Grado en Matemáticas **Sevilla, 2017**

TRABAJO DE FIN DE GRADO

Técnicas estadísticas en Minería de Textos

Ana Isabel Valero Moreno

Tutorizado por

Prof. José Luis Pino Mejías

Índice general

Resumen

Este trabajo presenta un análisis de distintas técnicas estadísticas existentes para la minería de textos, como son el Modelo de Espacio Vectorial Semántico, el Análisis de Semántica Latente y la Asignación de Dirichlet Latente.
Se explican técnicas relacionadas con el análisis de datos no estructurados como la minería de datos, el análisis de sentimientos, la extracción de información, la clasificación de documentos y la creación de resúmenes. Así como las etapas que hay que seguir para su realización y algunas áreas en las que se usa.
Se añade también, una lista de software que permite estudiar datos en forma de texto.
Finalmente, se desarrollan dos casos prácticos, donde se aplican algunos de los modelos introducidos a datos reales.
El primero es una pequeña aplicación usando el Análisis semántico latente para ver a qué documentos pertenece una consulta.
El segundo, se trata de una aplicación real de análisis de sentimientos para conocer las opiniones que tienen los usuarios sobre un producto a través sus comentarios.
El análisis de ambos se lleva a cabo mediante la aplicación informática estadística R.

Abstract

This paper presents an analysis of different statistical techniques for the text mining, like Semantic Vector Space Model, Latent Semantic Analysis and Latent Dirichlet Allocation.
Techniques related to the analysis of unstructured data such as data mining, sentiment analysis, feature extraction, clustering and creation of abstracts are explained. As well as the stages that must be followed for its realization and some areas in which it is used.
It is also added, a list of software that allow to study text data.
Finally, two practical cases are developed, where some of the introduced models are applied to real data.
The first is a small application using Latent Semantic Analysis to see which documents a query belongs to.
The second is a real application of sentiment analysis to know the opinions that users have about a product through their reviews.
The analysis of both is carried out by the statistical computer application R.

Índice de figuras

Capítulo 1

Introducción

El mundo actual genera más de 2,5 quintillones de bytes de datos a diario, y el 80 % de ellos son no estructurados (IBM Security, 2016). Es decir, que están expresados en lenguaje natural (hablado, escrito o visual) que un humano puede comprender fácilmente pero los software tradicionales no.

En la etapa actual de desarrollo de la humanidad y de la tecnología, la utilización de medios de comunicación como las redes sociales e internet en general ha conducido al aumento de la información online. Esto hace necesaria la creación de técnicas para el estudio de grandes volúmenes de datos textuales (no organizados en forma de datos). Desde el campo de las ciencias de la computación y la lingüística se están desarrollando nuevas herramientas para facilitar el acceso a mucha de la información que se genera diariamente.

El manejo cada vez más creciente de información en formatos no estructurados como mensajes de correo electrónico, respuestas de encuestas con final abierto, fuentes de noticias, publicaciones en redes sociales, etc se presenta como un problema para muchas empresas a la hora de preguntarse cómo recopilar, explorar y aprovechar toda esta información, lo que hace necesaria la denominada minería de textos o *text mining*.

Aunque la minería de textos nace en los años ochenta, los recientes avances informáticos han posibilitado el veloz crecimiento de esta área de estudio, que en un principio necesitaba de gran esfuerzo humano. Las nuevas técnicas posibilitan la explotación analítica de grandes volúmenes de información textual de un modo relativamente rápido y sin necesidad de emplear codificadores humanos. A pesar de que la mayor parte de las técnicas puramente automáticas de análisis textual se encuentran enfocadas a las ciencias de la computación y la lingüística ya existen algunos ejemplos de la aplicación de estas técnicas en otros ámbitos como el de las ciencias sociales.

Capítulo 2

¿Qué es la minería de textos?

La **minería textual** o *text mining* es una aplicación de la lingüística computacional y del procesamiento de textos que pretende facilitar la identificación y extracción de nuevo conocimiento a partir de colecciones de documentos textuales. Relacionada con la minería de datos podríamos decir que la minería textual es su "hermana pequeña". Por ello, profundizaremos en el siguiente punto sobre este término.
Para el procesamiento de un texto y su posterior extracción de información podemos describir cinco pasos fundamentales del proceso de la minería de textos:

1. Determinar el propósito del estudio. Teniendo claro lo que estamos buscando con la investigación.

2. Identificar el texto o los textos en los que se va a realizar el estudio. Si se encuentran en distintos archivos, procedemos a guardarlos en una misma ubicación para permitir el procesamiento posterior.

3. Pre-procesamiento del texto. En este paso procedemos a pasar todo el texo a minúculas, eliminar signos de puntuación y palabras vacías así como reducir dimensiones para facilitar el estudio posterior.

4. Determinación del modelo que de adecúe a nuestros objetivos. En este paso utilizaremos alguna de las ténicas estadísticas que definiremos más adelante.

5. Por úlimo nos queda analizar los resultados obtenidos, buscando coherencia, similitudes, etc. Aquí es cuando obtendremos conclusiones que nos ayuden a cumplir los objetivos propuestos en el primer paso.

2.1. La minería textual y su relación con otras disciplinas

Existe una clara relación entre minería de textos, minería de datos, recuperación de información y lingüística computacional.

2.1.1. Minería de textos y minería de datos (*Data Mining*)

A veces, la minería de textos resulta una actividad complementaria de la minería de datos. Esta última surgió para ayudar a la comprensión de los contenidos de las bases de datos. Se trata del proceso de detectar la información procesable de los conjuntos grandes de datos.
Para el *Data Mining* los datos son la materia prima bruta a los que los usuarios dan un significado convirtiéndolos en información que los expertos utilizarán para convertirla en conocimiento.
La minería de datos tiene muchos campos de aplicación pues puede ser útil en prácticamente todas las facetas de la actividad humana.
Veamos algunos ejemplos de los distintos campos en los que aplicamos minería de datos.

- **Campo empresarial** :

  Ponemos el ejemplo de un supermercado. La minería de datos para grandes superficies se llama "análisis de cesta de la compra". Por cada compra realizada, especialmente con tarjeta bancaria, se podría almacenar datos que nos permite conocer los gustos de los clientes, qué es lo que más compran, qué cantidades compran, etc. En una tabla de datos se podrían tener campos: cliente, gasto en leche, gasto en carne, gasto en refrescos, etc. En este caso la minería de datos nos permitiría tomar decisiones tales como, por ejemplo, la leche y el pan están muy correlacionados, convendría colocar ambos productos muy distanciados dentro del establecimiento para obligar al cliente a que recorra más distancia y al ver más productos incremente su consumo.
  Otro ejemplo sería que si un producto tiene mucha demanda se puede ofrecer con descuentos para servir de gancho y atraer a mayor número de clientes a ese supermercado.
  Estas técnicas se incluyen dentro de lo que se llama "Marketing Basado en Minería de Datos".

- **Campo sanitario** :

  En un hospital donde hay unos datos de pacientes y un diagnóstico se puede tener una tabla de datos que incluya datos: paciente, edad, nivel de azúcar, alergias, etc. En este caso, la minería de datos serviría para hacer un pre diagnóstico de la dolencia que con mayor probabilidad pueda tener un paciente en base a sus datos asociados. Estudiando y tratando los datos se pueden llegar a conclusiones, por ejemplo, si un paciente tiene más de 70 años, los glóbulos blancos muy altos y el azúcar alto es muy probable que esté desarrollando diabetes. Si el paciente corresponde a ese perfil, la decisión puede ser hacer unas pruebas específicas o poner cierto tratamiento preventivo.

- **Campo de trabajo para emprendedores** :

  Tenemos el ejemplo de una empresa de desarrollo de software. Un equipo de ingenieros puede desarrollar aplicaciones informáticas y por cada una de ellas se recopilan distintos datos relacionados con la métrica del software: horas de trabajo, líneas de código, número de errores por cada 10000 líneas de código, etc. En este caso la minería de datos serviría para saber el número de errores que previsiblemente se va a encontrar en un proyecto y el tiempo que puede necesitar para corregirlos, antes de que el proyecto en sí se haya desarrollado completamente.

Entendido qué es la minería de datos, podemos extrapolar la misma idea a la minería de textos.
La diferencia entre ambas aplicaciones radica en que con la minería de datos se pretende extraer conocimiento a partir de los patrones observables en grandes colecciones de datos estructurados. Por otro lado, la minería textual se tomará como punto de partida para la extracción de nuevos conocimientos documentales o de textos, es decir, información no estructurada.

2.1.2. Minería de textos y la recuperación de información

La recuperación de información tiene como objetivo identificar los documentos de interés dentro de un grupo, partiendo de la representación formal de dichos documentos sobre los que se realizará la búsqueda de información, así como de la formulación de la información necesaria para el usuario mediante un sistema de representación. Sin embargo, no pretende facilitar el proceso de extracción y análisis de nuevos conocimientos, al contrario que la minería de textos.
Además, cabe tener en cuenta que el objetivo de la recuperación de información se centra en tomar únicamente documentos útiles que satisfagan las necesidades del usuario mientras que la minería de texto no sólo no tiene esta necesidad, si no que necesita de una pregunta concreta para realizar el estudio.

2.1.3. Minería de textos y la lingüística computacional

Por su lado, la lingüística computacional tiene como objetivo el estudio gramatical y sintáctico de los documentos en formato electrónico, usando técnicas para procesarlos con el fin de que puedan ser comprensibles para un ordenador. Si bien la minería de textos necesita de la lingüística computacional, no comparten objetivo.

2.2. Historia de la minería de textos

Lo que conocemos como minería de textos ya surgió a comienzos de los años ochenta con la necesidad de gran esfuerzo humano. Los avances tecnológicos de la última década han hecho que esta área progrese rápidamente. Con el paso de los años hemos experimentado un exponencial aumento de información, más del 80 % almacenado como texto. Esto ha llevado a creer que la minería de textos tiene un gran valor comercial. Además, se le presta cada vez más atención a la minería de textos multilingual que permite ganar información en otros idiomas.

En 1977, el sistema THOMAS descubrió como ciertas frases o palabras claves eran un tipo muy útil de información abreviada a través de las cuales llegaríamos al descubrimiento de documentos de los que podíamos obtener información (Minería de textos, 2010). Sin embargo, la obtención de dichas palabras o frases eran puramente manual, eligiéndolas bien por los autores o títulos. En conclusión, condensaban documentos en algunas palabras o frases ofreciendo un resumen del texto.

El impacto alcanzado por la minería de datos no es comparable a la de la minería de textos, sin embargo, los atentados del 11 de septiembre de 2001 en Estados Unidos hicieron que distintos medios se interesaran por las tecnologías empleadas por la policía y organizaciones encargadas de luchar contra el terrorismo. Luego, a partir de esa fecha es cuando podemos encontrar una mayor cantidad de referencias de la minería textual y de datos con estos propósitos.

