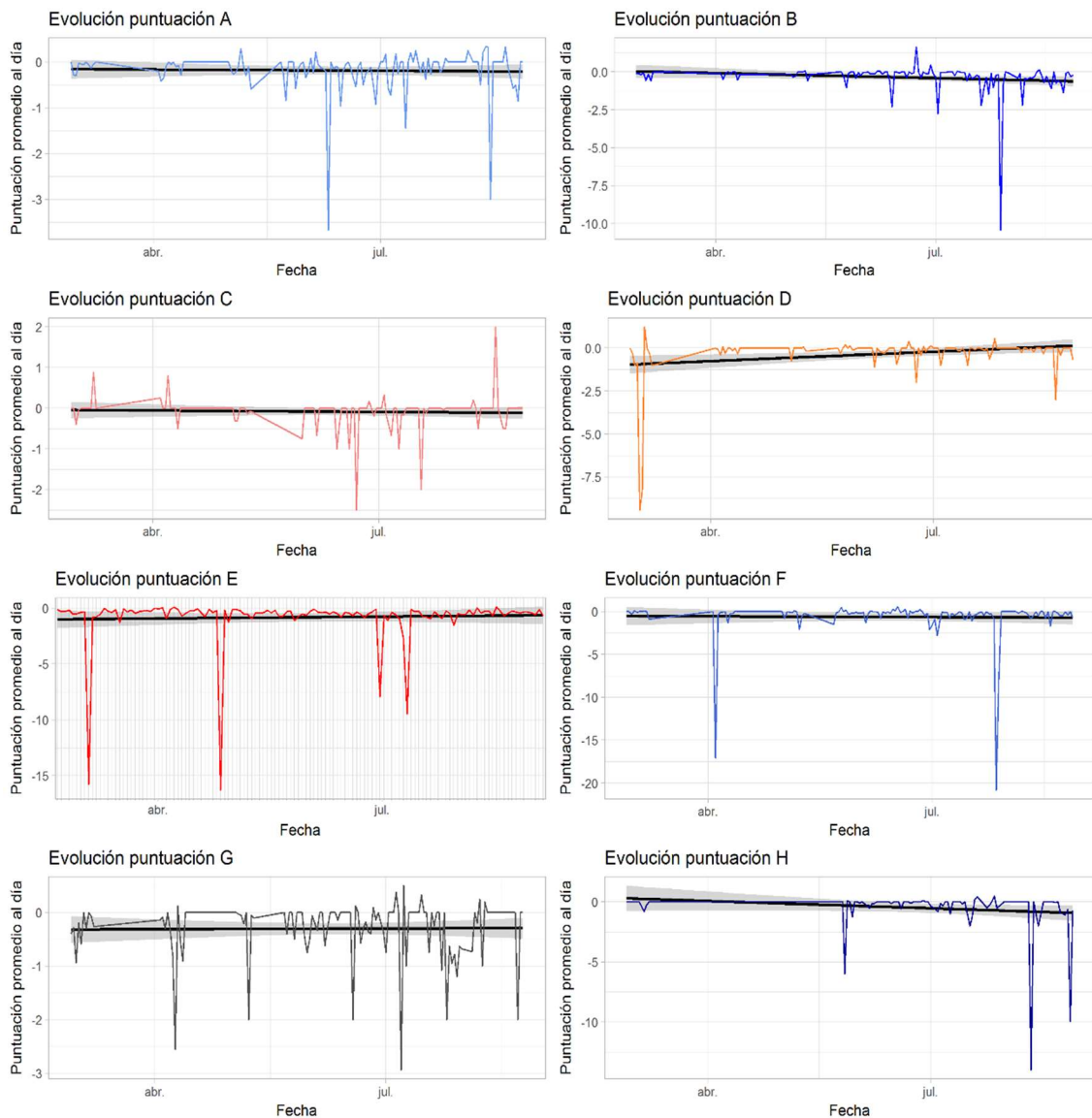


*Fuente: Elaboración propia.*

Como es lógico, debido al alto porcentaje de neutralidad comentado anteriormente, la mayor parte de los comentarios se aglutinan en la puntuación de 0, pero se pueden observar pequeñas diferencias entre las empresas. Se nota como los comentarios de las compañías G y H tienen unas puntuaciones más distribuidas y no tan aglutinadas en la neutralidad. Por otro lado, centrándonos en los comentarios positivos (parte derecha del gráfico), vemos como en la mayoría de las entidades el máximo de puntuación es de menos de 5 puntos, lo que puede indicar que los que hablan bien sobre las empresas no lo hacen con excesivo entusiasmo. En contraposición, la

puntuación en valor absoluto de los comentarios negativos (lado izquierdo del gráfico) supera en todos los casos, menos en el de la compañía D, los 5 puntos. Lo que parece mostrar que cuando se refieren a las compañías en términos negativos lo hacen con más intensidad. También se puede observar como en todos los casos la cantidad de negativos para cada puntuación es mayor que la de los positivos. Por ejemplo, este hecho se puede observar claramente en las compañías E y H, donde existen gran cantidad de comentarios con puntuación -1 y -2. En el gráfico posterior se muestra la evolución de la puntuación media diaria para cada compañía.

