UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

TÍTULO: Análisis de los comentarios en español de usuarios de Facebook para la clasificación de publicaciones utilizando técnicas inteligentes

AUTORES: Gianetto, Emiliano Ariel. Saporiti, Lucía

DIRECTOR: Dr. Lic. Hasperué Waldo.

CODIRECTOR:

ASESOR PROFESIONAL:

CARRERA: Licenciatura en Informática.

Resumen

Actualmente, la interacción de las personas mediante redes sociales está crecimiento exponencialmente. Motivo por el cual se optó elegir una de ellas, como nuestra fuente de información, y a partir de la misma poder captar las espontáneas manifestaciones de sentimientos por parte de los usuarios. Los datos en cuestión fueron transformados, utilizando diversas técnicas de Procesamiento del Lenguaje Natural. Posteriormente se realizó el entrenamiento de algoritmos de Machine Learning, con el fin de ser utilizado para el Análisis de Sentimiento, llevando a cabo un estudio comparativo respecto de la performance de los mismos.

Palabras Clave

Procesamiento de Lenguaje Natural, Minería de Datos, Minería de Opinión, Análisis de Sentimiento, Enfoque de aprendizaje automático, Enfoque basado en léxico, Máxima Entropía, Naïve Bayes, Máquinas de Vectores de Soporte, Machine Learning.

Conclusiones

Se pudo cumplir con los objetivos de estudiar y comparar diferentes técnicas de Análisis de Sentimiento y Procesamiento de Lenguaje Natural. Se combinaron estos dos grandes temas en una herramienta capaz de analizar la forma en la que reaccionan los usuarios ante publicaciones, pudiendo focalizar su uso tanto a una empresa que desee hacer un estudio de marca o servicio, como así también a una persona pública que esté interesada en conocer su imagen en la sociedad, cómo es su connotación.

Trabajos Realizados

Investigación y estudio de conceptos, características y elementos del Procesamiento de Lenguaje Natural.
Investigación y estudio de conceptos, características, elementos y técnicas de clasificación.
Análisis comparativo de distintas técnicas existentes de clasificación sobre un corpus etiquetado.
Una aplicación que permite obtener automáticamente contenido de las redes sociales, y a partir de él, aplicar técnicas de Procesamiento de Lenguaje Natural y de Minería de Opinión para realizar tareas de Análisis de Sentimientos.

Trabajos Futuros

- Ver la influencia entre usuarios dentro de un hilo de discusión, y analizar si modifican su opinión.
- Analizar la posibilidad de agrupar usuarios según su valoración sobre los noticias/publicaciones.
- Analizar la posibilidad de extraer información sobre el usuario, y así poder realizar un ranking de noticias/publicaciones más aceptadas o rechazadas para grupos con ciertas características similares.
- Permitir que el usuario que utilice la aplicación pueda reaccionar, de forma similar a Facebook, sobre los comentarios en tiempo real.
- Elegir las publicaciones y comentarios a recolectar según lo aprendido por el modelo.

Fecha de la presentación: Septiembre 2018

Análisis de los comentarios en español de usuarios de Facebook para la clasificación de publicaciones utilizando técnicas inteligentes

Tesina de grado

Alumnos:
Gianetto Emiliano Ariel,
Saporiti Lucía

Director:
Dr. Hasperué Waldo

Universidad Nacional de La Plata
Facultad de Informática

Resumen

Actualmente, la interacción de las personas mediante redes sociales está experimentando un crecimiento exponencial. Motivo por el cual se optó elegir una de ellas como nuestra fuente de información, y a partir de la misma poder captar las espontáneas manifestaciones de sentimientos por parte de los usuarios. Los datos en cuestión serán objeto de posteriores procesamientos aplicando diversas técnicas, a fin de realizar un Análisis de Sentimientos y así obtener información que refleje la connotación positiva o negativa sobre la actitud del redactor.

En este marco, nos propusimos indagar acerca de diversas técnicas de procesamiento del lenguaje natural y clasificación. Posteriormente se procederá a comparar y aplicar las mismas a comentarios realizados en publicaciones de un portal de noticias en la red social Facebook.

Teniendo entonces como objetivo realizar un aporte en español sobre estos temas, dado que la mayoría de la bibliografía existente utiliza el idioma inglés.

Indice general

Capítulo 1

Introducción

1.1 Motivación

Las redes sociales son cada vez más utilizadas, tal es así que poseen millones de usuarios participando día a día en ellas. Esta actividad puede manifestarse de diversas formas, ya sea compartiendo su estado de ánimo, opinando respecto a diversos temas y reaccionando al contenido publicado, entre algunos ejemplos.

Una de las redes sociales más conocidas y utilizadas en la actualidad es Facebook, que cuenta con aproximadamente 2.239 millones de usuarios activos por mes [17], de los cuales 20 millones pertenecen a Argentina. Además, es el país cuyos usuarios pasan más tiempo en Facebook: 9 horas promedio por mes [5]. Esta red social brinda a los usuarios diversas maneras de expresarse, ya sea mediante comentarios extensos, contestando a un comentario en particular, seleccionando la opción “me gusta” simpatizando con esta opinión, o eligiendo una reacción ante una publicación (“me gusta”, “me encanta”, “me divierte”, “me asombra”, “me entristece”, “me enoja”).

Toda esta participación e interacción se traduce en contenido generado por los usuarios (*User Generated Content* - UGC), el cual puede ser analizado para obtener información de interés, ya sea respecto a la opinión de un usuario sobre un producto o tema de debate, o su estado de ánimo debido a algún suceso. Una manera de analizar estas publicaciones es utilizando técnicas del Procesamiento de Lenguaje Natural (*Natural Language Processing* - NLP), que es un campo dentro de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano.