2.3. Etapas de la minería de textos

Para lograr resultados, la minería textual usa una serie de técnicas relacionadas con dos disciplinas mencionadas anteriormente: la recuperación de información y la lingüística computacional; entre ellas se incluyen:

2.3.1. Pre-procesamiento de los textos

Esta técnica tiene como fin dividir el texto de interés en distintas formas gráficas, definidas como una secuencia de letras delimitadas por espacios o signos de puntuación. Para ello se eliminan los espacios innecesarios y los signos de puntuación del documento original. Además, el programa informático debe convertir el texto en un formato plano, es decir, únicamente caracteres, sin distintos tipos de letra, negrita, subrayado, etc.
Dentro de esta etapa se encuentra la eliminación de palabras sin significado como las preposiciones, artículos y conjunciones.
Para finalizar se realiza lo que conocemos como normalización de las palabras obtenidas. Dicha normalización consiste en agrupar las palabras según su familia léxica. Por ejemplo, las palabras niño, niñez, niñera tiene misma raíz léxica (niñ-). Este proceso resulta interesante a la hora de ver el número de veces que aparecen y determinar las palabras idóneas que representan el contenido del texto. Para ello es necesario el uso de un diccionario y una base de conocimientos sobre las conjugaciones verbales, las familias léxicas, etc.
También conviene tener en cuenta las frases que más se repiten. Para ello es útil aplicar técnicas estadísticas que nos facilitarán la minería de textos. Por ejemplo, durante este trabajo se repetirán frases como "minería de textos", "análisis textual" o "técnicas estadísticas" las cuales definen a la perfección el tema principal del documento, que es el objetivo principal de la minería textual.

2.3.2. Identificación de nombres propios

Como podíamos imaginar la identificación de nombre propios ya sean de personas, organizaciones, empresas, etc, resulta fundamental para un buen análisis de minería de textos, así como la relación existente entre ellos y los términos que aparecen en el documento. Esto último necesita de técnicas de análisis sintácticos que permitan identificar verbos que sirvan de unión entre nombres propios y encuentren relación.

2.3.3. Representación de los documentos mediante un modelo

Para aplicar la minería textual es necesario representar el contenido de los documentos mediante un modelo. Generalmente el modelo que se usa hoy en día es el vectorial. Con este modelo, caracterizamos el documento mediante una serie de términos que lo representan. Estos términos pueden ser directamente sacados del documento (como hacíamos en el caso de pre-procesamiento de los textos) o con palabras que lo representen extraídas con un programa informático.
En este modelo, cada texto o documento es un vector. Cada vector es una estructura con un número fijo de componentes en la que la posición de cada elemento es importante. El proceso de recuperación de información consiste en la comparación de la distancia que existe entre los vectores (documentos) y un vector utilizado para representar la ecuación de búsqueda. De esta forma obtenemos una lista con los documentos más similares a la ecuación de búsqueda y los menos en orden ascendente o descendente. Más adelante hablaremos más en profundidad de este modelo.

2.3.4. Categorización automática

Esta técnica se encarga de clasificar los documentos de interés en categorías preestablecidas. Existen dos tipos de categorización. La primera, *singlelabel*, asigna a cada documento una categoría, mientras que la segunda, *multilabel*, asigna a un mismo documento más de una categoría.
Por último, podemos diferenciar este proceso entre *hard categorization* y *ranking categorization*. En la primera, el sistema clasificará cada documento en una categoría u otra mediante "verdadero" o "falso". Mientras que la segunda indica la probabilidad que tiene cada documento de pertenecer a cada categoría.

2.3.5. Relaciones entre términos y conceptos

Esta técnica se centra en la extracción de términos y conceptos y las relaciones existentes entre ellos.

2.4. Aplicaciones de la minería de textos

2.4.1. Extracción de información (*feature extraction*)

Lo que denominó Sullivan (2001) como *feature extraction* se centra en obtener nombres propios de personas, organizaciones o eventos, así como fechas y encontrar la relación existente entre ellas.
Por ejemplo, de un documento podrían obtenerse referencias a "Donald Trump", "presidente EEUU" y "relaciones comerciales con Cuba". Y encontrar relaciones entre éstos cómo "Donald Trump presidente de EEUU" y "Donald Trump prohibe las relaciones comerciales con Cuba".

2.4.2. Análisis de sentimientos o minería de opiniones (*sentiment analysis*)

Una de las técnicas de mayor interés para científicos sociales y periodistas por centrarse en analizar el vocabulario de un texto con el fin de determinar sus cargas emocionales (conocer si los mensajes contienen emociones positivas, negativas o neutras). El campo más útil es el de las redes sociales. Ayuda a revelar información importante sobre un tema específico.
Por ejemplo, a través del streaming de Twitter podemos clasificar el tono de los mensajes publicados por los usuarios para determinar el impacto de un cierto hecho (mediante etiquetas o hashtag) en distintas partes del mundo. Para ellos se pueden utilizar lo que llamamos diccionarios de sentimientos, que dan una valoración a cada palabra (positiva o negativa) y mediante la suma aritmética de cada palabra conocer, por ejemplo, los países con sentimientos más negativos hacia dicho hecho (hasgtag).

2.4.3. Clasificación de documentos (*clustering*)

Es especialmente útil para facilitar la recuperación y navegación de documentos. El *clustering* se encarga de agrupar documentos según la similitud que exista entre ellos. A diferencia de la categorización automática, el *clustering* generará grupos a partir de los documentos de los que disponemos. Recordemos que la categorización automática clasificaba los documentos según unas categorías preestablecidas.
Un ejemplo de uso puede darse en empresas que llevan un registro histórico de sus proyectos en documentos, si se deseara obtener información de las distintas áreas de desarrollo de un proyecto en concreto seguramente resultaría una tarea muy complicada. Gracias a algoritmos de minería de textos es posible agrupar dichos documentos, obteniendo información que resulta de utilidad.

2.4.4. Creación de resúmenes

Su objetivo es obtener una descripción general de un conjunto de documentos sobre un tema en específico.
Existen dos métodos de minería de textos para realizar resúmenes: el primero sería "copiar y pegar" información extraída de los documentos mientras que el segundo no necesariamente está compuesto de unidades de información de los documentos.

2.5. Áreas en las que encontramos minería de textos

2.5.1. Empresas

El incremento de los datos no estructurados, supone un gran reto para las empresas en el procesamiento de los mismos, con el fin de recaudar información útil que, tras el posterior análisis de las ideas principales, se traduzca en una toma de decisiones en tiempo real.
La minería de datos mejora exponencialmente dicha toma de decisiones por varios motivos:

- Resulta más efectiva a la hora de resolver los problemas empresariales.
- Más precisa al ofrecer productos a los clientes.
- Agiliza la interpretación de las opiniones de los clientes.
- Ahorro de recursos para analizar los datos manualmente.

Entre los distintos tipos de empresas que podemos encontrar, las empresas digitales, las compañías que tengan contacto directo con sus clientes (cadenas hoteleras, aerolíneas, etc) y las compañías que cuenten con servicios de encuestas de opinión, tienen más probabilidad de beneficiarse del software de minería de textos.
Algunos campos en los que es importante la minería de textos en las empresas:

- **Gestión de riesgos**

  Podemos definir la gestión de riesgo como el proceso de identificar, analizar y estudiar las probabilidades tanto de pérdidas como de efectos secundarios que pueden provocar desastres con el fin de realizar acciones preventivas, correctivas y reductivas para mejorar.
  Sea cual sea la industria que seleccionemos, un escaso análisis de riesgo es la causa principal de fracaso. Un sector importante es el financiero donde el uso de sistemas de gestión de riesgos orientados a la minería de texto puede aumentar la capacidad de reducir riesgos; analizando la cantidad de documentos en forma de texto.

- **Servicio de atención al cliente**

  Para el cuidado del cliente, la minería de textos, así como el procesamiento del lenguaje natural resultan de gran utilidad. Actualmente, uno de los principales usos es para mejorar la experiencia del cliente, conociendo sus opiniones mediante encuestas, comentarios sobre un producto, etc.
  El análisis de los textos nos permite dar una rápida respuesta al cliente sin necesidad de analizar uno a uno sus comentarios y reduciendo la necesidad de operadores de los centros de llamadas para solucionar problemas o tramitar quejas.

- **Opiniones de los clientes a través de redes sociales**

  Hoy en día, el número de usuarios en redes sociales aumenta exponencialmente. Estas son unas de las fuentes que nos proporcionan más cantidad de datos no estructurados, por ello muchas organizaciones las consideran una valiosa fuente para obtener información. Cada día más empresas usan la minería de textos para analizar o predecir las necesidades de los usuarios de sus productos, así como la percepción que tienen sus clientes sobre ellos.

2.5.1.1. Caso de compañías de seguros

Las compañías de seguros recogen a diario una enorme cantidad de información en forma de textos a través de sus agentes, correos electrónicos, agencias de atención al cliente, etc. Dicha información viene dada por pólizas, reclamaciones y quejas, resultados de encuestas e interacciones de usuarios a través de redes sociales entre otras. Debido a toda esta información encuentran necesaria la aplicación de tecnologías para análisis de texto como la minería de textos para atender, clasificar e interpretar la información útil.
Las aseguradoras, además, pretenden combinar los resultados del análisis de datos textuales con los datos estructurados para mejorar la toma de decisiones, lo que conlleva el uso tanto de la minería de textos como la de datos.
Uno de los temas más importantes a tratar en las compañías aseguradoras es la detención del fraude. (González, 2014) : "Según Accenture, en un informe publicado en 2013, las compañías de seguros pierden en Europa entre 8.000 y 12.000 millones de euros al año debido a reclamaciones fraudulentas. Además, se estima que entre un 5 % y un 10 % de las indemnizaciones abonadas por las compañías eran fraudulentas".
La toma rápida de decisiones ayuda a prevenir el fraude y a aumentar beneficios, por lo que la minería de textos es una herramienta muy útil para extraer información importante.

Por otro lado, el análisis de las quejas y reclamaciones de los clientes es otro punto fundamental para mejorar una compañía de seguros. Independientemente del canal de entrada de la reclamación (correo electrónico, encuestas, mensajes a través de redes sociales, etc.), la minería de textos es capaz de analizarla rápidamente para la posterior toma de decisiones.
En resumen, entre los beneficios que aporta la minería de texto en este sector están:

- Mejorar la productividad de los empleados.
- Reducir el trabajo de los empleados en los Centros de Atención al Cliente.
- Agilizar las respuestas a los clientes.
- Detección de los casos de fraude y de clientes defraudadores.
- Aumentar el grado de satisfacción de clientes y empleados.

2.5.1.2. Ejemplos

Airbnb es una web de hospedaje que une el poder de internet con las ganas de viajar de los usuarios. Comenzó en 2008 con un mercado en línea conectando huéspedes que buscan alquileres de vacaciones a corto plazo a los propietarios que desean alquilar sus viviendas. Las ganancias de dicha web provienen un 12 % de cuotas de reserva de los huéspedes y un 3 % de honorarios de servicio a los propietarios. Airbnb ha conectado a más de 20 millones de huéspedes con más 2.000.000 de propiedades en 192 países y 33.000 ciudades (Venkatesan, 2017). Su modelo de negocio ha influido en los últimos 5 años considerablemente en la mayoría de sus competidores. ¿Cómo podría mantenerse por delante de las empresas similares? Había logrado éxito ayudando a los anfitriones a mostrar mejor sus hogares y a los huéspedes a hallar con facilidad lo que estaban buscando.

Una de las claves de su éxito fue utilizar la minería de textos para traducir los comentarios del sitio web en datos útiles. En la minería de texto, el software utiliza algoritmos para asignar valores a palabras y patrones positivos y negativos, cuantificando así las preferencias del consumidor. De esa forma Airbnb ha implementado cambios como destacar los mejores anfitriones, sugerirles un baremo de precios y resaltar diferentes características en cualquier ubicación. El análisis de sentimientos a través de la minería de texto es fundamental a la hora de ayudar a personalizar una experiencia de un usuario y eliminar las conjeturas del marketing y la estrategia de una empresa.

Además, resulta interesante tener en cuenta otros temas como el diseño de la experiencia de los usuarios, los errores, el rendimiento, la privacidad, la complejidad de uso de la web y la atención al cliente.