**Gráfico 9.- Comparación porcentajes de polaridad.**



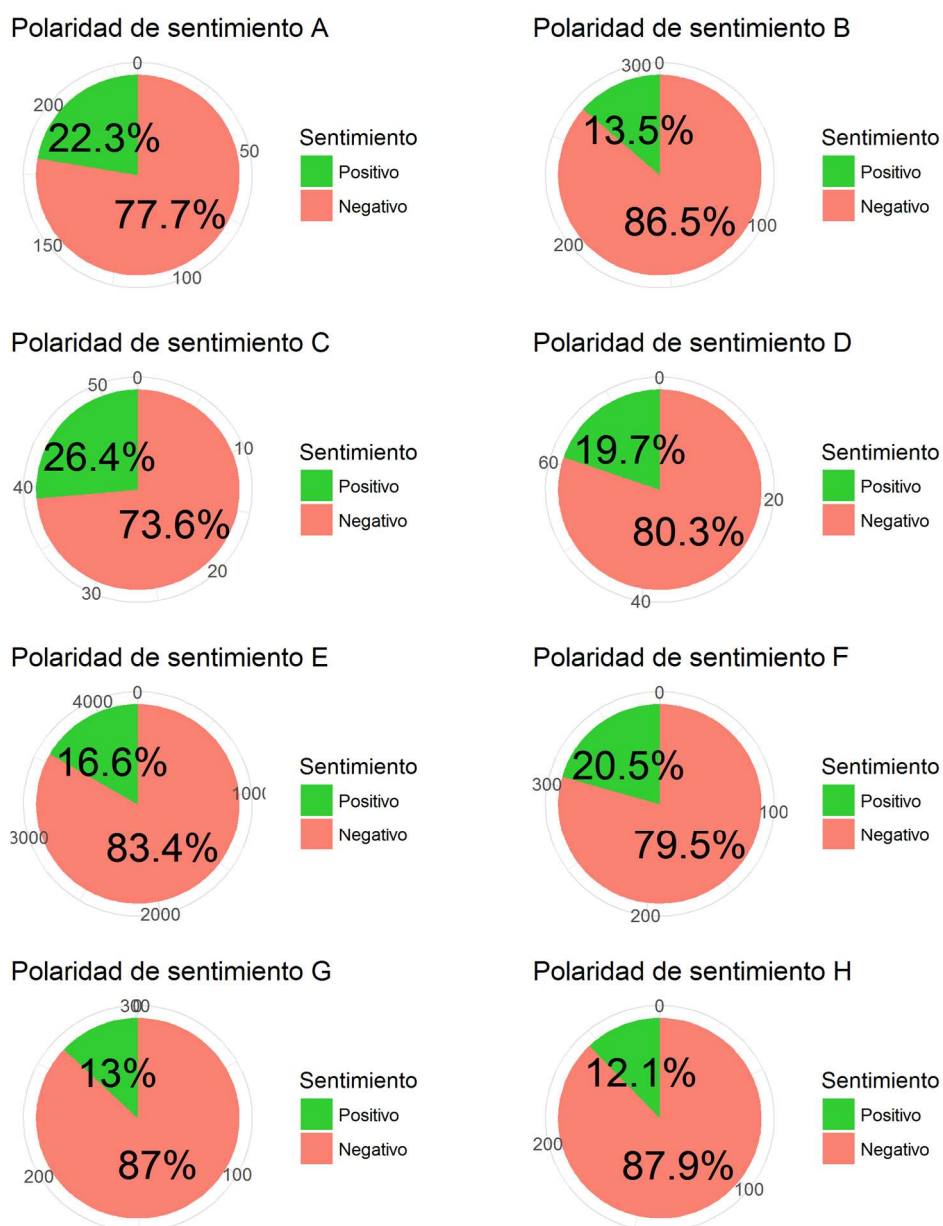
*Fuente: Elaboración propia.*



Se observan picos negativos muy pronunciados en todas las entidades. Mientras que algunas compañías presentan una clara evolución descendente (B y H) de la puntuación media diaria, las demás se mantienen prácticamente constantes, excepto la D que muestra una fuerte evolución ascendente.

Ahora, vamos a observar de forma clara, para cada empresa, la distribución entre positivos y negativos, sin incluir los neutrales.

**Gráfico 10.- Comparación porcentajes de polaridad.**



*Fuente: Elaboración propia.*

Como parecía indicar el gráfico 7, las empresas con mayor número de comentarios positivos respecto al total (sin contar neutrales), son la C y la A, mientras que la B, G y H muestran los peores resultados en cuanto a polaridad se refiere. Cabe recalcar obviamente, que en todas las compañías el porcentaje de negativos es mucho más alto que el de positivos, poniendo de manifiesto que los clientes están más descontentos que contentos con las PCSAE, o bien, muestran más dicho descontento en Twitter que su bienestar con las mismas. Esto puede ocurrir porque los clientes muestran su disconformidad cuando la aseguradora no cumple con sus expectativas, mientras que los clientes satisfechos no se expresan al pensar que la aseguradora únicamente ha cumplido con su parte del contrato.

Para terminar con el análisis y comparación se muestra una tabla que resume todos los indicadores obtenidos para todas las compañías.

**Tabla 12.- Tabla resumen resultados PCSAE.**

| Comp. / Indicador | Nº    | Score      | Neg%        | Neu% | Pos%        | NP          | Pr   | M    | m      | Mo | D     |
|-------------------|-------|------------|-------------|------|-------------|-------------|------|------|--------|----|-------|
| <b>A</b>          | 1835  | <b>3,6</b> | 9,9         | 87,3 | <b>2,80</b> | <b>3,48</b> | -0,2 | 5    | -31    | 0  | 1,27  |
| <b>B</b>          | 1975  | 3,5        | 13,6        | 84,3 | 2,1         | 6,38        | -0,2 | 5    | -31    | 0  | 1,27  |
| <b>C</b>          | 731   | <b>4,1</b> | 5,3         | 92,7 | 1,9         | <b>2,79</b> | -0,1 | 10,5 | -12    | 0  | 0,86  |
| <b>D</b>          | 835   | 3,5        | 7,3         | 90,9 | 1,8         | 4,07        | -0,2 | 2,5  | -56,5  | 0  | 2,12  |
| <b>E</b>          | 17214 | 3,5        | <b>20,7</b> | 75,2 | <b>4,1</b>  | 5,01        | -0,8 | 53   | -2158  | 0  | 24,56 |
| <b>F</b>          | 2935  | 3,3        | 10,4        | 86,9 | 2,7         | 3,87        | -0,6 | 7,5  | -607,5 | 0  | 12,25 |
| <b>G</b>          | 1447  | 3,3        | <b>18,1</b> | 79,2 | 2,7         | 6,72        | -0,4 | 8,5  | -44,5  | 0  | 1,96  |
| <b>H</b>          | 1145  | 3,1        | <b>21,5</b> | 75,5 | <b>3</b>    | 7,24        | -1,1 | 14,5 | -664   | 0  | 19,8  |

*Fuente: Elaboración propia.*

En definitiva, y atendiendo al *Score* ya explicado en el epígrafe 5.1, no se observan diferencias excesivamente significativas entre las entidades. Aunque cabe destacar el buen resultado de la compañía C y el malo de la compañía H en comparación a las demás.

## 8.- Conclusiones.

En este trabajo se ha conseguido obtener la opinión de los clientes sobre las PCSAE utilizando una fuente de información gratuita (Twitter). Este método para obtener información sobre que piensan los clientes sobre las empresas puede ser muy atractivo para las mismas, en contraposición a otras técnicas como las encuestas de opinión, que tienen un mayor coste y en las que existirá más subjetividad. En Twitter, los clientes comparten su opinión libremente, sin conocer a priori que esta información puede ser utilizada por las empresas. Esto supone que sus comentarios, a través de los tuits, sean mucho más objetivos y expresen sin ningún tipo de predisposición sus pensamientos.

El léxico específico del sector asegurador (Lexiseg) que hemos obtenido presenta unos resultados de exactitud muy buenos (accuracy = 92%) y superiores con holgura a todos los diccionarios ya existentes en castellano.

En cuanto al análisis y comparación de resultados, las principales conclusiones que obtenemos tienen que ver con el gran número de tuits neutrales, que en su mayoría se deben a publicidad. Los porcentajes negativos y positivos, así como la nota obtenida, no difieren notablemente entre las compañías, lo que puede indicar que la opinión de los clientes sobre el sector asegurador es bastante homogénea. La evolución de la polaridad se mantiene constante en la mayoría de las empresas, aunque son constantes los picos negativos en determinados días en todos los casos.

Este análisis de sentimiento sobre las PCSAE puede ser la base para continuar con otras líneas de investigación. En concreto, la automatización del proceso de captación y análisis de tuits a tiempo real sería de gran interés, pues permitiría una mejor toma de decisiones corporativas de forma rápida, así como reaccionar inmediatamente ante la reacción de los clientes a campañas publicitarias.

Por otro lado, aunque el Lexiseg obtenga unos datos de exactitud muy destacados, sería interesante utilizarlo como base para automatizar la alimentación y actualización

del mismo. Es decir, el léxico creado en este trabajo, se ha adaptado de forma manual al sector partiendo de otros dos léxicos ya existentes. La idea es la de seguir alimentándolo de palabras mediante técnicas de aprendizaje automático a medida que se vayan añadiendo nuevos tuits a la base de datos de estudio.

Otra línea sería la inclusión de otras redes sociales como podría ser Facebook, o blogs especializados.

En definitiva, mediante la utilización de un programa estadístico (R), una cuenta en la red social Twitter, la librería “twitteR” principalmente y unos léxicos de opinión, hemos conseguido obtener la opinión de miles de personas sobre las PCSAE.