El NLP es objeto de investigación en la actualidad y resulta de gran interés tanto para la comunidad educativa como para las empresas. Al ser un proceso

complejo, abarca un conjunto de tareas extenso, que comprende tanto análisis sintáctico como semántico.

Además de la complejidad que trae aparejada este proceso, el contenido generado por usuarios posee características no deseadas (y con mayor frecuencia en las redes sociales), como pueden ser el uso de: abreviaturas, lunfardo, sarcasmo, ambigüedad, errores de tipeo, siglas, emoticones y emojis. Si bien estos dos últimos no son palabras pertenecientes a algún idioma, podrían considerarse como algo interesante, ya que aplicando cierto procesamiento, es posible extraer información valiosa a partir de ellos, siendo una clara y simple representación de la opinión del usuario.

Así como NLP, la Minería de Datos también genera gran interés en la actualidad debido al crecimiento exponencial que están sufriendo los datos, y la necesidad de obtener información útil y conocimiento que pueda ser utilizado para la creación de diversas aplicaciones. Esta tarea se lleva a cabo mediante herramientas adicionales de análisis, aparte de las que proveen las bases de datos -como consultas y transacciones-, para un análisis más profundo sobre los mismos, permitiendo obtener patrones importantes.

La Minería de Datos es una fase del proceso de Extracción del Conocimiento en Bases de Datos (KDD - *Knowledge Discovery from Databases*). Éste último consiste en una secuencia iterativa de los pasos: limpieza, integración, selección, transformación y minería de datos, evaluación de patrones y presentación de conocimientos [14].

A partir del NLP y junto con algunas técnicas de la Minería de Datos es posible realizar la tarea conocida como el Análisis de Sentimientos (*Sentiment Analysis*), que se define como el proceso computacional de extraer sentimientos a partir de un texto, obteniendo información acerca de la connotación positiva o negativa sobre la actitud del escritor. Los enfoques existentes en el análisis de sentimientos se pueden agrupar en cuatro categorías: localización de palabras clave, afinidad léxica, métodos estadísticos y técnicas a nivel de concepto. Una forma de realizar este análisis es utilizando técnicas de minería de opinión, ya que genera una gran cantidad de datos clasificados, que permite extraer conocimiento mediante el estudio y análisis de los datos con el objetivo de descubrir patrones y generando modelos de predicción.

Como el Procesamiento del Lenguaje Natural, el Análisis de Sentimientos y la Minería de Datos son temas sobre los que se está trabajando mucho en la actualidad, existe material de investigación sobre estos conceptos, pero la mayoría son aplicados sobre textos en inglés y son muy escasas las aplicaciones al español.

Es por este motivo que decidimos realizar nuestra investigación aplicando estos conceptos sobre texto en español (publicaciones en Facebook, con sus respectivos comentarios y reacciones), comparando distintas técnicas de clasificación dentro de la Minería de Opinión, como son los clasificadores lineales y probabilísticos, contribuyendo con la creación de material y presentación de resultados orientado al público hispanoparlante.

1.2 Objetivo

El objetivo general de esta tesina es estudiar y comparar técnicas de Procesamiento de Lenguaje Natural y de clasificación, aplicándolas sobre comentarios realizados por usuarios a publicaciones en la red social Facebook para elaborar un análisis de sentimientos, detectando de esta manera en forma automática, la opinión general de los usuarios sobre dicha publicación y el modo en que las noticias son percibidas en la actualidad.

Los objetivos específicos de esta tesis son:

1) Estudiar y aplicar técnicas de procesamiento de lenguaje natural para detectar las características de las palabras que posean mayor relevancia para la descripción de un comentario.

2) Evaluar la performance de las técnicas de clasificación estudiadas, ya sea utilizando distintas formas de representación para los comentarios, aplicando un filtro por cantidad mínima de tokens y/o modificando el número de iteraciones para la validación cruzada, con el fin plantear diversos escenarios de prueba.

3) Implementar un prototipo que, a partir de la opinión del usuario sobre comentarios, permita realizar un filtrado automático y muestre solo aquellos que puedan resultar de su interés.

Cambio de rumbo

Al principio de esta tesis se tenía como objetivo utilizar las reacciones que los usuarios de Facebook tenían sobre una publicación (“me gusta”, “me encanta”, “me divierte”, “me asombra”, “me entristece”, “me enoja”), de esta manera se hubiera utilizado la reacción de un usuario junto con el texto de su comentario para conocer la polaridad del texto. Esta información hubiera sido de gran utilidad, proporcionando una categoría precisa para el sentimiento del usuario acerca de la publicación y poder así complementar el análisis del texto.

A raíz de los cambios realizados por Facebook en su política de permisos sobre la información disponible a través de su API (los cuales se encuentran detallados en la documentación oficial en [28]), al no poder obtener la identificación de quién generó una reacción en una publicación, nos vimos imposibilitados de poder asociar dicho usuario a su comentario, si es que hubiera realizado alguno dentro de la misma publicación.

A su vez, tampoco pudimos obtener las reacciones de cada comentario, sino únicamente los *“me gusta”*. Nuestra idea original era realizar un cálculo sobre las mismas, identificando cuál fue la polaridad predominante de cada uno. Dicha información hubiera sido comparada posteriormente con el Análisis de Sentimiento realizado sobre el texto, a fin de evaluar el grado de coincidencias entre los mismos. Al no poder contar con estos datos, nos vimos obligados a realizar la polaridad manualmente, etiquetando de manera positiva, neutral o negativa los comentarios recolectados, creando un conjunto de comentarios etiquetados *ad-hoc* para esta tesina. Por lo tanto, los resultados alcanzados en esta tesis dependen de la opinión propia de los tesistas, aportando, como resultado final de este trabajo, una aplicación que puede ser personalizada por la persona que la utilice.