Las redes sociales e internet han modificado la industria hotelera. La cantidad de páginas webs y aplicaciones móviles que proporcionan reseñas de contenido relacionado con viajes es cada vez mayor debido a la proliferación de comentarios realizados por los viajeros. Esto sitúa a los viajeros entre los consumidores más influyentes del mundo.

Para asegurar su éxito futuro, NH Hoteles se puso como reto desarrollar una solución para reunir y analizar la creciente cantidad de comentarios de sus clientes.

Según Google Cloud Platform, NH, con la ayuda de la consultoría Paradigma, cuenta actualmente con una herramienta de minería de textos con la que procesan más de 200.000 comentarios de clientes al año, 'Quality Focus Online'.

Gracias a esto, NH Hoteles ha conseguido convertir las redes sociales en el éxito de su negocio, ayudando a la toma de decisiones estratégicas y financieras para satisfacer las peticiones de los huéspedes casi a tiempo real.

Uno de los problemas que solucionaron después de la minería de textos fue proporcionar el acceso a Wi-Fi a lugares donde antes no llegaba. Lo que ha supuesto una media del 20 % menos de opiniones negativas.

Otro punto a tener en cuenta a la hora de la toma de decisiones a raíz de los comentarios es la relación directa entre gastos y satisfacción del cliente. A través de la frecuencia de términos, podemos evaluar si es significativo o no entre el conjunto de clientes.

2.5.2. Filtrado de correo no deseado (SPAM)

El correo electrónico es cada día utilizado por millones de personas en el mundo por ser una forma rápida, eficaz y barata de comunicarse. Uno de los problemas de su uso es el correo SPAM el cual va en aumento, y por ello son necesarias técnicas y métodos para erradicar este problema. Con el fin de evitar los llamados "correos basura" que son transmisores de virus e información inútil, existen técnicas en minería de textos. Estas realizan métodos de filtrado estadístico que, a partir de las palabras que aparecen en dichos correos SPAM, permiten diferenciar entre correos deseados y no deseados.

2.5.3. Prevención de la cibercriminalidad

A diario se publican cientos de blogs sobre seguridad con información sobre amenazas y cibercriminalidad. Para un analista de seguridad es muy complicado estudiar todo su contenido y, por otro lado, los sistemas de seguridad tradicionales son incapaces de analizar y aplicar dicho estudio como lo hace un analista.
El anonimato y la cantidad de funciones de comunicación que nos ofrece internet contribuye al aumento de delitos a través de este. La minería de textos junto con las aplicaciones contra el crimen ayudan a la prevención de la delincuencia en cualquier empresa, tanto privada como pública.

Capítulo 3

Técnicas en minería de textos

En este apartado desarrollamos algunas de las principales técnicas de la minería de textos. Todas ellas tienen la tarea de analizar el contenido de los textos con el fin de resumir de qué se tratan. Lo hacen con la lógica de que, contando el número de palabras del texto, la frecuencia con la que aparecen y su co-ocurrencia son buenos indicadores del tema del texto o de los sentimientos expresados en él.
Entre los modelos computacionales existentes para la representación de textos vamos a definir en profundidad cuatro: Modelo booleano, Análisis semántico latente, Asignación de Dirichlet Latente y Modelo de espacio vectorial.

3.1. Modelo Booleano

El Modelo Booleano resulta ser el modelo más sencillo y más útil tradicionalmente (Velasco, 2014). En este modelo cada documento o texto se representa mediante palabras clave. La importancia de cada palabra clave únicamente se mide por la presencia o ausencia de la misma en un documento. Una de las ventajas de este modelo es la simplicidad de implementación y la rapidez de realización. La gran desventaja es que se centra únicamente en las palabras que el analista tomó con antelación como relevantes, de forma que las demás quedan ignoradas.

3.2. Análisis de semántica latente (*Latent Semantic Analysis; LSA*)

Uno de los modelos de espacio vectorial es el análisis de semántica latente el cual supone que ciertas palabras aparentemente independientes estén relacionadas por temas subyacentes no observados. Por ejemplo, las palabras "soldado" y "tanque" pueden considerarse expresiones superficiales de un tema latente más relevante como es "guerra".

El LSA es una de las técnicas más actuales de la psicolingüística computacional, se trata de un modelo estadístico que permite determinar las distancias y relaciones semánticas entre piezas de información textual, ya sean entre palabras, frases o párrafos (V., 2003).

LSA fue originalmente descrito como un método de recuperación de información. Posteriormente concibieron este modelo como un modelo apto para la adquisición y la representación del conocimiento.

Entre las múltiples aplicaciones del LSA tenemos la corrección de textos en el ámbito académico y la medida de coherencia y cohesión textual.

El LSA comienza procesando un texto de grandes dimensiones que llamaremos corpus lingüístico. Dicho corpus contiene miles de palabras, párrafos y frases. Además, se representa como una matriz de frecuencias cuyas filas son las distintas palabras del corpus y cuyas columnas aparecen los distintos párrafos o frases. De esta forma la matriz contiene el número de veces que cada palabra aparece en el texto. A continuación, se realiza una ponderación con el fin de restar importancia a los términos muy frecuentes ya que en cualquier texto aparecen reiteradas veces artículos y determinantes que no aportan información relevante; y aumentarla a los menos frecuentes debido a que las palabras excesivamente frecuentes no nos sirven para seleccionar bien la información relevante del párrafo y las que aparecen de forma moderada sí.

El siguiente paso es aplicar un algoritmo denominado Descomposición en Valores Singulares (*Singular Value Descomposition; SVD*) con el fin de disminuir la dimensión de la matriz a una cifra más accesible sin perder información importante de la original (Botana, 2010).

Otro propósito interesante de este algoritmo es el de obtener una matriz que contenga únicamente los vectores con información relevante.

La ventaja de representar el lenguaje vectorialmente es que los vectores son aptos a comparaciones por medio de distancias euclídeas, cosenos y otras medidas.

Además, a partir de las coordenadas de la matriz que tenemos se puede introducir en el espacio nuevos vectores que representen textos introducidos a posteriori llamados pseudodocumentos. Los pseudodocumentos son textos que añadimos al espacio

semántico reducido que tenemos y que no forman parte del corpus del que partimos. El LSA permite realizar el proceso a este último y añadirlo a nuestra matriz reducida sin necesidad de realizar el proceso de nuevo con todos los documentos.

3.2.1. Ejemplo LSA con R

Tomamos 5 documentos diferentes de Hamlet y de El Principito. Vamos a hacer un ejemplo simple para detectar a qué documento pertenece las palabras 'rey'
y 'principito'.
d1 = Hamlet
d2 = HORACIO.– Yo le conocí personalmente. Era un buen rey.
d3 = HAMLET.– El Rey se divierte.
d4 = Y así fue como conocí al principito.
d5 = La prueba de que el principito ha existido está en que era un muchachito encantador.
Tenemos, en primer lugar, la matriz de términos-documentos de dimensión m x n (10x5 en el ejemplo), A, donde cada columna corresponde a un documento. Si el término i aparece a veces en el documento j, entonces A[i,j] = a.

	d1	d2	d3	d4	d5
buen	0	1	0	0	0
conocí	0	1	0	1	0
divierte	0	0	1	0	0
era	0	1	0	0	1
hamlet	1	0	1	0	0
horacio	0	1	0	0	0
personalmente	0	1	0	0	0
principito	0	0	0	1	1
que	0	0	0	0	2
rey	0	1	1	0	0

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

A través de esta obtenemos la matriz términos - términos

$$B = A \cdot A^T$$

de dimensión m x m (10x10) y la matriz documentos - documentos

$$C = A^T \cdot A$$

de dimensión n x n (5x5)

$$B = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 2 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 2 & 0 & 1 & 1 & 1 & 2 & 1 \\ 0 & 0 & 1 & 0 & 2 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 4 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 2 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 6 & 1 & 1 & 1 \\ 1 & 1 & 3 & 0 & 0 \\ 0 & 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 1 & 6 \end{pmatrix}$$

Si los términos i y j aparecen juntos en el documento b, entonces B[i,j] = b. Por otro lado, si el documento i y j tienen c palabras en común, entonces C[i,j] = c.

Definimos ahora las matrices:
S : matriz de autovalores de B
U : matriz de autovalores de C
P : matriz diagonal cuyos elementos son las raices cuadradas de los autovalores de la matriz B
Donde:

$$A = S \cdot P \cdot U^T$$

$$P = \begin{pmatrix} 2{,}735991 & 0 & 0 & 0 & 0 \\ 0 & 2{,}286329 & 0 & 0 & 0 \\ 0 & 0 & 1{,}765846 & 0 & 0 \\ 0 & 0 & 0 & 1{,}271242 & 0 \\ 0 & 0 & 0 & 0 & 0{,}7434965 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Por medio de SVD vamos a reducir las dimensiones eliminando los valores mas pequeños de la matriz P, que como están ordenados por relevancia de mayor a menor resulta muy fácil. Nos quedamos con los k valores más grandes creando la matriz P_k. A consecuencia, vamos a reducir también las matrices S y U^T. Aproximamos la matriz A por:

$$A_k = S_k \cdot P_k \cdot U_k^T$$

La dimensión de S_k será m x k, de P_k será k x k y de U_k^T será k x n. Luego, la matriz A_k tiene de nuevo dimensión m x n. Las palabras vendrán representadas por las filas de la matriz m x k

$$S_k \cdot P_k$$

mientras que los documentos lo estarán por las columnas de la matriz k x n

$$P_k \cdot U_k^T$$

Entonces la consulta será representada por el centroide de los vectores de sus términos.
Vamos a desarrollar el estudio para k = 2. Los términos están representados por las filas de S_2 y los documentos por las columnas de U_2^T.

$$P_2 = \begin{pmatrix} 2{,}735991 & 0 \\ 0 & 2{,}286329 \end{pmatrix}$$

$$S_2 = \begin{pmatrix} -0{,}25855642 & 0{,}26448823 \\ -0{,}34844520 & 0{,}24728081 \\ -0{,}05969270 & 0{,}13285910 \\ -0{,}49309786 & -0{,}05553351 \\ -0{,}06889652 & 0{,}16428793 \\ -0{,}25855642 & 0{,}26448823 \\ -0{,}25855642 & 0{,}26448823 \\ -0{,}32443022 & -0{,}33722917 \\ -0{,}46908288 & -0{,}64004349 \\ -0{,}31824912 & 0{,}39734733 \end{pmatrix}$$

$$U_2^T = \begin{pmatrix} -0{,}02518156 & -0{,}7074079 & -0{,}1633187 & -0{,}24593484 & -0{,}6417032 \\ -0{,}07185665 & -0{,}6047071 & -0{,}3037596 & 0{,}03934183 & 0{,}7316750 \end{pmatrix}$$

Las palabras están representadas por las filas de S_2 y los documentos por las columnas de U_2^T. Ahora calculamos

$$P_2 \cdot U_2^T$$

y

$$S_2 \cdot P_2$$

$$S_2 \cdot P_2 = \begin{pmatrix} -0{,}7074079 & 0{,}6047071 \\ -0{,}9533428 & 0{,}5653653 \\ -0{,}1633187 & 0{,}3037596 \\ -1{,}3491111 & -0{,}1269679 \\ -0{,}1885002 & 0{,}3756162 \\ -0{,}7074079 & 0{,}6047071 \\ -0{,}7074079 & 0{,}6047071 \\ -0{,}8876380 & -0{,}7710168 \\ -1{,}2834064 & -1{,}4633499 \\ -0{,}8707266 & 0{,}9084667 \end{pmatrix}$$

$$P_2 \cdot U_2^T = \begin{pmatrix} -0{,}06889652 & -1{,}935461 & -0{,}4468383 & -0{,}67287541 & -1{,}755694 \\ -0{,}16428793 & -1{,}382559 & -0{,}6944944 & 0{,}08994835 & 1{,}672850 \end{pmatrix}$$