## 9.- Bibliografía.

- Barbosa Santillán, L. I. (2015). Hacia un lexicón unificado de sentimientos basado en unidades de procesamiento gráfico. Universidad Politécnica de Madrid.
- Breen, J. (2011). R by example: mining Twitter for consumer attitudes towards airlines. Boston Predictive Analytics MeetUp, (Junio), 39.
- Cruz, Fermín L., José A. Troyano, Beatriz Pontes, F. Javier Ortega (2014). Building layered, multilingual sentiment lexicons at synset and lemma levels, Expert Systems with Applications.
- Daming, X., Xiaomei, W., & Wei, L. (2008). Social network analysis. CRC Press, 16, 412. [https://doi.org/10.1007/978-1-4614-1599-2\\_26](https://doi.org/10.1007/978-1-4614-1599-2_26)
- Diaz Rangel, I. (2013). Detección de afectividad en texto en español basada en el contexto lingüístico para síntesis de voz. Instituto Politécnico Nacional.
- Dubiau, L. (2013). Procesamiento de Lenguaje Natural en Sistemas de Análisis de Sentimientos. Universidad de Buenos Aires.
- Feinerer, I. (2015). Introduction to the tm Package: Text Mining in R, 8. <https://doi.org/10.1201/9781420068740>
- Fernández, J., Boldrini, E., Gómez, J. M., & Martínez-Barco, P. (2011). Análisis de sentimientos y minería de opiniones: El corpus EmotiBlog. Procesamiento Del Lenguaje Natural, 47, 9.
- Fernández, J. M. (2016). Análisis de contenidos en Social Media: Clasificación de mensajes e identificación de influyentes en el Banco Central Europeo (BCE). Universidad Complutense de Madrid.
- Henriquez, C., Guzmán, J., & Salcedo, D. (2016). Minería de Opiniones basado en la adaptación al español de ANEW sobre opiniones acerca de hoteles. Procesamiento Del Lenguaje Natural, 56, 25–32.
- Hernández Petlachi, R., & Li, X. (2014). Análisis de sentimiento sobre textos en Español basado en aproximaciones semánticas con reglas lingüísticas. Tass 2014, 7.
- Lage Garcia, L. (2014). Herramienta para el análisis de la opinión en tweets periodísticos. Universidad Pompeu Fabra.
- Mir Montserrat, D. (2015). Analítica de datos en Twitter. Universitat Autònoma de Barcelona (UAB).

- Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3), 436–465. <https://doi.org/10.1111/j.1467-8640.2012.00460.x>
- Mohammad, S. M., & Turney, P. D. (2010). Emotions evoked by common words and phrases: using mechanical turk to create an emotion lexicon. CAAGET '10 Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, (Junio), 26–34. Retrieved from <http://dl.acm.org/citation.cfm?id=1860631.1860635>
- Moreno Sandoval, A. (2014). Análisis de opinión y contenido en los medios sociales. Instituto de Ingeniería Del Conocimiento, 11.
- Ortiz, A. M., Pozo, A. P., & Sanchez, S. T. (2010). Sentitext: sistema de analisis de sentimiento para el espanol. *Procesamiento Del Lenguaje Natural*, 45(45), 297–298.
- Oyarzún Delgado, C. A. (2014). Análisis automático de sentimientos sobre opiniones y/o comentarios de novelas en español. Universidad del Bío-Bío.
- Pérez Vera, S. A. (2017). Análisis y Clasificación de Textos con Técnicas Semi Supervisadas Aplicado a Área Atención al Cliente. Universidad Católica de Valparaíso.
- Pérez-Rosas, V., Banea, C., & Mihalcea, R. (2012). Learning Sentiment Lexicons in Spanish. Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC), 3077-3081. Istanbul, Turkey. <https://doi.org/10.1.1.383.5959>
- Rangel, I. D., Sidorov, G., & Guerra, S. S. (2014). Creación y evaluación de un diccionario marcado con emociones y ponderado para el español. *Onomazein*, 29(1), 31-46. <https://doi.org/10.7764/onomazein.29.5>
- Rincón García, S. (2016). Minería de textos y análisis de sentimientos en sanidadysalud.com. Universidad Complutense de Madrid.
- Rodríguez Aldape, F. M. (2013). Cuantificación del Interés de un usuario en un tema mediante minería de texto y análisis de sentimiento. Universidad Autónoma de Nuevo León.
- Rodríguez, J. M. (2014). Procesamiento de Lenguaje Natural en Sistemas de Análisis de Sentimientos, 1–7.
- Salas-zárate, M. P., Rodríguez-García, M. Á., & Almela, Á. (2011). Estudio de las categorías LIWC para el análisis de sentimientos en español Introducción, 2–5.

Saralegi, X., & San Vicente, I. (2013). Elhuyar at TASS 2013. XXIX Congreso de La Sociedad Española de Procesamiento de Lenguaje Natural, 143–150. Madrid, Spain. Retrieved from <http://www.sepln.org/workshops/tass/2013/papers/tass2013-submission3-Elhuyar.pdf>. ISBN: 978-84-695-8349-4

Troyano, A., Pontes, B., & Ortega, F. J. (2014). ML-SentiCon : Un lexicón multilingüe de polaridades semánticas a nivel de lemas, 113–120.

Zhao, Y. (2015). R and Data Mining: Examples and Case Studies, (Octubre), 1–160. <https://doi.org/10.1016/B978-0-12-396963-7.00001-5>

Zhao, Y. (2015). Introduction to Data Mining with R, (Mayo), 46.

Zhao, Y. (2015). Text Mining with R – Twitter Data Analysis, (Mayo), 34.

#### Librerías de R:

Baptiste Augue (2016). gridExtra: Miscellaneous Functions for "Grid" Graphics. R package version 2.2.1. <https://CRAN.R-project.org/package=gridExtra>

Dirk Eddebuettel and Romain Francois (2011). Rcpp: Seamless R and C++ Integration. Journal of Statistical Software, 40(8), 1-18. URL <http://www.jstatsoft.org/v40/i08/>.

Erich Neuwirth (2014). RColorBrewer: ColorBrewer Palettes. R package version 1.1-2. <https://CRAN.R-project.org/package=RColorBrewer>

Grün B and Hornik K (2011). "topicmodels: An R Package for Fitting Topic Models." *Journal of Statistical Software*, 40(13), pp. 1-30. doi: 10.18637/jss.v040.i13 (URL: <http://doi.org/10.18637/jss.v040.i13>).

Hadley Wickham (2011). The Split-Apply-Combine Strategy for Data Analysis. Journal of Statistical Software, 40(1), 1-29. URL <http://www.jstatsoft.org/v40/i01/>.

Hadley Wickham (2007). Reshaping data with the reshape package. Journal of Statistical Software, 21(12).

Hadley Wickham (2009). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York.