1.3 Contribuciones

La presente tesina pretende contribuir con:

- Los resultados de la investigación y estudio de conceptos, características y elementos del Procesamiento de Lenguaje Natural.

- Los resultados de la investigación y estudio de conceptos, características, elementos y técnicas de clasificación.
- Los resultados obtenidos en un análisis comparativo de distintas técnicas existentes de clasificación sobre los comentarios etiquetados.
- Una aplicación que permite obtener automáticamente comentarios realizados por usuarios y, a partir de él, aplicar técnicas de procesamiento de lenguaje natural y de minería de datos para realizar tareas de análisis de sentimiento.

1.4 Organización del documento

Este capítulo hizo una breve introducción, presentando el tema general de la Tesis, su motivación, objetivo y contribuciones.

En el capítulo 2 se expone el concepto abordado de redes sociales, su utilidad e importancia en la sociedad, el contenido que se puede extraer para realizar nuestro trabajo y finalmente se describe la elegida, Facebook.

El capítulo 3 introduce el Procesamiento de Lenguaje Natural, sus diferentes análisis y las posibles técnicas a utilizar con el objetivo de generar una entrada estructurada para nuestro estudio.

En el capítulo 4, se presenta la Minería de Datos, haciendo hincapié en el Análisis de Sentimientos y la Minería de Opinión, sus diferentes enfoques, técnicas, y las métricas que pueden ser obtenidas.

El capítulo 5 comienza con una breve introducción acerca del desarrollo propuesto. Luego se describe el servicio utilizado para la obtención del contenido, continuando con un análisis de la arquitectura planteada y las tecnologías aplicadas. Por último se describen con mayor nivel de detalle cada una de las etapas del procesamiento realizado, con el fin de poner en práctica todo el marco teórico estudiado.

En el capítulo 6, se puntualizan los diferentes escenarios que utilizamos para aplicar las técnicas, y así poder realizar una exhaustiva comparación de las mismas, teniendo en cuenta las métricas antes mencionadas.

En el capítulo 7, culminamos el trabajo con las conclusiones a las que llegamos, haciendo un repaso general de la investigación realizada. Además, enumeramos posibles trabajos futuros que pueden surgir a partir del presente ensayo, profundizando en aquellos análisis que podrían ser apropiados para perfeccionar nuestro desarrollo pero que no formaron parte de nuestro alcance.

En el final del documento se encuentra la bibliografía a la que recurrimos.

Capítulo 2

Redes Sociales

2.1 ¿Qué son las redes sociales?

La **red social** es un concepto creado en la comunicación, que hace referencia al conjunto de grupos, comunidades y organizaciones vinculados unos a otros a través de relaciones sociales (relación profesional, amistad, parentesco, etcétera), y que tienen como fin la interacción de dos o más actores.

Las plataformas de Internet que facilitan la comunicación entre personas de una misma estructura social se denominan servicios de red social o redes sociales virtuales. En ellas las personas interactúan a través de perfiles creados por ellos mismos, en los que comparten sus fotos, videos, historias, eventos o pensamientos.

Un **servicio de red social** es un medio de comunicación, que permite reunir a las personas a través de un sitio o aplicación web, para hablar, compartir ideas, hacer nuevos amigos y socializar. Está conformado por un conjunto de entidades (equipos, servidores, programas, conductores, etcétera), y sobre todo por individuos que establecen alguna relación, principalmente de amistad, y que mantienen intereses y actividades en común o se encuentran interesados en explorar los intereses y las actividades de otros usuarios. En este trabajo, nos interesa abordar este concepto y nos referiremos al mismo, a partir de aquí, como red social.

Hay dos desafíos asociados con la conceptualización de las redes sociales. En primer lugar, la velocidad a la que la tecnología se está expandiendo y evolucionando, impide una definición clara del propio concepto. Las tecnologías de medios sociales incluyen una amplia gama de PC y plataformas basadas en dispositivos móviles que se continúan desarrollando, lanzando, relanzando, abandonando e ignorando todos los días en países de

todo el mundo y en distintos niveles de conciencia pública. En segundo lugar, los servicios de redes sociales facilitan diversas formas de comunicación que son similares a las habilitadas por otras tecnologías como teléfono, correo electrónico, etcétera.

Las redes sociales se han convertido, en pocos años, en un fenómeno global. Si bien las redes sociales, en términos sociológicos, han continuado casi tanto como las propias sociedades han existido, el potencial de la Web para facilitar la interacción social ha llevado a una expansión exponencial y en curso de ese fenómeno. Esto y los bajos costos de almacenamiento de datos en línea hicieron posible, por primera vez, ofrecer a las masas de internautas acceso a una serie de espacios centrados en el usuario que podrían llenar con contenido generado por el usuario, junto con un conjunto diverso de oportunidades para vincular estos espacios juntos para formar redes sociales virtuales. Además en los últimos años, el tipo de contenido disponible en la Web, se ha transformado. Desde principios de la década de 1990 en adelante, la mayoría del contenido en Internet era publicado por unos pocos y consumido por muchos usuarios, similar a los medios tradicionales. Desde principios de la década de 2000, el contenido generado por los usuarios se ha vuelto cada vez más popular en la web: cada vez más usuarios participan en la creación de contenido, en lugar de solo consumirlo.