De aquí tenemos:

$$buen = \begin{pmatrix} -0{,}7074079 \\ 0{,}6047071 \end{pmatrix}, \; conoci = \begin{pmatrix} -0{,}9533428 \\ 0{,}5653653 \end{pmatrix}, \; divierte = \begin{pmatrix} -0{,}1633187 \\ 0{,}3037596 \end{pmatrix},$$

$$era = \begin{pmatrix} -1{,}3491111 \\ -0{,}1269679 \end{pmatrix}, \; hamlet = \begin{pmatrix} -0{,}1885002 \\ 0{,}3756162 \end{pmatrix}, \; horacio = \begin{pmatrix} -0{,}7074079 \\ 0{,}6047071 \end{pmatrix},$$

$$personalmente = \begin{pmatrix} -0{,}7074079 \\ 0{,}6047071 \end{pmatrix}, principito = \begin{pmatrix} -0{,}8876380 \\ -0{,}7710168 \end{pmatrix}, que = \begin{pmatrix} -1{,}2834064 \\ -1{,}4633499 \end{pmatrix},$$

$$rey = \begin{pmatrix} -0{,}8707266 \\ 0{,}9084667 \end{pmatrix}$$

y

$$d1 = \begin{pmatrix} -0{,}06889652 \\ -0{,}16428793 \end{pmatrix}, d2 = \begin{pmatrix} -1{,}935461 \\ -1{,}382559 \end{pmatrix}, d3 = \begin{pmatrix} -0{,}4468383 \\ -0{,}6944944 \end{pmatrix}, d4 = \begin{pmatrix} -0{,}67287541 \\ 0{,}08994835 \end{pmatrix},$$

$$d5 = \begin{pmatrix} -1{,}755694 \\ 1{,}672850 \end{pmatrix}$$

Ahora calculamos el vector centroide para nuestra consulta:

$$q = \frac{\begin{pmatrix} -0{,}8707266 \\ 0{,}9084667 \end{pmatrix} + \begin{pmatrix} -0{,}8876380 \\ -0{,}7710168 \end{pmatrix}}{2} = \begin{pmatrix} -0{,}8791823 \\ 0{,}06872495 \end{pmatrix}$$

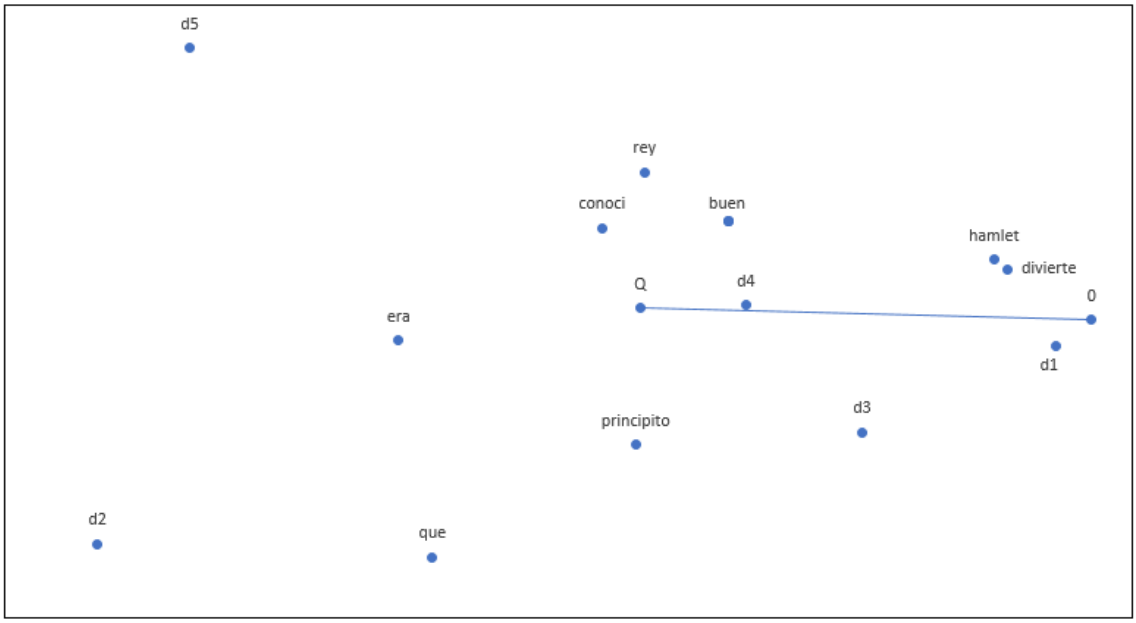

Figura 3.1: Representación geométrica de los documentos, sus palabras y la consulta.

3.3. Asignación de Dirichlet Latente (*LDA*)

El modelo de asignación de Dirichlet latente se trata de un *Topic modeling* y se basa en la intuición de que cada documento contiene palabras o términos de distintos temas o tópicos. La proporción de los tópicos en cada documento es diferente, mientras que dichos tópicos son los mismos para todos los documentos (Elkan, 2014). Dicho modelo necesita conocer a priori los textos y el número de tópicos a los que hacen referencia los textos.
El modelo LDA sostiene que los tópicos y los documentos sean dibujados mediante distribuciones de Dirichlet. LA función de densidad de la distribución de Dirichlet se define como:

$$P(x|\alpha_1, ..., \alpha_k) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \cdot \prod_{i=1}^{k} x_i^{\alpha_i - 1}$$

Con α un vector positivo de dimensión k (parámetro de Dirichlet) y Γ la función Gamma que viene dada por:

$$\Gamma(z) = \int_o^{\infty} t^{z-1} \cdot e^{-t}$$

Vamos a resumir que es lo que hace el LDA:

1. Para cada tópico: decide qué palabras son más probables.
2. Para cada documento: decide que proporción de tópicos hay en el documento.
3. Para cada palabra: elige un tópico y dado ese tópico elige la palabra más probable (del paso 1).

El modelo LDA consiste en extraer muestras de las distribuciones de Dirichlet y de distribuciones multinomiales (generalizacion de la distribución binomial).
Formalmente, podemos describir el modelo LDA de la siguiente forma:

1. Para cada tópico k, sacamos una distribución sobre las palabras $\varphi_k \sim Dir(\alpha)$.
2. Para cada documento d_i, obtenemos un vector proporcional de tópicos $\theta_d \sim Dir(\beta)$.
3. Para cada palabra i:
   a) Tomamos una asignación de tópicos $z_{d,i} \sim Mult(\theta_d)$ con $z_{d,i} \in 1, ..., k$
   b) Tomamos una palabra $w_{d,i} \sim Mult(\varphi_{z_{d,i}})$ con $w_{d,i} \in 1, ..., V$

K	número de tópicos $k \in 1, ..., K$
V	número de palabras en el vocabulario
α	vector positivo de tópicos de dimensión K
β	vector positivo de palabras de dimensión V
$Dir(\alpha)$	distribución de Dirichlet parametrizada por el vector α de dimensión K
$Dir(\beta)$	distribución de Dirichlet parametrizada por el vector β de dimensión V
z	índice de tópicos.$z_{d,i} = k$ significa que a la palabra i del documento d se le ha asignado el tópico k

Cuadro 3.1: Símbolos utilizados para el LDA

El objetivo es encontrar valores probables para variables ocultas, es decir, los vectores de parámetros multinomiales cuyo tema z pertenece a cada palabra de cada documento.

3.4. Modelo de espacio vectorial semántico

Como ya definimos brevemente en el punto 2.3, el modelo de espacio vectorial surge como mejora del modelo Booleano y representa conjuntos de documentos mediante vectores de términos. Se trata de uno de los modelos más usados en tareas de procesamiento del lenguaje natural y recuperación de información (*Natural Language Processing; NLP*).
Este modelo considera los textos como bolsas de palabras pero no es capaz de obtener relaciones semánticas entre las palabras del mismo. Según García (2016), en el modelo vectorial, el enfoque semántico de las palabras se dividen en dos partes: la semántica distribucional y la semántica composicional. El primero analiza los significados de cada palabra aislada mientras que el segundo lo hace de frases o párrafos. El estudio del significado de las expresiones lingüísticas está relacionado, desde el punto de vista léxico, con una noción de distancia. Por ejemplo, el significado de la palabra "colegio" está más cerca del de la palabra "alumno" que del de la palabra "coche". De esta forma, los significados de las palabras se pueden representar en forma de vectores como parte del espacio semántico completo. Podemos comparar las palabras a través de similitudes en el espacio vectorial mediante alguna distancia. Como ya dijimos anteriormente, una de las más usadas es la del coseno que realiza comparaciones calculando el ángulo entre dos vectores.
A continuación, veremos más en profundidad las dos formas, ya mencionadas, de estudiar los significados de palabras en el modelo vectorial.

3.4.1. Semántica distribucional

La semántica distribucional se encarga de obtener patrones estadísticos de las palabras a partir de los cuales encontramos las disimilitudes y semejanzas entre ellas. Describimos, a continuación, los tipos de modelos distribucionales semánticos.

3.4.1.1. Modelos distribucionales semánticos basados en vectores de conteo

Los modelos distribucionales basados en vectores de conteo siguen cuatro etapas fundamentales para estudiar la coocurrencia de palabras en un corpus. Dichas etapas son:

3.4.1.1.1. Extracción de cantidades de coocurrencia. Los modelos distribucionales semánticos basados en vectores de conteo representan mediante una matriz la frecuencia de aparición de palabras en un documento. Existen tres tipos de matrices según las similitudes que analicen:

- **Matrices término-documento** Estas matrices presentan en sus filas a los términos y en las columnas a los documentos. Es una de las más utilizadas para representar documentos. Con dichas matrices se cuentan las veces que ocurre cada término, pero no se tiene en cuenta el orden en el que aparecen. La mayoría de los elementos son 0 pues en la mayoría de los documentos aparecerá una parte de todo el vocabulario.
  Por último, cabe mencionar que los vectores columna de dichas matrices indican si los textos son similares o no.

- **Matrices palabra-contexto** En estas matrices las filas corresponden a los términos y las columnas a contextos. Se basan en la hipótesis de que las palabras que aparecen en contextos similares propenden a tener significados similares. En estas matrices, al contrario que las anteriores, los vectores fila similares son los que indican palabras con significados similares. En el caso de que los contextos sean reducidos a palabras (palabra objetivo) cercanas a la palabra de la que queremos obtener palabras similares, tenemos las matrices palabra-palabra. En estas se cuentan el número de veces que una palabra contexto ocurre en el contexto de una palabra objetivo.

- **Matrices par-patrón** Sus filas presentan pares de palabras, por ejemplo, coche-mecánico y sus columnas corresponden a los patrones en los que los pares ocurren, por ejemplo, “X es arreglado por Y”, “Y arregla a X”. En este caso, se trata de medir la similitud semántica de los vectores columnas.

3.4.1.1.2. Esquemas de pesos. El esquema de pesos más utilizado es la frecuencia de aparición de los términos en los textos (*Term Frequency / Inverse Document Frequency; TF-IDF*) para expresar el peso referente al término w en el vector asociado al texto o documento d. Se calcula mediante la ecuación:

$$tfidf(w,d) = tf(w,d) \cdot idf(w)$$

Donde:
tf(w,d) es la cantidad de veces que ocurre la palabra w en el documento d.
idf(w) es la frecuencia inversa de documentos, es la cantidad de documentos donde aparece la palabra w pero de forma inversa. A menor cantidad de documentos en los que aparece una palabra mayor peso. Se calcula mediante la ecuación:

$$idf(w) = log(\frac{N}{df(w)})$$

df(w) es la frecuencia de documentos, es decir, la cantidad de documentos en los que aparece la palabra w.
N es la cantidad de documentos de los que disponemos en el corpus.