Hadley Wickham (2016). scales: Scale Functions for Visualization. R package version 0.4.1. <https://CRAN.R-project.org/package=scales>

- Hadley Wickham (2017). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.2.0. <https://CRAN.R-project.org/package=stringr>
- Ian Fellows (2014). wordcloud: Word Clouds. R package version 2.5. <https://CRAN.R-project.org/package=wordcloud>
- Ingo Feinerer and Kurt Hornik (2017). tm: Text Mining Package. R package version 0.7-1. <https://CRAN.R-project.org/package=tm>
- Jeff Gentry and Duncan Temple Lang (2015). ROAuth: R Interface For OAuth. R package version 0.9.6. <https://CRAN.R-project.org/package=ROAuth>
- Jeff Gentry (2015). twitterR: R Based Twitter Client. R package version 1.1.9. <https://CRAN.R-project.org/package=twitterR>
- Jens Oehlschlägel (2017). bit64: A S3 Class for Vectors of 64bit Integers. R package version 0.9-7. <https://CRAN.R-project.org/package=bit64>
- Jockers ML (2015). \_Syuzhet: Extract Sentiment and Plot Arcs from Text\_. <URL: <https://github.com/mjockers/syuzhet>>.
- Kirill Müller and Hadley Wickham (2017). tibble: Simple Data Frames. R package version 1.3.3. <https://CRAN.R-project.org/package=tibble>
- Matt Dowle and Arun Srinivasan (2017). data.table: Extension of `data.frame`. R package version 1.10.4. <https://CRAN.R-project.org/package=data.table>
- P. PONCET (2012). modeest: Mode Estimation. R package version 2.1. <https://CRAN.R-project.org/package=modeest>
- R. Gentleman, Elizabeth Whalen, W. Huber and S. Falcon (2017). graph: graph: A package to handle graph data structures. R package version 1.54.0.



## Anexo I.

```
#####
/// TFM: ANÁLISIS DE SENTIMIENTO ///
/// /////////////////////////////////////////
/// Francisco José Mtnez Mtnez ///
#####
library(ROAuth)
library(syuzhet)
library(bit64)
library(twitterR)
library(Rcpp)
library(plyR)
library(stringr)
library(reshape)
library(tibble)
library(scales)
library(ggplot2)
library(RColorBrewer)
library(tm)
library(modeest)
library(wordcloud)
library(topicmodels)
library(data.table)
library(gridExtra)
#source('http://bioconductor.org/biocLite.R')
#biocLite('Rgraphviz')
library(Rgraphviz)
library(graph)
source("score_sentiment.R")
pos.words <- scan('positivasV6.txt', what='character', comment.char=';')
neg.words <- scan('negativasV6.txt', what='character', comment.char=';')

#####
#### COMPañÍA A ####
#####
#### A.- Carga y preparación: ####

datossent <- read.csv("compA.csv")

datosent <- datosent[!datosent$screenName=="AXA",]
datosent <- datosent[!datosent$screenName=="AXAContigo",]
datosent <- datosent[!datosent$screenName=="AXAExclusiv",]
datosent <- datosent[!datosent$screenName=="FundacionAXA",]
datosent <- datosent[!datosent$screenName=="AXAResponde",]
datosent <- datosent[!datosent$screenName=="JAT_AXA",]

datosent <- datosent[!datosent$screenName=="catalanaocc",]
datosent <- datosent[!datosent$screenName=="SegPlusUltra",]
datosent <- datosent[!datosent$screenName=="CatalanaOcciAlm",]
datosent <- datosent[!datosent$screenName=="SegPlusGisbert",]

datosent <- datosent[!datosent$screenName=="GeneraliLiher",]
datosent <- datosent[!datosent$screenName=="generalimairena",]

names <- data.frame(table(datosent$screenName))
names <- names[order(-names$Freq),]

datosent <- datosent[!duplicated(datosent$text),]
datosent$text <- sapply(datosent$text,function(row) iconv(row, "latin1",
"ASCII", sub=""))
```

```

datosent$text = gsub("(f|ht)tp(s?)://(.*)[.][a-z]+", "", datosent$text)

text <- datosent$text

#### B.- Limpieza y sentimiento: ####

result = score.sentiment(text, pos.words, neg.words)

# Crear tres diferentes data frames por Score, Positivo and Negativo:
test1 <- result[[1]]
test2 <- result[[2]]
test3 <- result[[3]]

# Eliminar columnas de texto:
test1$text=NULL
test2$text=NULL
test3$text=NULL

# Almacenar la primera fila (contiene el Score de sentimiento):
q1 <- test1[1,]
q2 <- test2[1,]
q3 <- test3[1,]
qq1 <- melt(q1, ,var='Score')
qq2 <- melt(q2, ,var='Positivo')
qq3 <- melt(q3, ,var='Negativo')
qq1['Score'] = NULL
qq2['Positivo'] = NULL
qq3['Negativo'] = NULL

# Crear un data frame:
table1 <- data.frame(Text=result[[1]]$text, Score=qq1)
table2 <- data.frame(Text=result[[2]]$text, Score=qq2)
table3 <- data.frame(Text=result[[3]]$text, Score=qq3)

# Fusionar los tres data frames en uno:
tablasent <- data.frame(Text=table1$Text, Score=table1$value,
Positivo=table2$value, Negativo=table3$value)
tablasent$Retweet <- datosent$retweetCount
tablasent$Fav <- datosent$favoriteCount
tablasent$Fecha <- datosent$created
tablasent$Name <- datosent$screenName

# Añadir bonus/malus por retweet y/o favorito:
for(i in 1:nrow(tablasent)) {
  if (tablasent$Retweet[i] == 0)
    tablasent$Score[i] <- tablasent$Score[i] else
    if (tablasent$Retweet[i] > 0 & tablasent$Score[i] > 0)
      tablasent$Score[i] <- tablasent$Score[i]+(tablasent$Retweet[i]*0.5)
else
  if (tablasent$Retweet[i] > 0 & tablasent$Score[i] < 0)
    tablasent$Score[i] <- tablasent$Score[i]-
(tablasent$Retweet[i]*0.5)}

for(i in 1:nrow(tablasent)) {
  if (tablasent$Fav[i] == 0)
    tablasent$Score[i] <- tablasent$Score[i] else
    if (tablasent$Fav[i] > 0 & tablasent$Score[i] > 0)
      tablasent$Score[i] <- tablasent$Score[i]+(tablasent$Fav[i]) else
    if (tablasent$Fav[i] > 0 & tablasent$Score[i] < 0)
      tablasent$Score[i] <- tablasent$Score[i]-(tablasent$Fav[i])}

sent1 <- qplot(data=tablasent, factor(Positivo), geom="bar",
  main="N° de palabras positivas por comentario A",
  ylab="N° comentarios",xlab="N° de palabras positivas",
  fill=factor(Positivo)) + theme_light() +
  scale_fill_hue(name="N° +") +

```

```

  theme(legend.position="none")
sent1

sent2 <- qplot(data=tablasent, factor(Negativo),
              geom="bar", main="N° de palabras negativas por comentario A",
              ylab="N° comentarios",xlab="N° de palabras negativas",
              fill=factor(Negativo)) + theme_light() +
  scale_fill_hue(name="N° -") + theme(legend.position="none")
sent2

sent3 <- qplot(data=tablasent, factor(Score),
              geom="bar", main="N° de puntos por comentario A",
              ylab="N° comentarios",xlab="Resultado",
              fill=factor(Score)) + theme_light() +
  scale_fill_hue(name="Puntos") + theme(legend.position="none")
sent3
COMP4 <- tablasent

#### C.- Gráficos de sentimiento: ####

Sc = tablasent$Score

# Positivo:
positivo <- sapply(Sc, function(Sc) Sc > 0)
list_positivo = Sc[positivo]
value_positivo = length(list_positivo)

# Negativo:
negativo <- sapply(Sc, function(Sc) Sc < 0)
list_negativo = Sc[negativo]
value_negativo = length(list_negativo)

# Neutral:
neutral <- sapply(Sc, function(Sc) Sc == 0)
list_neutral = Sc[neutral]
value_neutral = length(list_neutral)

slices1 <- c(value_negativo,value_neutral, value_positivo )
Sentimiento <- c("Negativo","Neutral", "Positivo")
result1 <- data.frame(Sentimiento)
result1$slices1 <- slices1
nivcol <- c("salmon","cornflowerblue","limegreen")

tabla01A <- cbind(result1,nivcol)
tabla01A1 <- tabla01A
tabla01A <- tabla01A[order(slices1),]
tabla01A$Sentimiento <- with(tabla01A, reorder(Sentimiento, slices1))
z <- Corpus(VectorSource(tabla01A$nivcol))
z <- z$content

sent4 <- ggplot(tabla01A, aes(x=Sentimiento, y=slices1, fill=Sentimiento)) +
  geom_bar(stat="identity",width = 1.1) +
  scale_fill_manual(values=z, name="Sentimiento") + theme_light() +
  theme(axis.title.y = element_text(angle = 0)) +
  coord_polar(theta="x") + aes(x=reorder(Sentimiento, slices1)) +
  theme(axis.text.x = element_text(angle = 0)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
        panel.border = element_blank(), axis.ticks = element_blank()) +
  ggtitle("Polaridad de sentimiento A") +
  geom_text(aes(y = max(slices1)*0.5,
                label = percent(slices1/sum(slices1))), size=5)

sent4
CA4 <- sent4

# Bueno:
bueno <- sapply(Sc, function(Sc) Sc <= 1 && Sc > 0)