Una diferencia importante entre el material generado por los usuarios y el material tradicional, que es particularmente significativo para los medios basados en el conocimiento, como los portales de preguntas y respuestas, es la variación en la calidad del contenido. El principal conflicto que plantea el contenido en los sitios de redes sociales es el hecho de que la distribución de la calidad es alevosamente despareja: desde contenido sumamente relevante hasta contenidos que no aportan absolutamente nada (url, spam, etcétera). Esto hace que las tareas de filtrado y clasificación en dichos sistemas sean más complejas que en otros ámbitos. Sin embargo, para las tareas de recuperación de información, las redes sociales presentan ventajas inherentes sobre las colecciones tradicionales de documentos: su estructura ofrece más datos disponibles que en otros dominios. Además del contenido del documento y la estructura del enlace, las redes sociales exhiben una gran variedad de tipos de relaciones usuario-documento y de usuario a usuario.

2.2 ¿Cuál es su función y qué utilidad pueden tener?

Las redes sociales ayudan a las personas a mantenerse conectadas con sus amigos y familiares y son una manera fácil de encontrar lo que cada uno hace cada día en su círculo social. Para identificar las funcionalidades básicas de las redes sociales, partimos de su definición, entonces teóricamente se desprenden dos categorías (1) mantenerse en contacto y (2) gestión de identidad.

(1) Mantenerse en contacto puede dividirse en comunicación directa (intercambio directo con alguien) y comunicación indirecta (mediante artefactos) de acuerdo con las teorías de comunicación. En el contexto de la comunicación indirecta, existe una necesidad de gestión de contactos, es decir, para definir filtros de quién podrá obtener información sobre las actividades de uno (control de acceso) y de quién quiere ver información.

(2) El campo de la gestión de la identidad se puede especificar con más detalle sobre las razones para presentarse, para encontrarse, para construir un contexto común más rápidamente, habilitando a los demás y generar información para la comunicación indirecta. Esto también se puede ver desde el otro lado: encontrar a alguien, construir un contexto común (ver si uno tiene algo en común con el otro) o mantenerse informado acerca del otro (mediante comunicación indirecta). Debido a que las personas son observadas y analizadas por otros, construyen una identidad social que presentan a otros. En las redes sociales, el perfil que las personas construyen es esta presentación de uno mismo: para un público o tarea en particular. Por lo tanto, en nuestro contexto, la gestión de identidades significa administrar la información de identidad y establecer los derechos de acceso, a quién se le permite ver qué. Los derechos de acceso pueden ser directos o basados en roles, permitiendo el acceso de todos los usuarios en la red personal. Los ejemplos de funciones que permiten la administración de identidades en las redes sociales son: perfil de usuario y membresías de grupo.

Se pueden encontrar otras funcionalidades comunes en las diferentes redes sociales:

- Búsqueda avanzada: En este contexto, uno tiene que distinguir entre la

posibilidad de buscar en la red de acuerdo con diferentes criterios (nombre, intereses, empresa) y la posibilidad de recibir recomendaciones proactivamente de contactos interesantes por parte de las redes sociales. Los ejemplos de funciones que permiten la búsqueda avanzada en redes sociales son: cuadros de búsqueda.

- Conciencia contextual: Se refiere a conocer el contexto común con otras personas. Puede tratarse de información sobre contactos, intereses o lugares comunes (colegio, universidad, empresa, etcétera). La conciencia contextual contribuye mucho a crear una confianza común entre los usuarios. Ejemplos de funciones que permiten la conciencia contextual en redes sociales son: los *boxes* "Cómo estás conectado a ...".
- Gestión de contactos: La gestión de contactos combina todas las funcionalidades que permiten el mantenimiento de la red personal digital. Los ejemplos de funciones que permiten la administración de contactos en redes sociales son: etiquetar personas y restricciones de acceso al perfil.
- Conciencia de la red: El conocimiento de las actividades y/o el estado actual, y los cambios de este último, de los contactos en la red personal también es respaldado por funcionalidades. Estas funcionalidades permiten la comunicación indirecta a través del conocimiento. Ejemplos de funciones que permiten la concientización de redes en las redes sociales son: noticias y cumpleaños.
- Intercambio: Se combinan todas las posibilidades para intercambiar información directamente (mensajes) o indirectamente (fotos o mensajes a través de tableros de anuncios). Los ejemplos de funciones que permiten el intercambio en redes sociales son: mensajes y álbumes de fotos.

Los sitios y aplicaciones de redes sociales más populares del mundo ciertamente han cambiado a lo largo de los años, y continuarán cambiando, agregando más funcionalidades y permitiendo que interactúen otros tipos de actores. Como actualmente ocurre, las redes sociales están siendo utilizadas no solamente para publicitar, promocionar o vender cierto producto o negocio, sino que son aprovechadas para que cada uno exponga su talento o trabajo,

como hoy en día, son los llamados *youtubers*, *instagramers*, etcétera. De esta forma, brindan más oportunidades al llegar masivamente a ciertos tipos de usuarios.

En lo que respecta a nuestro trabajo, nos interesa la utilidad que tienen las redes sociales en cuanto a informar la actualidad. Existen diferentes portales de noticias en varias redes sociales, lo que provoca un consumo incidental de las mismas y que sea la forma principal de informarse en muchas sociedades modernas. Se consume noticias sin buscarlas: la información sobre la actualidad nos encuentra en nuestras cuentas de Twitter, WhatsApp, Facebook, Instagram, Snapchat, etcétera [25]. Así mismo, las redes sociales permiten opinar sobre las noticias, generando un gran contenido e interesante intercambio entre los diferentes usuarios.

2.3 Ejemplos de redes sociales más utilizadas

Generalistas u horizontales

Permiten la libre participación, centrándose en los contactos, y no a una clase específica de usuario o un tópico concreto. El objetivo de los usuarios al acceder a las mismas, es la interrelación general y su función principal es la de relacionar personas a través de las herramientas que ofrecen, y poseen las siguientes características: crear un perfil, compartir contenidos y generar listas de contactos. Ejemplos de ellas son: Facebook, Twitter, Google+, MySpace, Tuenti y Badoo.