3.4.1.1.3. Métodos de reducción de dimensiones. Entre los modelos para reducir dimensiones encontramos:

- **Análisis semántico latente (*LSA*)**

  Este modelo, mencionado anteriormente, emplea una técnica para disminuir las dimensiones de sus matrices llamada descomposición de valores singulares (*Singular Value Descomposition; SVD*), la cual pretende encontrar un espacio semántico latente mediante la factorización de matrices. Esta técnica descompone una matriz de término- documento en un conjunto de factores ortogonales, desde los cuales la matriz original puede aproximarse por una combinación lineal.
  La SVD motiva relaciones entre columnas y filas que se asemejan a las columnas y filas de la matriz de coocurrencia original, de esta forma el LSA reúne los términos que aparecen en contextos parecidos.

- **El hiperespacio análogo al lenguaje (*HAL*)**

  El HAL usa una matriz de coocurrencia palabra-palabra que contiene coocurrencias de palabras dentro de un contexto direccional de 10 palabras. Las coocurrencias son medidas según la distancia entre las palabras. Las palabras que ocurren cercanas en la ventana de contexto tienen mayor peso y las que ocurren en lados opuestos tienen menor peso.
  El resultado de aplicar esta técnica es una matriz de coocurrencia direccional cuyas filas y columnas simbolizan cantidades de coocurrencia en distintas direcciones. Cada par fila y columna representan las coocurrencias del contexto izquierdo y derecho. Dichos pares se concatenan para obtener un vector de contexto de gran dimensión.
  Para reducir dicha dimensión, el HAL calcula las varianzas de los vectores filas y columnas eliminando los de menor varianza. Es a partir de aquí donde se procede a estudiar la semejanza entre vectores.

- **El análisis semántico latente probabilístico (*PLSA*)**

  El PLSA se trata de un *Topic modeling* y es una versión probabilística del LSA. Este modelo se usa para descubrir la semántica de tópicos ocultos en textos mediante una bolsa de palabras.

- **El modelo de asignación latente de Dirichlet (*LDA*)**

  El LDA es un modelo probabilístico generativo bayesiano de tres niveles (documento, término y tópico). Considera el tópico como "una distribución sobre un vocabulario fijo". El modelo coge una serie de tópicos predefinidos y toma las palabras asociadas a estos tópicos. El fin de dicho modelo es estudiar como aparecen los tópicos en los textos.
  El primer paso es seleccionar un conjunto de tópicos predefinidos junto a sus palabras más probables. El segundo sería escoger una serie de tópicos para cada palabra que aparece en el texto y seleccionar una palabra para cada tópico. Por último, habría que agrupar las palabras por tópicos, es decir, los tópicos asociados a las palabras más frecuentes. De esta forma obtenemos el o los tópicos que mejor definen el contenido del documento.

3.4.1.1.4. Coeficientes para comparar documentos. Para saber cuándo una palabra, párrafo, oración o documento es similar a otra se conocen varias funciones de similitud útiles para la recuperación de información, agrupamiento de documentos, detección de tópicos, etc. Encontrar la similitud entre palabras es fundamental para obtener la similitud de documentos, párrafos u oraciones.
Existen dos formas de similitud en cuanto a las palabras: similitud léxica si tienen secuencias de caracteres similares (mismo morfema, lexema, etc) y similitud semántica si tienen el mismo significado o son usadas en el mismo contexto. La primera se presenta a través de algoritmos basados en cadenas y la segunda lo hace mediante algoritmos basados en corpus y en conocimiento.

- **Medidas basadas en cadenas:** actúan en secuencia de cadenas y composición de caracteres. Evalúan la semejanza o diferencia entre dos cadenas de textos para compararlas o estimar su equivalencia.
- **Medidas basadas en corpus:** determinan la semejanza entre palabras a través de la información que se obtiene de grandes textos.
- **Medidas basadas en conocimiento:** especifican el grado de semejanza entre palabras usando la información obtenida de redes semánticas.

La semejanza entre dos vectores de palabras se puede obtener a través del ángulo que forman, específicamente mediante el coseno del ángulo. Se considera que cuanto más pequeño sea el ángulo, y en consecuencia el coseno del ángulo, mayor similitud habrá entre ellos.

Con la siguiente ecuación obtenemos la medida coseno entre los documentos d_i y d_j, donde d_{ik} es el peso del rasgo semántico k en el documento d_i.

$$sim(d_i, d_j) = cos(a) = \frac{\sum_{k=1}^{m} d_{ik} \cdot d_{jk}}{\sqrt{(\sum_{k=1}^{m} d_{ik}^2) \cdot (\sum_{k=1}^{m} d_{jk}^2)}} = \frac{d_i}{|d_i|} \cdot \frac{d_j}{|d_j|}$$

Con la siguiente fórmula de la distancia Euclidiana se mide cómo de lejos se encuentran dos vectores en el espacio vectorial. Esta fórmula sólo será útil cuando se trate de dos vectores con dimensiones no muy grandes.

$$dist(d_i, d_j) = \sqrt{\sum_{k=1}^{m} (d_{ik} - d_{jk})^2}$$

Para finalizar este apartado comentamos la forma de obtener la similitud entre palabras en modelos probabilísticos. Se trata de calcular el alcance que comparten los mismos tópicos mediante las distribuciones de tópicos condicionales $\theta^{(1)}$ y $\theta^{(2)}$:

$$\theta^{(1)} = P(z|w_i = w_1)$$

$$\theta^{(2)} = P(z|w_i = w_2)$$

Donde w_1 y w_2 son las palabras y z el tópico.

3.4.1.1.5. Modelos basados en la predicción de contextos. Un principal objetivo de los modelos basados en la predicción de contextos es representar vectores de aprendizaje de palabras usando redes neuronales.
La idea del modelo de red neuronal es el representar cada palabra por un vector concatenado con vectores palabra en un contexto obteniendo un vector útil para predecir otras palabras en el contexto.
Un ejemplo sería el tomar un vector con la palabra "bueno" y, mediante espacios vectoriales con palabras semánticamente semejantes, realizar una concatenación de los vectores palabra "amable", "solidario", etc.
Las etapas principales para aprender el modelo de red neuronal son:

- Asignar a cada palabra del vocabulario un vector de rasgos de palabra.
- Determinar una función de probabilidad unificada (*joint probability function*) de secuencias de palabras dependiendo de los vectores de rasgos de dichas palabras.
- Estudiar los vectores de rasgos de palabra y los parámetros de la función de probabilidad unificada.

El vector de rasgos muestra diferentes aspectos de una palabra, cada palabra está asociada a un punto en el espacio vectorial. La función de probabilidad anterior se muestra como el producto de probabilidades condicionadas de la próxima palabra en función de las anteriores.

3.4.2. Semántica composicional

En el apartado 3.2.1 hemos visto que la semántica distribucional permitía conocer si dos palabras significaban lo mismo mediante una representación vectorial.
En este apartado, queremos conocer si dos oraciones significan lo mismo. Para ello no podemos usar el mismo método ya que no se pueden obtener rasgos distribucionales de una oración.
La semántica composicional permite estudiar una jerarquía de rasgos, donde los altos niveles de abstracción se obtienen mediante niveles más bajos.
Una función general de composición semántica se puede manifestar mediante u y v que son representaciones pequeñas, R que es la información relacional y K que es el conocimiento histórico.

Capítulo 4

Software

En este apartado describiremos algunas de las aplicaciones comerciales específicamente desarrolladas para la minería de textos (Senso, 2004). Definiremos 5 softwares con el fin de ver cómo los fabricantes han implementado las técnicas descritas en el punto anterior, y mostrar algunas de las aplicaciones de las que disponemos en el mercado para estudiar la minería textual.

4.1. R

R es un software para la manipulación de datos, cálculo y representación gráfica. Cuenta con una serie de paquetes para la minería de textos, entre los que se encuentran:

- *tm*: específico para la minería de textos.
- *qordcloud*: para realizar nubes de palabras.
- *ggplot2*: una gramática de gráficas que expande las funciones básicas de R.
- *readr*: permite leer y escribir documentos.
- *cluster*: que contiene funciones para realizar análisis de grupos.
- *dplyr*: incluye funciones auxiliares para manipular y transformar datos.

Más adelante se muestra una aplicación de minería de textos con R que nos ayudará a ver las funciones de estos paquetes y cómo actúan.

4.1.1. Minería de textos en Twitter con R

Vamos a mencionar el paquete *TwitteR* de R que nos permite obtener los datos que cede Twitter a través de sus APIs. Este paquete nos permite conseguir los tweets de los usuarios que nos interesen, sus seguidores, a quien sigue y sus timelines entre otras cosas. Además, nos permite extraer los tweets relacionados con un tema, así como la ubicación de los usuarios que los publican. Este paquete es de utilidad en muchos aspectos relacionados con las redes sociales y el Big Data como:

- Minería de textos. Mediante los tweets de uno o varios usuarios se pueden descubrir tendencias y comparar usuarios.
- Geolocalización de tweets y mapas de calor. TwitteR admite extraer los tweets relacionado con una palabra en un rango de tiempo. En el caso de que estén geoposicionados se pueden crear mapas de situación y de frecuencia sobre un tema en concreto.
- Evolución y tendencias de palabras. Debido a que podemos obtener tweets en rangos temporales podemos estudiar las tendencias de uso de una o varias palabras.

4.2. SAS Text Miner

De la misma forma que LexiQuest, Text Miner se trataba de una empresa independiente que fue comprada por SAS en mayo de 2002 (Senso, 2004).
Esta aplicación incorpora herramientas características de la minería de textos como:

- Capacidad de procesar documentos en varios formatos como pdf,ascii y html, extraer palabras o conjuntos de palabras, eliminar palabras vacías y reducir palabras por lexemas. El sistema incorpora tres idiomas: inglés, francés y alemán.
- Poder de identificación de la palabra en el documento con el objetivo de evitar ambigüedades.
- Representación de textos a través de un vector de términos ponderados según su frecuencia.
- Identificación de nombres propios (*feature extraction*)
- Agrupación automática de documentos (*clustering*)
- Categorización automática de documentos.

4.3. SPSS LixiQuest

IBM SPSS Statistics es un software de análisis estadístico que presenta las funciones necesarias para realizar un proceso analítico.
SPSS LixiQuest incluye tres herramientas: LexiQuest Mine, LexiQuest Categorize y LexiQuest Guide.
LexiQuest se trataba de una empresa independiente hasta que SPSS lo compró en febrero de 2002 para ampliar su campo de ejecución (Senso, 2004).

LexiQuest Mine tiene el propósito de automatizar el proceso de leer documentos con el fin de conocer su contenido. Ofrece la posibilidad de procesar una gran cantidad de documentos, identificar términos y nombres propios e informar sobre términos relacionados mediante su co-ocurrencia.

4.4. Megaputer TextAnalyst

Megaputer TextAnalyst funciona de la siguiente forma: parte de un texto o conjunto de textos en formato ASCII o RTF, a partir de ahí se identifican los principales términos. Una vez identificados los términos principales se le asocia una ponderación a cada una que equivale a la significación que tienen en el texto o textos que estamos usando. Los términos pueden ser una o varias palabras. En el caso de términos con varias palabras, el software estudia la frecuencia con la que aparecen juntas.
Este sistema no analiza sintáctica ni gramaticalmente los términos. Además, cada par de términos tiene asociado un valor que muestra la relación que tienen.