```

```

list_bueno = Sc[bueno]
value_bueno = length(list_bueno)

# Muy bueno:
muybueno <- sapply(Sc, function(Sc) Sc > 1 && Sc < 6)
list_muybueno = Sc[muybueno]
value_muybueno = length(list_muybueno)

# Excepcional:
excepcional <- sapply(Sc, function(Sc) Sc >= 6)
list_excepcional = Sc[excepcional]
value_excepcional = length(list_excepcional)

# Malo:
malo <- sapply(Sc, function(Sc) Sc >= -1 && Sc < 0)
list_malo = Sc[malo]
value_malo = length(list_malo)

# Muy malo:
muymalo <- sapply(Sc, function(Sc) Sc < -1 && Sc > -6)
list_muymalo = Sc[muymalo]
value_muymalo = length(list_muymalo)

# Horrible:
horrible <- sapply(Sc, function(Sc) Sc <= -6)
list_horrible = Sc[horrible]
value_horrible = length(list_horrible)

# Neutral:
neutral <- sapply(Sc, function(Sc) Sc == 0)
list_neutral = Sc[neutral]
value_neutral = length(list_neutral)

slices2 <- c(value_excepcional, value_muybueno, value_bueno, value_neutral,
value_malo, value_muymalo, value_horrible)
Sentimiento <- c("6.Excepcional", "5.Muy bueno", "4.Bueno", "3.Neutral",
"2.Malo", "1.Muy malo", "0.Horrible")
result2 <- data.frame(Sentimiento)
result2$slices2 <- slices2

nivcol <- c("brown3", "brown1", "salmon", "cornflowerblue",
"limegreen", "forestgreen", "darkgreen")

tabla02 <- cbind(result2, nivcol)
z <- Corpus(VectorSource(tabla02$nivcol))
z <- z$content

sent5 <- ggplot(result2, aes(x=Sentimiento, y=slices2, fill=Sentimiento)) +
  geom_bar(stat = "identity") + theme_light() +
  scale_fill_manual(values=z, name="Sentimiento") +
  ggtitle("Clasificación de sentimientos A") +
  xlab("Sentimiento") + ylab("N° comentarios") +
  geom_text(aes(y = slices2/2, label = percent(slices2/sum(slices2))),
  size=5) + theme(legend.position="none")
sent5

sent6 <- ggplot(result2, aes(x=Sentimiento, y=slices2, fill=Sentimiento)) +
  geom_bar(stat="identity", width = 1.1) + theme_light() +
  scale_fill_manual(values=z, name="Sentimiento") +
  theme(axis.title.y = element_text(angle = 0)) +
  coord_polar() + aes(x=reorder(Sentimiento, slices2)) +
  theme(axis.text.x = element_text(angle = 0)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
  panel.border = element_blank(), axis.ticks = element_blank()) +
  ggtitle("Clasificación de sentimientos A") +
  geom_text(aes(y = max(slices2)*0.5,

```

```

      label = percent(slices2/sum(slices2)),
      size=5)
sent6

CA6 <- sent6

x <- data.frame(matrix(c(0:2), nrow = 3, ncol = 1))

s <- sum(result2$slices2[!result2$Sentimiento=="3.Neutral"])
a <- result2$slices2[result2=="6.Excepcional"]/s
b <- result2$slices2[result2=="5.Muy bueno"]/s
c <- result2$slices2[result2=="4.Bueno"]/s
#d <- result2$slices2[result2=="3.Neutral"]/sum(result2$slices2)
e <- result2$slices2[result2=="2.Malo"]/s
f <- result2$slices2[result2=="1.Muy malo"]/s
g <- result2$slices2[result2=="0.Horrible"]/s

h <- matrix(c(5:0), ncol=6)
i <- c(a,b,c,e,f,g)

nota6 <- sum(h*i)

nota_sent <- round(nota6*10/5, digits = 1)

#### C.- Gráficos de sentimiento (sin neutral): ####

polaridad <- tabla01A[!tabla01A$Sentimiento=="Neutral",]
nivcol <- c("limegreen","salmon")

sent13 <- ggplot(polaridad, aes(x="", y=slices1, fill=Sentimiento)) +
  geom_bar(stat="identity",width = 1.1) +
  geom_text(size=7, aes(y = tabla01A$slices1[2]/2,
    label =
percent(tabla01A$slices1[2]/(tabla01A$slices1[1]+tabla01A$slices1[2]))) +
  geom_text(size=7, aes(y = (tabla01A$slices1[1]/2) + tabla01A$slices1[2],
    label =
percent(tabla01A$slices1[1]/(tabla01A$slices1[1]+tabla01A$slices1[2]))) +
  coord_polar(theta = "y") + theme_light() +
  scale_fill_manual(values=nivcol, name="Sentimiento") +
  theme(axis.title.y = element_text(angle = 0)) +
  theme(axis.text.x = element_text(angle = 0)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
    panel.border = element_blank(), axis.ticks = element_blank()) +
  ggtitle("Polaridad de sentimiento A")
sent13

CA13 <- sent13

#### D.- Emociones: ####

df <- do.call("rbind", lapply(datosent, as.data.frame))

mySentiment <- get_nrc_sentiment(datosent$text)

df <- cbind(df, mySentiment)

sentimentTotals <- data.frame(colSums(df[,c(2:9)]))
names(sentimentTotals) <- "count"
sentimentTotals <- cbind("Emoción" = rownames(sentimentTotals),
sentimentTotals)
rownames(sentimentTotals) <- NULL
sentimentTotals$`Emoción` = gsub("anger", "Enfado", sentimentTotals$`Emoción`)
sentimentTotals$`Emoción` = gsub("anticipation", "Expectación",
sentimentTotals$`Emoción`)
sentimentTotals$`Emoción` = gsub("disgust", "Odio", sentimentTotals$`Emoción`)
sentimentTotals$`Emoción` = gsub("fear", "Miedo", sentimentTotals$`Emoción`)