Cerradas

Son aquellas que se utilizan para compartir archivos en diferentes formatos, como pueden ser YouTube, SlideShare, Snips y Flickr.

Temáticas o verticales

Son aquellas que abarcan un público determinado, o sea que son especializadas. La motivación de los usuarios a acudir a ellas es un interés en común.

- Profesionales: su objetivo es establecer un nexo entre distintos profesionales. A través de ellas se puede compartir información sobre una especialidad concreta, desde estudios, capacitaciones hasta experiencias, originando relaciones laborales, por ejemplo LinkedIn o blogs temáticos.
- De ocio: su fin es reunir a usuarios interesados en actividades de entretenimiento como deportes, música o videojuegos, por ejemplo Wipley (videojuegos) o Dogster (perros).
- Mixtas: son una combinación entre las dos anteriores, proporcionando al usuario un lugar concreto donde desarrollar actividades profesionales y personales, por ejemplo Unience (red social de bolsa y mercados).

Por tipo de conexión

- Simétricas: dos usuarios deben aceptarse mutuamente, es decir, que se deben realizar acciones desde ambos lados para poder establecer este nexo, que sean amigos, por ejemplo Facebook.
- Asimétricas: Un usuario puede seguir a otro, el cual puede optar por seguir o no al primero, por ejemplo Twitter, Google+, Instagram.

En función del sujeto

- Humanas: están orientadas a la interacción entre personas según sus gustos, intereses y actividades en general, por ejemplo Dopplr y Tuenti.
- De contenido: el centro de interés es en el contenido de lo que se publica, o sea que dependen del tipo de archivos a los que tengan acceso los usuarios. Por ejemplo Flickr, Instagram, YouTube, Pinterest y Vimeo.

En función de la localización geográfica

- Sedentarias: son aquellas que se modifican según los contenidos, relaciones, eventos, etcétera, por ejemplo Blogger y Wordpress.
- Nómadas: similares a las redes sociales sedentarias, se les suma un nuevo elemento basado en la ubicación geográfica del usuario, cambian de acuerdo a la cercanía existente entre los integrantes o los lugares visitados, por ejemplo Google Latitude y Fire Eagle.

RED SOCIAL	**TIPO**	**Nº USUARIOS (millones)**
Facebook (*www.facebook.com*)	General	2239
Twitter (www.twitter.com)	Mensajería	2228
Instagram (www.instagram.com)	Foto/Vídeo/Mensajería	1890
YouTube (www.youtube.com)	Videos	1800
Whatsapp (www.whatsapp.com)	Mensajería	1800
Google+ (www.plus.google.com)	General	343
Tagged (http://www.tagged.com)	General	300
Line (line.me/es/)	Mensajería	300
Habbo (www.habbo.com)	General	250
Tumblr (www.tumblr.com)	General	200
SoundCloud (www.soundcloud.com)	Música	200
Hi5 (www.hi5.com)	General	200
Badoo (www.badoo.com)	Contactos	200
Snapchat (www.snapchat.com)	Mensajería	178
NetlogTWOO (www.twoo.com)	General	115
Daily Motion (www.dailymotion.com)	Foto/Video	115
VK (www.vk.com)	General	100
Telegram (www.telegram.org)	Mensajería	100
Soundhound (www.soundhound.com)	Música	100
Pinterest (www.pinterest.com)	Foto/Video	100
Spotify (www.spotify.com)	Música	90
Match (www.match.com)	Contactos	90
Flickr (www.flickr.com)	Foto/Video	90
Slideshare (www.slideshare.net)	Foto/Video	85
Reddit (www.reddit.com)	Blog/Foro	70

Tabla 1. Tabla de ranking correspondiente al año 2017. [16] [17].

2.4 Facebook

Facebook es una compañía estadounidense que ofrece servicios de redes y medios sociales en Internet. Su sitio web fue lanzado el 4 de febrero de 2004 por Mark Zuckerberg, junto con otros estudiantes de la Universidad de Harvard, Eduardo Saverin, Andrew McCollum, Dustin Moskovitz y Chris Hughes. Está disponible en español desde febrero de 2008, extendiéndose a los países de Latinoamérica y a España. En esta época, Facebook ya contaba con más de 2.167 usuarios activos. A fecha de marzo de 2018, cuenta con más de 2200 millones de usuarios activos.

Inicialmente, los fundadores limitaron la membresía del sitio web a los estudiantes de Harvard, pero posteriormente lo ampliaron a instituciones de educación superior en el área de Boston, en las escuelas de la Ivy League y en la Universidad de Stanford. Asimismo, gradualmente agregó soporte para estudiantes en varias otras universidades, y finalmente a estudiantes de secundaria. Desde 2006, a cualquier persona que diga tener al menos 13 años se le ha permitido convertirse en usuario registrado de Facebook, aunque existen variaciones en este requisito según las leyes locales. El nombre proviene de los directorios de fotos personales que a menudo se entregan a estudiantes universitarios estadounidenses.

Se puede acceder a Facebook desde una amplia gama de dispositivos con conexión a Internet, como computadora personal (PC), portátiles, *tablet* y teléfonos inteligentes. Una vez registrados, los usuarios pueden crear un perfil personalizado que indique su nombre, ocupación, escuelas atendidas, etcétera. Este mismo registro, se puede utilizar para asociar cuentas en otras redes sociales como Instagram y Spotify. Los usuarios pueden agregar a otros usuarios como «amigos», intercambiar mensajes, publicar actualizaciones de estado, compartir fotos, vídeos y enlaces, usar varias aplicaciones de software y recibir notificaciones de la actividad de otros usuarios. Además, los usuarios pueden unirse a grupos de usuarios de interés común organizados por lugar de trabajo, escuela, pasatiempos u otros temas, y categorizar a sus amigos en listas como «Personas del trabajo» o «Amigos cercanos». También proporciona opciones para reportar o bloquear a personas o amistades no

deseadas.