4.5. Google Cloud Platform

Google Cloud Platform se trata de una API de Google que dispone de bases de datos y almacenamiento, servicios de machine learning y big data entre muchos otros.
La API Natural Language de Cloud extrae información de textos no estructurados. Se puede usar para extraer información de textos, artículos o blogs, así como para saber las opiniones sobre un producto a través de comentarios en páginas webs o redes sociales.
Ofrece características como:

- Análisis sintáctico del texto.
- Análisis de opiniones.
- Análisis de textos en varios idiomas.
- Filtrar textos no apropiados.
- Clasificar documentos por temas.

Capítulo 5

Aplicación de minería de textos con R

En este apartado vamos a realizar una aplicación de minería de textos con R orientada a la extracción de información (feature extraction), una de las aplicaciones mencionadas anteriormente.

Contamos con datos reales cedidos por la empresa de videojuegos Genera Games, una de las principales empresas desarrolladoras de juegos y aplicaciones para iOS y Android.

Se tratan de los comentarios escritos por los usuarios sobre uno de sus juegos para móvil llamado "Gladiadores", en un periodo de aproximadamente dos meses (04/04/2017 - 05/24/2017).

En la tabla que se muestra a continuación podemos ver parte de los comentarios junto con otras variables como país, fecha y plataforma.

	A	B	C	D	E	F	G	H	I	J	K	L	M
14	Platform	Country	Date	App ID	App Name	Publisher ID	Publisher Name	User	Version	Rating	Title	Review	
15	iOS	Indonesia	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Tantandtaufik	1/07/2003	5	Good	Addicted game	
16	iOS	Thailand	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	สวนยาง	1/07/2003	1	Not received gen	Please check	
17	iOS	Thailand	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Explant	1/07/2003	5	Good	Good	
18	iOS	Thailand	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Wdcfdehv	1/07/2003	5	Good	Very good	
19	iOS	Thailand	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	เด็ก	1/07/2003	5	Good	Good good	
20	iOS	Thailand	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	สเนล	1/07/2003	5	good game	fun to play	
21	iOS	Thailand	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	จนจิงๆ	1/07/2003	5	มันส์	มันส์	
22	iOS	Thailand	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Tungggggggggg	1/07/2003	5	Good game	Good & fun	
23	iOS	Thailand	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	mummum555	1/07/2003	5	good game	good fun	
24	iOS	Thailand	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	1522562	1/07/2003	5	Ggwp	Good	
25	iOS	Thailand	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Wanchaimazda	1/07/2003	5	Good	Good	
26	iOS	Thailand	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Ddfhootfgj	1/07/2003	4	ก็พนุกดีนะแต่	สุ่มไม่เคยได้สกิลสดันสักที สกิลโคตร	
27	iOS	Thailand	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Monxmonx	1/07/2003	5	เพลิน เพลิน	ดี	
28	iOS	Thailand	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Mikoe17	1/07/2003	5	Mac	Very fun game	
29	iOS	Taiwan	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Yeh hsiu	1/07/2003	4	卡住	現在一直卡在榮耀之戰的動畫 沒有	
30	iOS	Saudi Arabia	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Dababa159	1/07/2003	5	:)	Goood	
31	iOS	Switzerland	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Kaiser sauce	1/07/2003	5	Cool	Bon jeu!!!	
32	iOS	Spain	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Peluki99	1/07/2003	2	Los hay mejores	Como es posible k te para subir d Un poco de idea a la hora de subi	
33	iOS	Spain	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Iphone 4s Marcu	1/07/2003	5	Bien	Divertido y entretenido sin absorb	
34	iOS	Spain	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Husmeador	1/07/2003	5	Vale la pena	Hasta ahora Muy entretenido y co	
35	iOS	Canada	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Fakknor	1/07/2003	4	Great	Good game... would be nice to be	
36	iOS	Sweden	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Indianen66	1/07/2003	5	GladiatorHeroes	Roligt och lättspelat	
37	iOS	Sweden	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	ColourMedia	1/07/2003	5	Sjukt kull	Rekommenderas	
38	iOS	Italy	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	lorisss46	1/07/2003	4	Carino	Carino	
39	iOS	Netherlands	05/24/2017	1061896024	Gladiator Heroes	324658540	Genera Mobile	Little Rickster	1/07/2003	5	Leuk	Leuk	

Figura 5.1: Tabla de datos generales Gladiadores.

Contamos en total con 4516 reviews en distintos idiomas. Vamos a estudiar únicamente los comentarios de usuarios en España. Por ello, mediante fórmulas con Excel seleccionamos los comentarios en español.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Platform	Country	Date	App ID	App Name	Publisher ID	Publisher Nai	User	Version	Rating	Title	Review	Language
2	iOS	Spain	05/24/2017	1061896024	Gladiator He	324658540	Genera Mob	Peluki99	01/07/2003	2	Los hay mejores	Como es posible k te para subir c	Español
3	iOS	Spain	05/01/2017	1061896024	Gladiator He	324658540	Genera Mob	Iphone 4s M	01/07/2003	5	Bien	Divertido y entretenido sin abso	Español
4	iOS	Spain	05/01/2017	1061896024	Gladiator He	324658540	Genera Mob	Husmeador	01/07/2003	5	Vale la pena	Hasta ahora Muy entretenido y c	Español
5	iOS	Spain	05/01/2017	1061896024	Gladiator He	324658540	Genera Mob	Patimipito	01/07/2003	5	Gran juego	Te lo pasas bien jugando	Español
6	iOS	Spain	05/01/2017	1061896024	Gladiator He	324658540	Genera Mob	ConGáGá	01/07/2003	5	Muy recomendable!	Me ha encantado el juego, busca	Español
7	iOS	Spain	05/01/2017	1061896024	Gladiator He	324658540	Genera Mob	Grusko	01/07/2003	5	Fantástico	Realmente divertido y entreteni	Español
8	iOS	Spain	05/01/2017	1061896024	Gladiator He	324658540	Genera Mob	€Willy€	01/07/2003	5	Genial	Divertido, original y más accesibl	Español
9	iOS	Spain	05/01/2017	1061896024	Gladiator He	324658540	Genera Mob	Fer5000	01/07/2003	5	Brutal me encanta !!	Divertido y adictivo sin la tediosa	Español
10	iOS	Spain	05/01/2017	1061896024	Gladiator He	324658540	Genera Mob	Nadie911	01/07/2003	5	Muy adictivo	Genial	Español
11	iOS	Spain	05/02/2017	1061896024	Gladiator He	324658540	Genera Mob	Loveiswhatw	01/07/2003	5	Adictivo!!!	Gráficos muy buenos lo único las	Español
12	iOS	Spain	05/02/2017	1061896024	Gladiator He	324658540	Genera Mob	rakataplam	01/07/2003	5	Fantástico	Muy bueno	Español
13	iOS	Spain	05/02/2017	1061896024	Gladiator He	324658540	Genera Mob	Dieeguitto	01/07/2003	4	Dije	La ----- con cebolla	Español
14	iOS	Spain	05/02/2017	1061896024	Gladiator He	324658540	Genera Mob	Piculon	01/07/2003	5	Entretenido	Entretenido	Español
15	iOS	Spain	05/02/2017	1061896024	Gladiator He	324658540	Genera Mob	gamercat300	01/07/2003	5	Muy bueno	Entretenido y sin demasiadas dif	Español
16	iOS	Spain	05/02/2017	1061896024	Gladiator He	324658540	Genera Mob	javie84	01/07/2003	5	Gran juego	Gran juego y muy adictivo!!!	Español
17	iOS	Spain	05/02/2017	1061896024	Gladiator He	324658540	Genera Mob	Sasha2.0	01/07/2003	5	Adictivo	Juego muy adictivo, sin necesida	Español
18	iOS	Spain	05/02/2017	1061896024	Gladiator He	324658540	Genera Mob	Iruxx	01/07/2003	5	No está mal	No está mal	Español
19	iOS	Spain	05/02/2017	1061896024	Gladiator He	324658540	Genera Mob	83737281	01/07/2003	5	Simpático y divertid	Muy guapo te entretiene un rato	Español
20	iOS	Spain	05/02/2017	1061896024	Gladiator He	324658540	Genera Mob	Jiruloedu	01/07/2003	5	Original y divertido	Entretenido muy bien	Español
21	iOS	Spain	05/02/2017	1061896024	Gladiator He	324658540	Genera Mob	Alkorte	01/07/2003	5	Gran juego de gladia	El mejor juego que tengo instala	Español
22	iOS	Spain	05/03/2017	1061896024	Gladiator He	324658540	Genera Mob	davidsr20	01/07/2003	4	España	Very nice one	Español
23	iOS	Spain	05/03/2017	1061896024	Gladiator He	324658540	Genera Mob	Alexitobi	01/07/2003	5	Buen juego.	👍🖼.	Español
24	iOS	Spain	05/03/2017	1061896024	Gladiator He	324658540	Genera Mob	jmos13	01/07/2003	5	Divertido y jugarle	Me gusta, gracias	Español
25	iOS	Spain	05/03/2017	1061896024	Gladiator He	324658540	Genera Mob	Karonte1983	01/07/2003	5	Brutal	Me encanta	Español
26	iOS	Spain	05/03/2017	1061896024	Gladiator He	324658540	Genera Mob	Te lo digo yo	01/07/2003	5	Vaya vicio	Gran juego, muy adictivo 👍🖼	Español
27	iOS	Spain	05/03/2017	1061896024	Gladiator He	324658540	Genera Mob	Ok1972	01/07/2003	5	Increíble	Llevo poco jugándolo pero si te g	Español

Figura 5.2: Tabla de datos en español Gladiadores.

Una vez que tenemos únicamente los datos que queremos, vamos a crear un archivo .txt con todas las reviews.

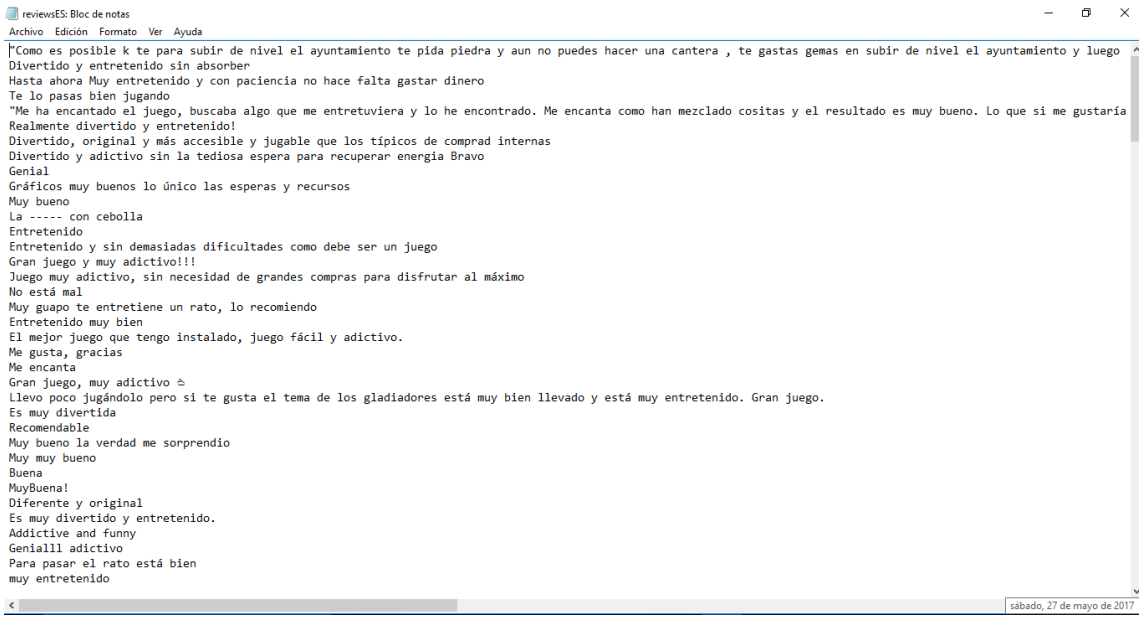

Figura 5.3: Lista de comentarios en español de Gladiadores.