```

```

sentimentTotals$`Emoción` = gsub("joy", "Alegría", sentimentTotals$`Emoción`)
sentimentTotals$`Emoción` = gsub("sadness", "Tristeza",
sentimentTotals$`Emoción`)
sentimentTotals$`Emoción` = gsub("surprise", "Sorpresa",
sentimentTotals$`Emoción`)
sentimentTotals$`Emoción` = gsub("trust", "Confianza",
sentimentTotals$`Emoción`)

sum(sentimentTotals$count)
sent7 <- ggplot(sentimentTotals, aes(x = `Emoción`, y = count, width=0.6)) +
  geom_bar(aes(fill = `Emoción`), stat = "identity") + theme_light() +
  theme(legend.position = "none") +
  aes(x=reorder(`Emoción`, -count)) +
  xlab("Emociones") + ylab("N° de Palabras") +
  ggtitle("Clasificación de emociones A") +
  geom_text(aes(y = count/2,
                label = percent(count/sum(count))), size=5)
sent7

#### E.- Nube de palabras: ####

text_corpus <- Corpus(VectorSource(text))

# Limpiar texto:
text_clean <- tm_map(text_corpus, removePunctuation)
text_clean <- tm_map(text_clean, content_transformer(tolower))
text_clean <- tm_map(text_clean, removeWords, stopwords("english"))
text_clean <- tm_map(text_clean, removeWords, stopwords("spanish"))
text_clean <- tm_map(text_clean, removeNumbers)
text_clean <- tm_map(text_clean, stripWhitespace)
text_clean <- tm_map(text_clean,
                    removeWords,
                    c("seguros", "seguro", "seguroel", "allianzseguros", "santaflowfenix", "horas",
"paulosp", "sos", "omztxglh"))

# Crea una matriz de términos
tdm <- TermDocumentMatrix(text_clean)

# Convierte a una matriz
m = as.matrix(tdm)

# Conteo de palabras en orden decreciente
wf <- sort(rowSums(m), decreasing=TRUE)
wf[1]
# Crea un data frame con las palabras y sus frecuencias
dm <- data.frame(word = names(wf), freq=wf)
nm <- cbind(nota_sent, max(dm$freq*1.1))
nm <- data.frame(nm)
colnames(nm) <- c("word", "freq")
dm <- rbind(nm, dm)
df <- dm[2:11,]

# Gráfico de frecuencias:
sent8 <- ggplot(df, aes(x=word, y=freq, width=0.4)) +
  geom_bar(stat="identity") +
  aes(x=reorder(word, freq)) + theme_light() +
  xlab("Palabras") + ylab("Frecuencia") + coord_flip() +
  theme(axis.text=element_text(size=7)) +
  ggtitle("Palabras más frecuentes A")
sent8

# Nube de palabras:
sent9 <- wordcloud(dm$word, dm$freq,
                  random.order=FALSE, colors=brewer.pal(4, "RdYlBu"),
                  scale=c(4,0.5), max.words=150)

```

```
#### F.- Diagrama de asociaciones: ####

sent10 <- plot(tdm, term = df$word, corThreshold = 0.01,
              weighting = T, main="Gráfico de relaciones A")
sent10

#### G.- Evolución de la polaridad: ####

tablasent$Pol[tablasent$Score > 0] <- 3
tablasent$Pol[tablasent$Score == 0] <- 1
tablasent$Pol[tablasent$Score < 0] <- 2
table(tablasent$Pol)

date1 <- as.IDate(datosent$created)
pola <- data.frame(date=date1, tablasent$Pol)
table(pola$tablasent.Pol)
pola$tablasent.Pol <- as.character(pola$tablasent.Pol)
pola$tablasent.Pol[pola$tablasent.Pol == 1] <- "Neutrales"
pola$tablasent.Pol[pola$tablasent.Pol == 2] <- "Negativos"
pola$tablasent.Pol[pola$tablasent.Pol == 3] <- "Positivos"

nivcol <- c("salmon","cornflowerblue","limegreen")

table(pola$date)

sent12 <- ggplot(data = pola) + geom_bar(aes(x = date, fill = tablasent.Pol),
                                       stat = "bin") + theme_light() +
  ggtitle("Evolución de la polaridad de los comentarios A") + xlab("Fecha") +
  ylab("Nº de comentarios") + scale_fill_manual(values=nivcol,
name="Polaridad")
sent12

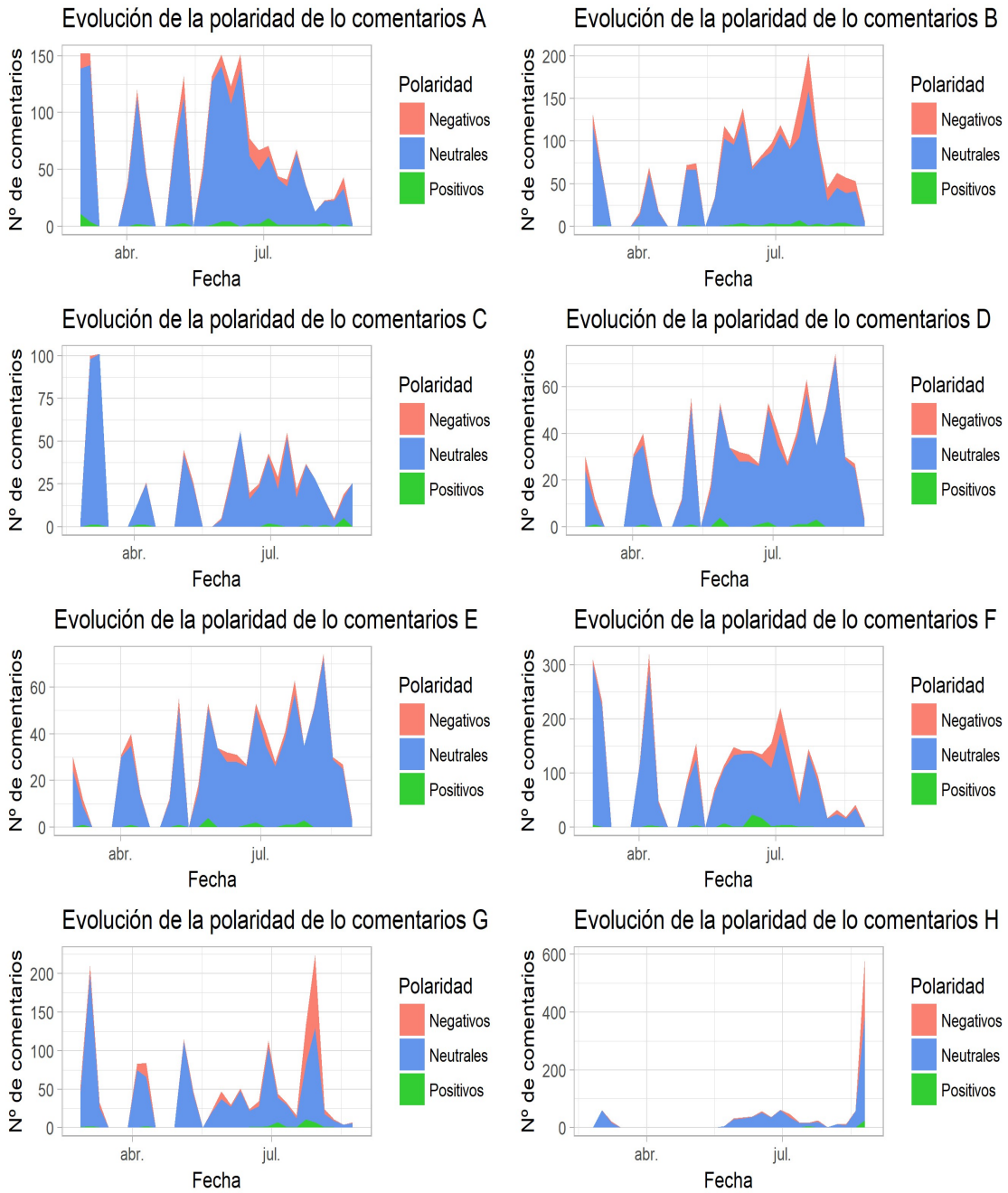
#### H.- Tablas: ####

ScoreMin <- round(min(COMPA$Score),digits=2)
ScoreMean <- round(mean(COMPA$Score), digits=2)
ScoreMax <- round(max(COMPA$Score), digits=2)
ScoreDesv <- round(sd(COMPA$Score), digits=2)
ScoreModa <- mlv(COMPA$Score,method = "discrete")
ScoreModa <- ScoreModa[1]

Tot <- sum(result1$slices1)
pos0 <- result1$slices1[result1=="Positivo"]
neg0 <- result1$slices1[result1=="Negativo"]
neu0 <- result1$slices1[result1=="Neutral"]
Pos <- round((pos0/Tot)*100,digits=1)
Neg <- round((neg0/Tot)*100,digits=1)
Neu <- round((neu0/Tot)*100,digits=1)
tablasent$spam <- tablasent$Positivo + tablasent$Negativo
Spam <- round(sum(tablasent$spam==0)/Tot*100, digits=1)
Posneg <- round(neg0/pos0,digits=2)

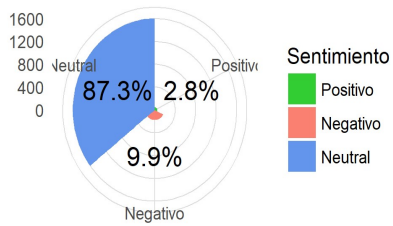
Tcompa <- cbind( nota_sent, Tot, Pos, Neg, Posneg, Neu,ScoreMean, ScoreMax,
                ScoreMin, ScoreModa, ScoreDesv, Spam)
rownames(Tcompa) <- "A"
write.csv(Tcompa, file = "Tcompa.csv")
```