Es la red social más popular en Internet, aumentó en 60 millones de usuarios activos mensuales de 1.940 millones en marzo de 2017 a 2.20 mil millones al 31 de marzo de 2018. La tasa de crecimiento parece continuar en 20 millones de usuarios activos por mes [24].

Servicios

Muro: el muro es un espacio en cada perfil de usuario que permite que los amigos escriban mensajes para que el usuario los vea. Solo es visible para usuarios registrados. Permite ingresar imágenes y poner cualquier tipo de logotipos en la publicación. Una mejora llamada supermuro permite incrustar animaciones flash, etcétera.

Biografía: En noviembre de 2011, Mark Zuckerberg anunciaba una nueva presentación para Facebook, se trata de la *Biografía*, que reemplazaría al *Muro*. Se publicó en diciembre del mismo año, y tiene como objetivo agilizar y optimizar el paseo de los usuarios por los perfiles de todos los contactos. Contiene algunas mejoras, como por ejemplo, fecha exacta de publicaciones, actualizaciones de estado, comentarios, etcétera, y brinda la posibilidad de llegar a ellas casi de inmediato, así tengan mucho tiempo. Permite agregar una foto de portada adicional en la parte superior del perfil de la persona (cabe mencionar que esta es visible para todo el mundo, y no existe la posibilidad de cambiar la privacidad), mantiene ordenadas y organizadas las actividades de la persona: Lista de amigos, *Me gusta* en las páginas seleccionadas por el usuario, información personal, suscripciones, etcétera; también es posible agregar eventos que pasaron antes que el usuario se registrara en Facebook. El 30 de marzo de 2012, los organismos de Facebook implementaron la Biografía para las páginas.

Actualmente Facebook ofrece una variedad de servicios a los usuarios y ofrece los que se mencionan a continuación:

- **Lista de amigos**: En ella, el usuario puede agregar a cualquier persona que conozca y esté registrada, siempre que acepte su invitación. En Facebook se pueden localizar amigos con quienes se perdió el contacto o agregar otros nuevos con quienes intercambiar fotos o mensajes. Para

ello, el servidor de Facebook posee herramientas de búsqueda y de sugerencia de amigos.

- **Chat**: Servicio de mensajería instantánea en dispositivos móviles y computadores a través de Facebook Messenger.
- **Grupos y páginas**: Es una de las utilidades de mayor desarrollo reciente. Se trata de reunir personas con intereses comunes o que puedan recurrir a ellos en búsqueda de algo puntual. En los grupos se pueden añadir fotos, vídeos, mensajes, etcétera. Tienen su normativa, entre la cual se incluye la prohibición de grupos con temáticas discriminatorias o que inciten al odio y falten al respeto y la honra de las personas; Si bien esto no se cumple en muchas ocasiones, existe la opción de denunciar y reportar los grupos que van contra esta regla, por lo cual Facebook incluye un enlace en cada grupo el cual se dirige hacia un cuadro de reclamos y quejas. Los grupos pueden ser públicos (cualquiera puede buscarlo, ver los miembros y lo que se publica), cerrado (cualquier persona puede buscar el grupo pero solo los miembros pueden ver quién pertenece y los contenidos del mismo) o secreto (solo los miembros pueden verlo, junto con quien pertenece y las publicaciones). Por el contrario, las páginas son públicas, cualquiera puede acceder a ellas y su contenido. A diferencia de los grupos no contienen un histórico de archivos, ya que su administrador es el único que puede realizar publicaciones y, generalmente, están orientadas hacia marcas, personajes específicos o, lo que es de nuestro interés, portales de noticias.
- **Fotos y videos**: Permite publicar nuestras fotos o videos a nuestros “amigos” o compartir los de otros usuarios.
- **Botón «Me gusta»**: Esta función aparece en la parte inferior de cada publicación o comentario hechos por el usuario o sus contactos (actualizaciones de estado, contenido compartido, etcétera). Se caracteriza por un pequeño ícono en forma de una mano con el dedo pulgar hacia arriba. Permite valorar si el contenido es del agrado del usuario actual en la red social, del mismo modo se notifica a la persona que expuso ese tema originalmente si es del agrado del alguien más (alguno de sus contactos). Si se posiciona el cursor sobre este botón,

luego de unos segundos le aparece las llamadas “reacciones”, permitiendole al usuario elegir el nivel de su agrado agregando las opciones “Me encanta”, “Me divierte”, “Me asombra”, “Me entristece” y “Me enoja”.

- **Aplicaciones**: Son pequeñas aplicaciones con las que se puede conocer la galleta de la suerte de uno como usuario, quien es el usuario considerado como mejor amigo, descubrir cosas de la personalidad de uno mismo, etcétera.
- **Juegos**: la mayoría de aplicaciones encontradas en Facebook se relacionan con juegos de rol, juegos parecidos al Trivial Pursuit (por ejemplo geografía), o pruebas de habilidades (digitación, memoria).