Ya tenemos todo lo necesario para empezar a programar.
El primer paso es instalar los paquetes que nos ofrece R para la minería de textos, ya comentados en el Capítulo Softwares.

```
library(NLP)
library(tm)
library(RColorBrewer)
library(wordcloud)
library(readr)
library(ggplot2)
library(dplyr)
```

Procedemos ahora a leer el documento mediante la función read_lines del paquete readr y convertirlo en ASCII.

```
txt <- read_lines("reviewsES.txt")
txt = iconv(txt, to="ASCII//TRANSLIT")
```

Construimos, a continuación, el corpus:

```
corpus <- Corpus(VectorSource(txt))
```

Ahora, modificamos el documento transformándolo a minúsculas, quitando espacios en blanco, signos de puntuación y números:

```
d <- tm_map(corpus, tolower)
d <- tm_map(d, stripWhitespace)
d <- tm_map(d, removePunctuation)
d <- tm_map(d, removeNumbers)
```

Construimos un archivo .txt con palabras vacías en español, es decir, palabras que no resultan de interés para conocer el contenido o los sentimientos que contiene nuestro documento.

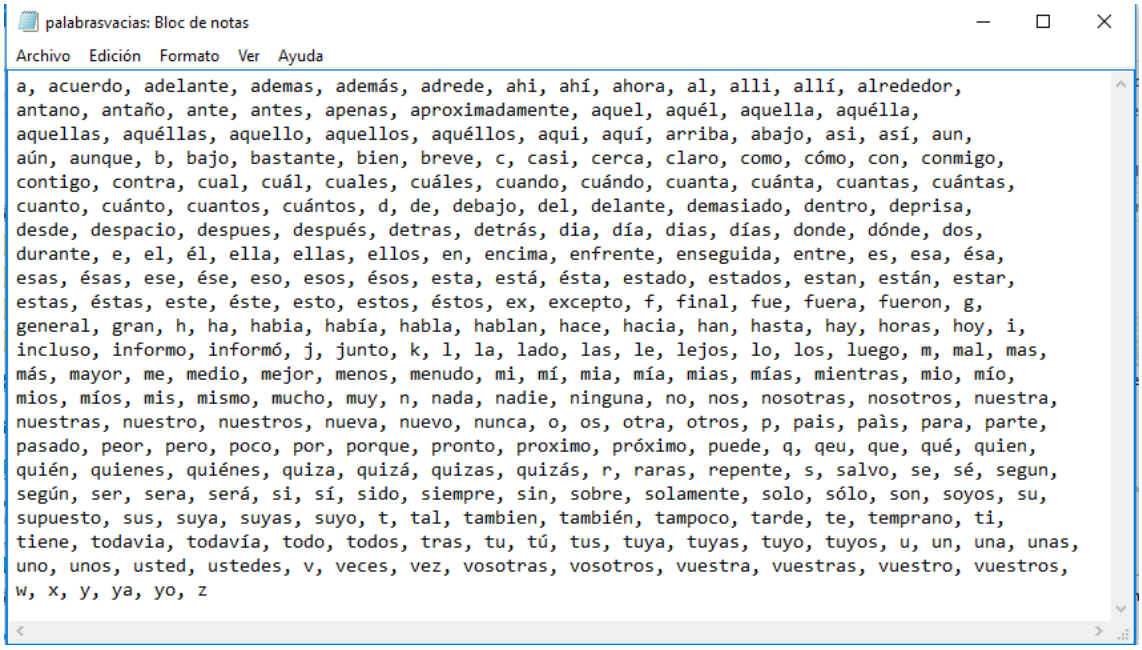

Figura 5.4: Lista de palabras vacías.

Una vez tenemos las palabras vacías, eliminamos estas de nuestro documento:

```
sw <- read_lines("palabrasvacias.txt")
sw = iconv(sw, to="ASCII//TRANSLIT")
d <- tm_map(d, removeWords, sw)
```

Ahora quitamos las palabras vacías genéricas:

```
d <- tm_map(d, removeWords, stopwords("spanish"))
```

Creamos una matriz de términos con los más frecuentes habiendo eliminado todo lo anterior que no nos interesaba:

```
mdt <- TermDocumentMatrix(d)
```

Introduciendo el siguiente comando obtenemos una lista con las palabras que aparecen con frecuencia mayor o igual a 15:

```
findFreqTerms(mdt, lowfreq=15)
```

```
[1] "divertido"   "entretenido"   "bien"   "bueno"   "juego"   "adictivo"
"gran"
[8] "gladiadores"
```

Figura 5.5: Palabras más frecuentes en los comentarios de Gladiadores.

Ahora lo que haremos es cargar nuestra matriz de términos y usando un data frame obtendremos una nube de palabras usando el paquete wordcloud:

```
m <- as.matrix(mdt)
dim(m)
m <- m %>% rowSums() %>% sort(decreasing = TRUE)
m <- data.frame(palabra = names(m), frec = m)
wordcloud(
words = m$palabra,
freq = m$frec,
max.words = 43,
random.order = F,
colors=brewer.pal(name = "Dark2", n = 8))
```

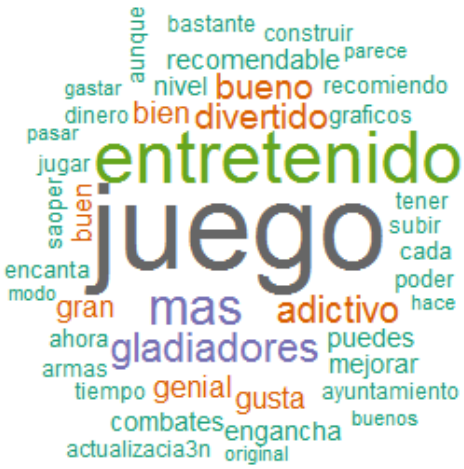

Figura 5.6: Nube de palabras de los comentarios de Gladiadores.

También podemos ver una tabla con la dimensión que queramos, en este caso tomamos 20 palabras, junto a sus frecuencias:

```
m[1:20, ]
```

	palabra	frec
juego	juego	99
entretenido	entretenido	54
mas	mas	33
gladiadores	gladiadores	25
adictivo	adictivo	22
divertido	divertido	20
bueno	bueno	19
bien	bien	16
gran	gran	15
genial	genial	14
gusta	gusta	14
buen	buen	13
recomendable	recomendable	12
nivel	nivel	11
combates	combates	11
puedes	puedes	10
mejorar	mejorar	10
engancha	engancha	10
poder	poder	9
encanta	encanta	9

Figura 5.7: Tabla de palabras más frecuentes con sus frecuencias de Gladiadores.

En forma de gráfico:

```
m[1:10, ] %>%
ggplot(aes(palabra, frec)) +
geom_bar(stat = "identity", color = "black", fill = "#87CEFA"
   ) +
geom_text(aes(hjust = 1.3, label = frec)) +
coord_flip() +
labs(title = "Diez palabras mas frecuentes en reviews
   Gladiadores",  x = "Palabras", y = "Numero de usos")
```

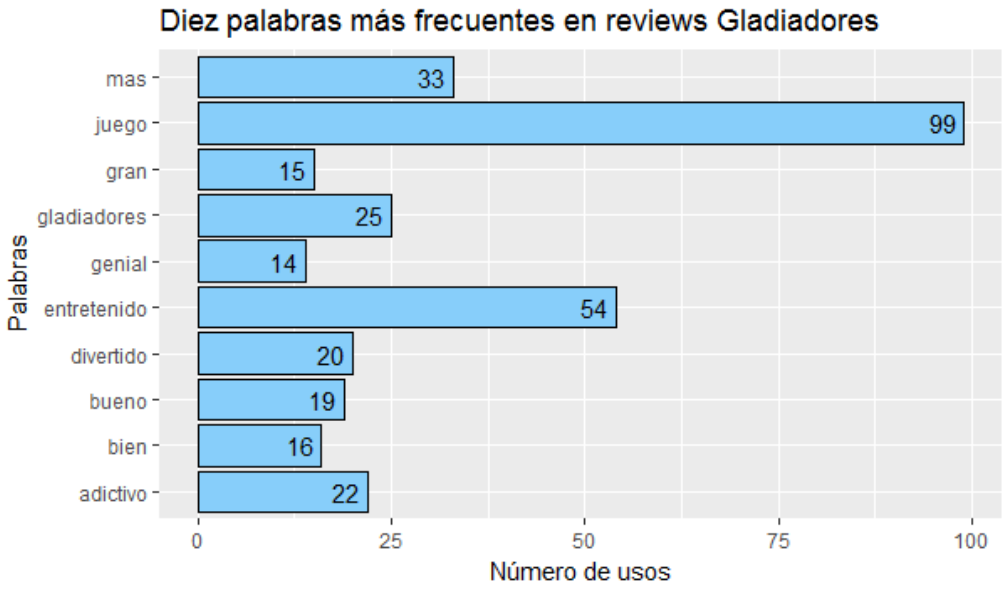

Figura 5.8: Gráfico de palabras más frecuentes con sus frecuencias de Gladiadores.

Otra cosa que resulta de interés a la hora de analizar los sentimientos de los usuarios es ver la relación que tienen ciertas palabras claves con otras. Para ello vamos a ver junto a qué palabras aparecen las palabras "juego", "buen" y "mejorar" y la frecuencia de aparición de las mismas.

```
findAssocs(mdt, terms = c("juego", "buen", "mejorar"),
   corlimit = .25)
```

```
$juego
modo gran
0.27 0.26

$buen
   rato       demas       echar    distraido     semana      trata
   0.35        0.27        0.27        0.27        0.27        0.27
   disfrutando    decirles    gastarse      leen       reseaas      ven
      0.27        0.27        0.27        0.27        0.27        0.27
   currado  facilmente    opciones      estrellitas   jueguecito   podais
    0.27        0.27        0.27           0.27        0.27        0.27
   verde
   0.27

$mejorar
    cogiendo   constructores       cosa       decirlo    dejar     fallo
      0.58          0.58          0.58          0.58      0.58      0.58
    reventar     sola       varias       madera      juegazo
      0.58       0.58         0.58          0.51        0.42
      oro          pide          voy          duda     muchisimo
      0.40         0.40          0.40          0.40        0.40
    wapisimo     solo          mas         puedes        puedo   jugabilidad
      0.40       0.37         0.36         0.34          0.32       0.32
    hacer    acelerarlo       hacerlo        marmol      atienden
     0.31        0.28          0.28          0.28          0.28
    pelear       estrellitas    jueguecito        podais
     0.28          0.28          0.28              0.28
    verde  equipaciones    ofrecer      desafaos         mucha    muchas
     0.28      0.28          0.28          0.28          0.28       0.28
    posibilidades      vida       mejoras        rapido
       0.28            0.28          0.27          0.27
```

Figura 5.9: Relación entre las palabras de interés.