## Anexo II.

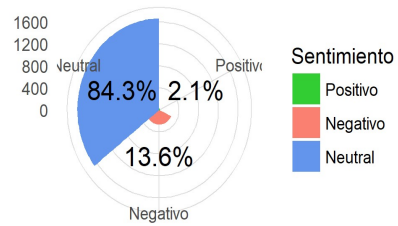




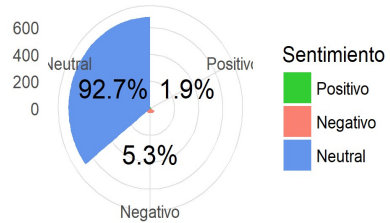
Polaridad de sentimiento A



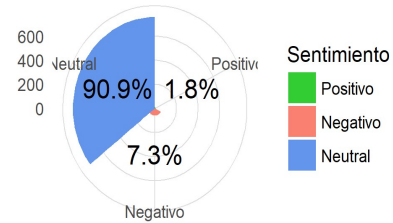
Polaridad de sentimiento B



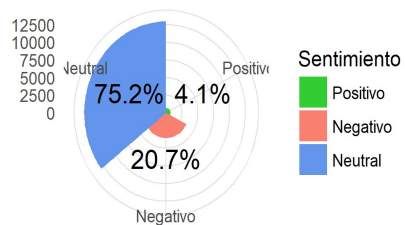
Polaridad de sentimiento C



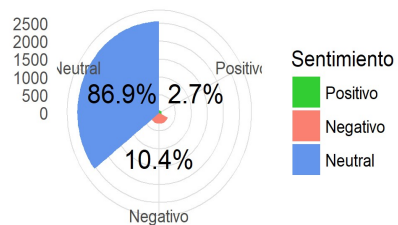
Polaridad de sentimiento D



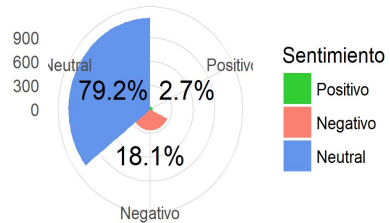
Polaridad de sentimiento E



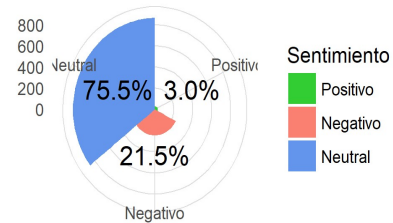
Polaridad de sentimiento F



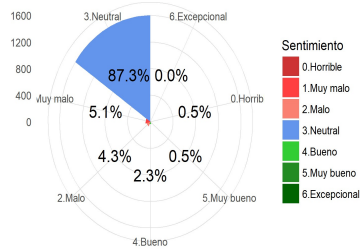
Polaridad de sentimiento G



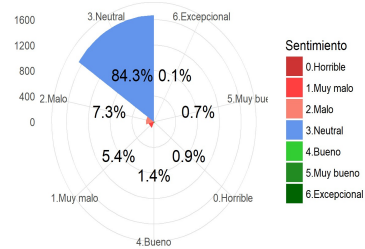
Polaridad de sentimiento H



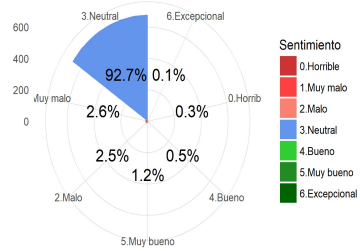
Clasificación de sentimientos A



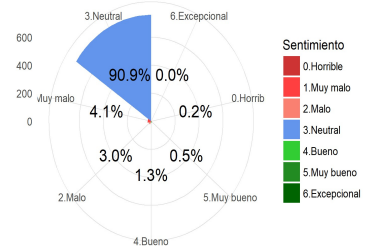
Clasificación de sentimientos B



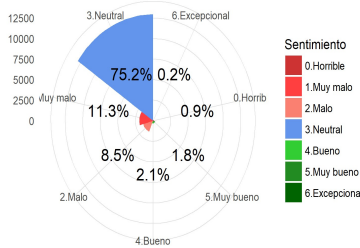
Clasificación de sentimientos C