Capítulo 3

Procesamiento del Lenguaje Natural

3.1 Introducción

Si bien el Procesamiento del Lenguaje Natural, NLP en adelante, no es una nueva ciencia, la tecnología avanza rápidamente gracias a un mayor interés en las comunicaciones entre personas, además de la disponibilidad de *big data*, la informática potente y los algoritmos mejorados. Entonces, NLP es un campo en continuo desarrollo de la ciencia de la computación, inteligencia artificial y lingüística que ayuda a las computadoras a comprender, interpretar y manipular el lenguaje humano. Por ejemplo, NLP hace posible que las computadoras lean el texto, escuchen el habla, lo interpreten, midan el sentimiento y determinen qué partes son importantes. Tiene como objetivo principal hacer posible la comprensión y el procesamiento asistido de información para determinadas tareas, como son la corrección, resumen y traducción automáticos, y el análisis de sentimientos.

Una persona puede hablar y escribir en inglés, español, etcétera. Pero el idioma nativo de una computadora, conocido como código de máquina o lenguaje de máquina, es en gran medida incomprensible para la mayoría de las personas. En los niveles más bajos de un dispositivo, la comunicación ocurre no con palabras sino a través de millones de ceros y unos que producen acciones lógicas. El NLP se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la comunicación entre personas y máquinas por medio de lenguajes naturales, es decir, que se pueda realizar por medio de programas que ejecuten o simulen la comunicación. El uso de técnicas computacionales, procedentes especialmente de la inteligencia artificial, no aportaría soluciones adecuadas sin una concepción profunda del fenómeno lingüístico. Los modelos aplicados se enfocan no solo a la comprensión del lenguaje, sino a aspectos generales cognitivos humanos y a la

organización de la memoria. Hasta la década de 1980, la mayoría de los sistemas de NLP se basaban en un complejo conjunto de reglas diseñadas a mano. A partir de finales de 1980, sin embargo, hubo una revolución en NLP con la introducción de algoritmos de aprendizaje automático para el procesamiento del lenguaje.

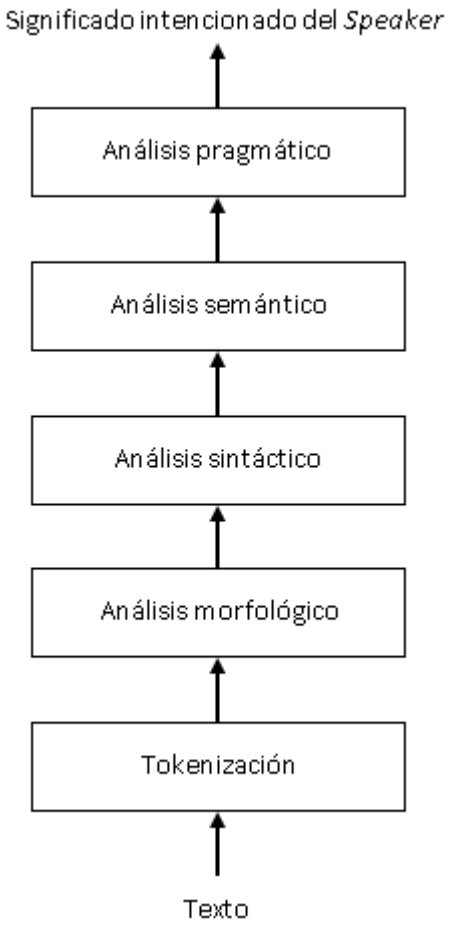


Figura 1. Las etapas de análisis en el procesamiento del lenguaje natural. [26]

El trabajo en NLP ha tendido a descomponer el proceso del análisis del lenguaje en varias etapas, reflejando las distinciones lingüísticas teóricas entre *sintaxis*, *semántica* y *pragmática*. La vista simple es que las oraciones de un texto se analizan primero en términos de su sintaxis; esto proporciona un orden y una estructura que es más susceptible a un análisis en términos de semántica o significado literal; y esto es seguido por una etapa de análisis pragmático por el cual se determina el significado del enunciado o texto en contexto. Esta última etapa se considera a menudo como una cuestión de *DISCURSO*, mientras que las dos anteriores generalmente se refieren a

cuestiones orales. Este intento de correlación entre una distinción estratificante (sintaxis, semántica y pragmática) y una distinción en términos de granularidad (oración versus discurso) a veces causa cierta confusión al pensar acerca de los problemas involucrados en el procesamiento del lenguaje natural; y es ampliamente reconocido que en términos reales no es tan fácil separar el procesamiento del lenguaje en cajas que corresponden a cada uno de los estratos. Sin embargo, dicha separación constituye la base de los modelos arquitectónicos que hacen que la tarea del análisis del lenguaje natural sea más manejable desde el punto de vista de la ingeniería del software. No obstante, la distinción tripartita en sintaxis, semántica y pragmática solo sirve, en el mejor de los casos, como punto de partida cuando consideramos el procesamiento de texto en lenguaje natural real. Una descomposición más fina del proceso es útil cuando tomamos en cuenta el estado actual de la técnica en combinación con la necesidad de tratar con datos de lenguaje real; esto se refleja en la Figura 1 [26].

Identificamos aquí la etapa de tokenización y la segmentación de oraciones como un primer paso crucial. El texto en lenguaje natural generalmente no está compuesto por oraciones cortas, ordenadas, bien formadas y bien delimitadas que encontramos en los libros de texto. También tratamos el análisis léxico como un paso separado en el proceso.