A continuación, realizaremos un análisis de agrupaciones jerárquicas para ver grupos de palabras y las relaciones que tienen entre sí mediante la distancia que existe entre ellos.

Empezamos eliminando los términos dispersos de nuestra matriz de términos (mdt) para obtener únicamente las palabras más frecuentes que nos sirvan para interpretar los comentarios.

```
mdt_new<- removeSparseTerms(mdt, sparse = .95)
```

Vamos a ver cómo ha disminuido la cantidad de palabras de la matriz de términos original:

```
mdt$nrow
```

```
[1] 731
```

```
mdt_new$nrow
```

```
[1] 13
```

Definimos ahora la matriz de distancia por el método de distancia euclídea:

```
mdt_new <- mdt_new %>% as.matrix()
mdt_new <- mdt_new / rowSums(mdt_new)
mdt_dist <- dist(mdt_new, method = "euclidian")
```

De este modo podemos observar los grupos de palabras que existen en reviewsES. Además, podemos ver qué palabras pertenecen a grupos lejanos entre sí, por ejemplo, 'bien'y 'juego':

```
mdt_hclust <-  hclust(mdt_dist, method = "ward.D")
plot(mdt_hclust, main = "Dendrograma de reviewsES Gladiadores
   ", sub = "", xlab = "")
```

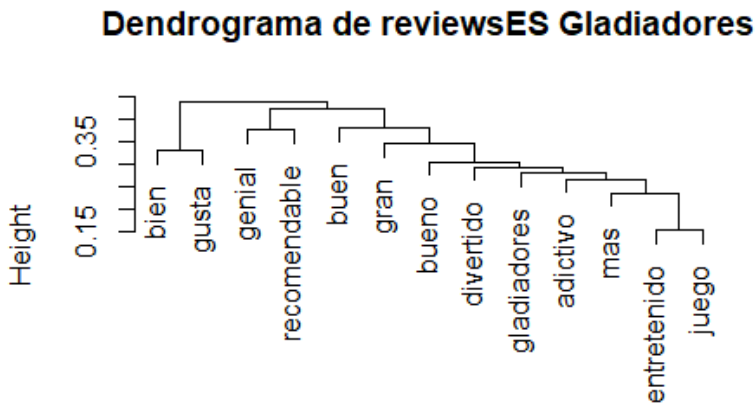

Figura 5.10: Dendograma de grupos de palabras que aparecen juntas.

En general, a lo largo de este estudio, hemos visto que las opiniones sobre el juego son muy buenas. Sin embargo, mirando en profundidad la aparición de ciertas palabras con otras podemos obtener información para mejorar el juego, así como posibles errores en el desarrollo del mismo.

Capítulo 6

Conclusión

Como consecuencia del gran desarrollo que la sociedad ha experimentado a la hora de generar información y la capacidad existente para almacenarla, la minería de textos es cada día más útil, ya que gran parte de la información que se recoge hoy en día es en forma de texto.
En el proyecto se expone que el procesado automático de datos no estructurados resulta una tarea difícil por varias razones. Entre ellas está la gran dimensionalidad; por ello muchas de las técnicas estadísticas tienen entre sus tareas reducir la dimensión de los documentos sin variar el contenido semántico del mismo.
Hemos visto también lo útil que es la minería de textos para cualquier tipo de organización, ya que puede servir para ahorrar dinero y resolver problemas. Además, las conclusiones obtenidas pueden usarse para la toma de decisiones.
Gracias a las técnicas de minería de textos, actualmente se puede analizar y obtener información no solo de datos estructurados, sino que también de textos.
En el ámbito empresarial, mediante la aplicación realizada en el punto anterior, hemos visto una forma rápida de conocer las opiniones de los clientes sin necesidad de realizar encuestas o tener un servicio de atención al cliente. A las empresas les sirve, entre otras cosas, para crear nuevas estrategias comerciales dirigidas a todos los clientes o a un grupo en concreto.
Por último, cabe mencionar el aumento de software dedicados a la minería de textos, tanto sistemas desarrollados exclusivamente para el análisis de datos no estructurados como sistemas que incorporan aplicaciones específicas de minería de textos.

Bibliografía

[1] Álvarez Gálvez, Javier; Plaza, Juan F.; Muñiz, José Antonio y Lozano Delmar, Javier; *Aplicación de técnicas de minería de textos al frame analysis: identificando el encuadre textual de la inmigración en la prensa* , 20 de mayo de 2014

[2] Zhao, Yanchang; *R and Data Mining: Examples and Case Studies*, 26 de Abril de 2013

[3] Miner, G.; Delen, D.; Elder, J.; Fast, A.; Hill, T. y Nisbet, R.; *The Seven Practice Areas of Text Analytics*, Enero, 2012

[4] Expert System; *10 text mining examples*, 18 de Abril de 2016 : **www..com/10-text-mining-examples/**

[5] IBM; *Acerca de IBM SPSS Modeler Text Analytics* : **www.ibm.com**

[6] GOOGLE CLOUD; *API Natural Language de Cloud* : **https://cloud.google.com/natural-language/?hl=es**

[7] Contreras Barrera,Marcial; *Minería de texto:una visión actual* : **www.redalyc.org/html/285/28540279005/**

[8] Botana,Guillermo de Jorge; «Una aproximación distribuida al análisis semántico»,*La técnica del Análisis de la Semántica Latente (LSA/LSI) como modelo informático de la comprensión del texto y el discurso*. Septiembre, 2010.

[9] Román Carrillo,Javier; *Minería de datos y aplicaciones*, **www.it.uc3m.es/jvillena/irc/practicas/06-07/22.pdf**

[10] MICROSOFT; textitConceptos de minería de datos, 2016: **https://msdn.microsoft.com/es-es/library/ms174949.aspx**

[11] Bholat,David; Hansen, Stephen; Santos, Pedro y Schonhardt-Bailey, Cheryl;*Minería de textos para bancos centrales.*

[12] Elkan,Charles; *Text mining and topic models*, **http://cseweb.ucsd.edu/ elkan/250B/topicmodels.pdf**, 12 de Febrero de 2014.

[13] Torres López,Carmen ; *El papel de la Minería de Texto en el Sector de Seguros*, 2 de Octubre de 2014: **www.meaningcloud.com/es/blog/mineria-de-texto-en-sector-de-seguros**

[14] GOOGLE CLOUD; *Hotel chain makes a reservation for the future with Cloud Platform*: **htpps://cloud.google.com/customers/nh-hotels/**

[15] Lozada,Alea; *Minería de textos y sus aplicaciones*: **http://www.semanticwebbuilder.org.mx/es/swb/**

[16] Minería de textos; *Minería de textos. Sistemas Avanzados de Recuperación de la Información*, 7 de Mayo de 2010: **http://mineriadetextos.tripod.com/**

[17] Fernández Ollero,Iván; *Minería de Textos o Text Mining*: **http://textmining.galeon.com/**

[18] Eíto Brun,Ricardo y Senso,Jose A.; «El profesional de la información»,*Minería textual*, 2004: **http://www.elprofesionaldelainformacion.com/contenidos/2004/enero/2.pdf**

[19] Venegas V.,René; *Análisis Semántico Latente: una panorámica de su desarrollo*, 2013: **http://www.scielo.cl/**

[20] Velasco,Guillermo de la Calle; *Modelo basado en técnicas de procesamiento de lenguaje natural para extraer y anotar información de publicaciones científicas*, Abril, 2014, Madrid.

[21] Venkatesan,Rajkumar; *AIRBNB AND A HOST OF DATA*, 31 de Enero de 2017: **https://ideas.darden.virginia.edu/2017/01/airbnb-and-a-host-of-data/**

[22] Sevilla,Julio M.; *TwitteR: aprovecha los datos de las redes sociales*, 31 de Marzo de 2015: **http://www.epidom.es/blog/2015/03/libreria-twitter/**

[23] IBM Security; *Seguridad cognitiva*, Abril de 2016

Anexo - Código en R

A continuación se detalla el código ejecutado en el programa R para el ejemplo LSA del punto 3.2.1.

```
library(NLP)
library(tm)
library(RColorBrewer)
library(readr)
library(Matrix)


# lee el documento UTF-8 y lo convierte a ASCII:

txt <- read_lines("doc.txt")
txt = iconv(txt, to="ASCII//TRANSLIT")

#Creamos el corpus:

corpus <- Corpus(VectorSource(txt))

#Matriz de terminos:

mdt <- TermDocumentMatrix(corpus)

#Renombramos las columnas:

colnames(mdt) <- c("d1", "d2", "d3", "d4", "d5")

#Convertimos la mdt de lista a matriz:

A <- as.matrix(mdt)

AT = t(A)

```

```
#Matriz terminos - terminos, B:

B = A%*%AT

#Matriz documentos - documentos, C:

C = AT%*%A

#Matriz de autovectores y autovalores de B, S:

b <- eigen(B)
S<- eigen(B)$vectors
AutvalB <- eigen(B)$values


#Matriz de autovectores y autovalores de C, U:

c <- eigen(C)
U <- eigen(C)$vectors
UT =t(U)
AutvalC <- eigen(C)$values


#Matriz diagonal cuyos elementos son las raices cuadradas
   de los autovalores de B, P:

v = sqrt(AutvalB)
P <- diag(v)

####k = 2:
#Matriz P:

P2 <-P[1:2, 1:2]

#Matriz S:

S2 <- S[1:18, 1:2]
dimnames(S2) <- list(c("Hamlet" , "Buen" , "Conoci" , "Era
   " , "Horacio" , "Personalmente" , "Rey" , "Divierte" ,
   "Asi" , "Como" , "Fue" , "Principito" , "Encantador" ,
```

```
    "Esta" , "Existido" , "Muchachito" , "Prueba" , "Que"),
    NULL)

#Matriz Ut:

UT2 <- UT[1:2, 1:5]
dimnames(UT2) <- list(NULL, c("d1", "d2", "d3", "d4", "d5"
    ))

#Calculamos por partes para aproximar A:

T1 = S2%*%P2
T2 = P2%*%UT2

#Separamos los vectores por palabras y documentos:

hamlet = T1[1:1,1:2]
buen = T1[2:2,1:2]
conoci = T1[3:3,1:2]
era = T1[4:4,1:2]
horacio = T1[5:5,1:2]
personalmente = T1[6:6,1:2]
rey = T1[7:7,1:2]
divierte = T1[8:8,1:2]
asi = T1[9:9,1:2]
como = T1[10:10,1:2]
fue = T1[10:11,1:2]
principito = T1[10:12,1:2]
encantador = T1[10:13,1:2]
esta = T1[10:14,1:2]
existido = T1[10:15,1:2]
muchachito = T1[10:16,1:2]
prueba = T1[10:17,1:2]
que = T1[10:18,1:2]


d1 = T2[1:2,1:1]
d2 = T2[1:2,2:2]
d3 = T2[1:2,3:3]
d4 = T2[1:2,4:4]
d5 = T2[1:2,5:5]
```

```
106
107 q = (rey + principito)/2
108 Q<-c(q)
109 O <- c(0,0)
110
111 #Unimos todas las matrices y vectores en una matriz que
      llamamos T3:
112
113 T3 <- rbind(T1,Q,t(T2),O)
114
115 #Representacion geometrica de los documentos, terminos y
      la consulta:
116
117 plot(T3[,1:1], T3[,2:2], col = "blue", pch = 16,xlim = c
      (-2.1, 0.2), ylim = c(-1.5,1.7 ))
118
119 text(T3[,1:1] , T3[,2:2], labels = row.names(T3),  pos =
      4, col = "black")
```