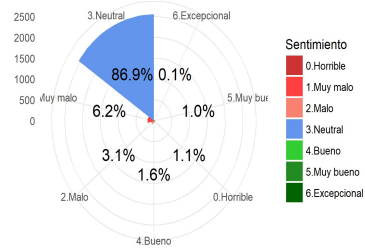
Clasificación de sentimientos D



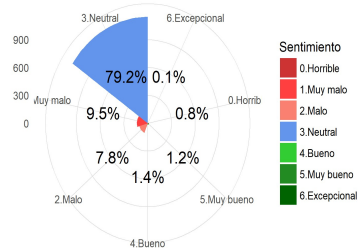
Clasificación de sentimientos E



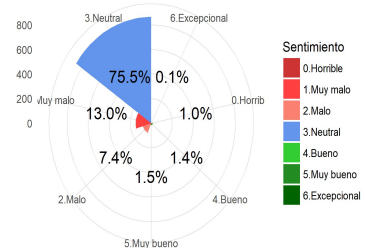
Clasificación de sentimientos F



Clasificación de sentimientos G



Clasificación de sentimientos H



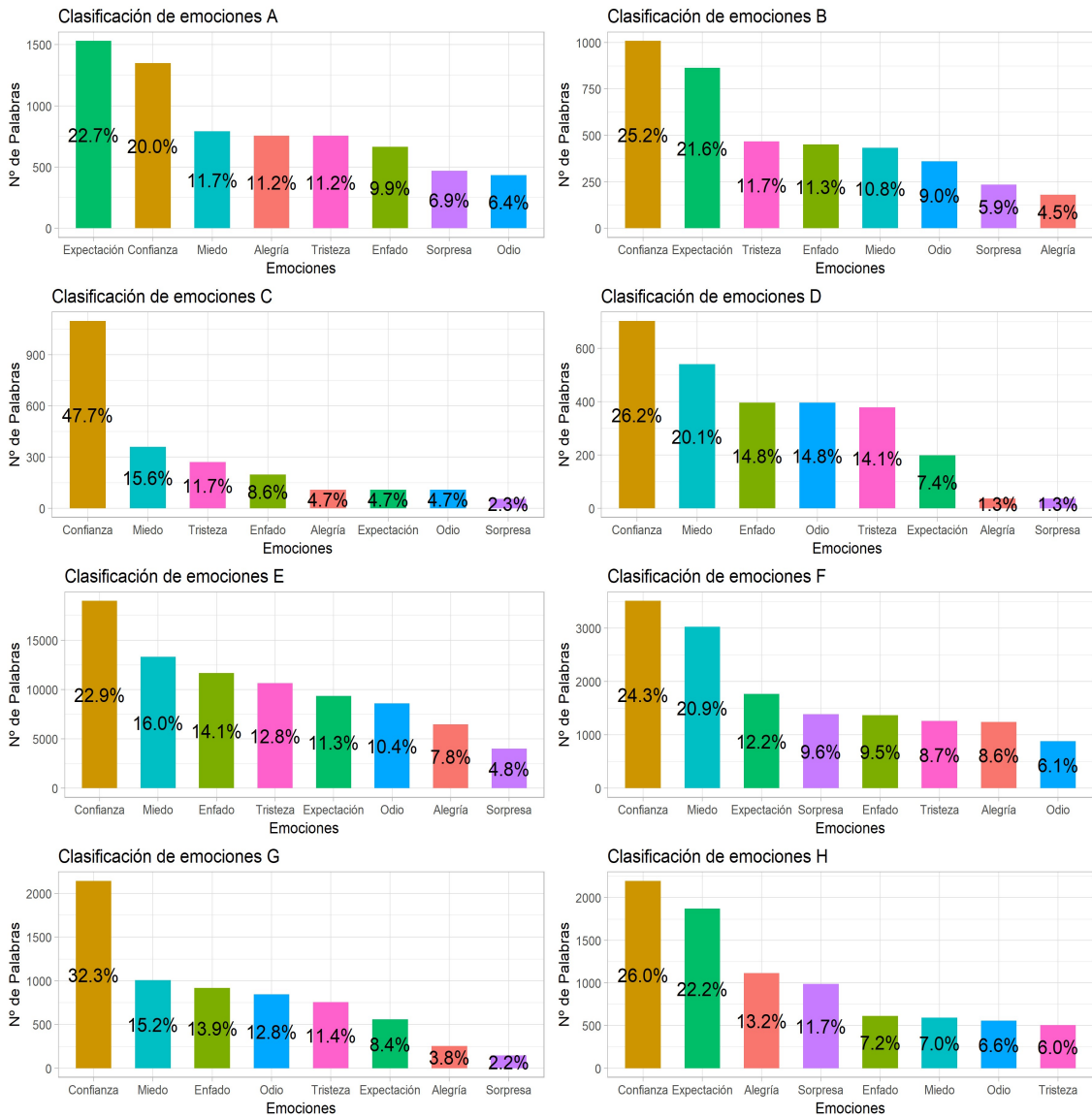




Gráfico de relaciones A

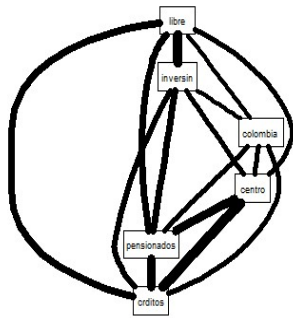


Gráfico de relaciones B

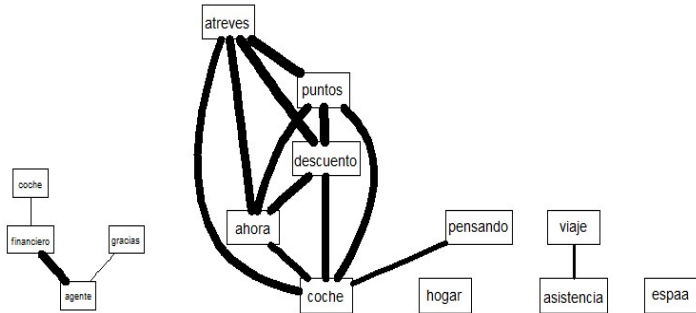


Gráfico de relaciones C

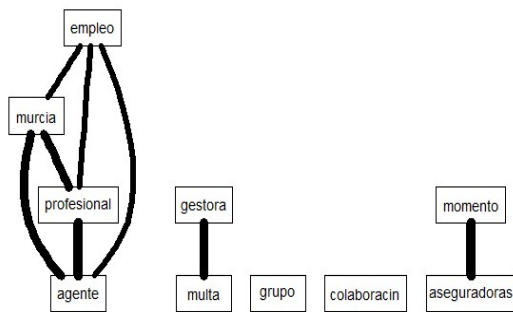


Gráfico de relaciones D

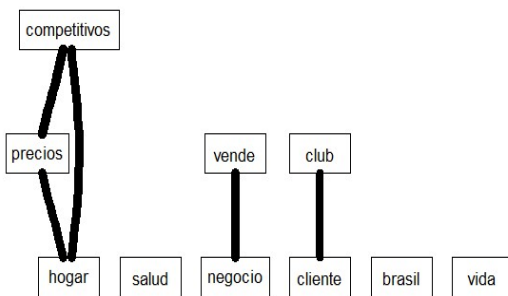


Gráfico de relaciones E

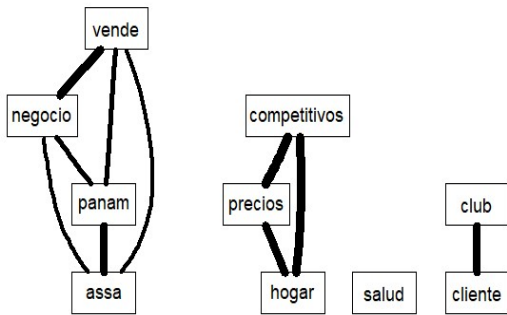


Gráfico de relaciones F

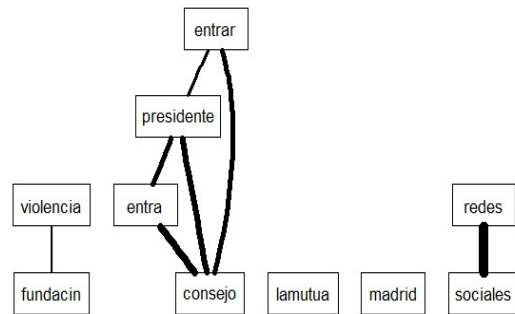


Gráfico de relaciones G

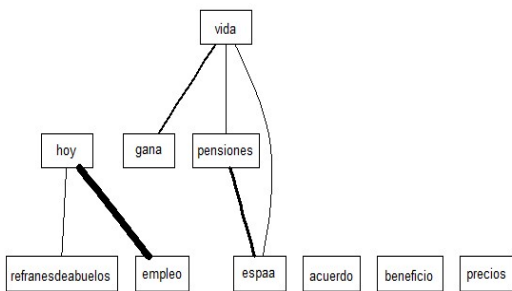


Gráfico de relaciones H