3.2 Preprocesamiento de texto

Antes de comenzar con el análisis propiamente dicho, es necesario realizar un preprocesamiento del texto, a fin de mejorar sus condiciones y mejorar la efectividad de los procesos futuros. Esto se debe principalmente a que, el texto redactado por humanos, suele contener imperfecciones, errores, abreviaturas, jerga, o simplemente datos que no nos interesan. Además, existen problemas de segmentación y tokenización en lenguajes aparentemente fáciles de segmentar, como el español. Fundamentalmente, la cuestión es qué constituye una palabra. En esta sección, hacemos referencia a la tarea de convertir un archivo de texto sin procesar, en una secuencia bien definida de unidades

lingüísticamente significativas: en el nivel más bajo, los caracteres representan los grafemas[1] individuales en el sistema escrito de un idioma, las palabras que constan de uno o más caracteres y las oraciones que consisten en una o más palabras. El preprocesamiento de texto es una parte esencial de cualquier sistema NLP, ya que los caracteres, palabras y oraciones identificados son las unidades fundamentales, las cuales se utilizan en etapas de procesamiento posteriores. Abarcando desde componentes de análisis y etiquetado, como por ejemplo algoritmos analizadores morfológicos, utilizadas por aplicaciones, tales como recuperación de información y sistemas de traducción automática.

El preprocesamiento de texto se puede dividir en dos etapas: clasificación de documentos y segmentación de texto. Los corpus[2] actuales cosechados en Internet pueden abarcar miles de millones de palabras por día, lo que requiere un proceso de clasificación de documentos totalmente automatizado. Este proceso puede implicar varios pasos, según el origen de los archivos que se procesan. En primer lugar, para que un documento en lenguaje natural sea legible por máquina, sus caracteres deben estar representados en una codificación de caracteres, en la que uno o más bytes en un archivo se correlacionan con un carácter conocido. La identificación de codificación de caracteres determina la codificación de caracteres para cualquier archivo. En segundo lugar, para saber qué algoritmos específicos del idioma aplicar a un documento, la identificación del lenguaje determina el lenguaje natural de un documento. Finalmente, el corte de texto identifica el contenido real dentro de un archivo mientras se descarta elementos indeseables, como imágenes, tablas, encabezados, enlaces y formato HTML. El resultado de la etapa de clasificación de documentos es un corpus de texto bien definido, organizado por lenguaje, adecuado para la segmentación de texto y análisis posteriores.

La segmentación de texto es el proceso de convertir un corpus de texto bien definido en las palabras y oraciones que lo componen. La segmentación de palabras divide la secuencia de caracteres en un texto al ubicar los límites de las palabras, los puntos donde termina una palabra y comienza otra. Para propósitos de lingüística computacional, las palabras así identificadas se conocen como *tokens*, y la segmentación de palabras también se conoce como

[1] Unidad mínima e indivisible de la escritura de una lengua.
[2] Conjunto cerrado de textos o de datos destinado a la investigación científica.

tokenización. La segmentación de oraciones es el proceso de dividir un texto en oraciones para su posterior procesamiento. Una oración puede ser vista como una unidad de procesamiento compuesta por una o más palabras.

3.2.1 Tokenización

Tokenización es el proceso de convertir una secuencia de caracteres en una secuencia de tokens. Los tokens resultantes se pasan a otra forma de procesamiento. Algunos métodos utilizados para identificar tokens incluyen: expresiones regulares, secuencias específicas de caracteres denominadas *flags*, caracteres de separación específicos llamados *delimitadores* y definición explícita por un diccionario. Los caracteres especiales, incluidos los signos de puntuación, son comúnmente utilizados para identificar tokens debido a su uso natural en el lenguaje.

Los lenguajes de programación a menudo clasifican los tokens como identificadores, operadores, símbolos de agrupación o por tipo de datos. Los lenguajes escritos comúnmente categorizan tokens como sustantivos, verbos, adjetivos o signos de puntuación. Las categorías se usan para el procesamiento posterior de los tokens.

Después de tokenizar un texto, éste se analiza y los datos interpretados pueden cargarse en estructuras de datos para uso general, interpretación o compilación. Los desafíos específicos planteados por la tokenización dependen en gran medida tanto del sistema de escritura como de la estructura tipográfica de las palabras. Hay tres categorías principales en las que se pueden colocar estructuras de palabras. La morfología de las palabras en un idioma puede ser aislante, donde las palabras no se dividen en unidades más pequeñas; aglutinante, donde las palabras se dividen en unidades más pequeñas (morfemas) con límites claros entre los morfemas; o inflexión, donde los límites entre los morfemas no son claros y donde los morfemas componentes pueden expresar más de un significado gramatical.

3.2.2 Segmentación de oraciones

Las oraciones en la mayoría de los idiomas escritos están delimitadas por signos de puntuación, pero las reglas de uso específicas para la puntuación no siempre están definidas de forma coherente. Incluso cuando existe un conjunto estricto de reglas, el cumplimiento de las reglas puede variar dramáticamente en función del origen de la fuente de texto y el tipo de texto. Por lo tanto, la segmentación de oraciones exitosa para un idioma determinado requiere una comprensión de los diversos usos de los caracteres de puntuación en ese idioma.

En la mayoría de los lenguajes, el problema de la segmentación de oraciones se reduce a la desambiguación de todas las instancias de caracteres de puntuación que pueden delimitar oraciones. El alcance de este problema varía mucho según el idioma, al igual que el número de signos de puntuación diferentes que deben tenerse en cuenta. A veces se usa un espacio en los saltos de frase, o muy a menudo no hay separación entre oraciones.

Incluso los idiomas con sistemas de puntuación relativamente ricos, como el español, presentan problemas. Reconocer límites en un lenguaje escrito de este tipo implica determinar los roles de todos los signos de puntuación, que pueden denotar límites de oraciones: puntos, signos de interrogación, signos de exclamación y, a veces, puntos y comas, dos puntos, guiones y comas. En las colecciones de documentos grandes, cada uno de estos signos de puntuación puede servir para varios propósitos diferentes además de marcar los límites de las oraciones. Un punto, por ejemplo, puede indicar un punto decimal o un marcador de miles, una abreviatura, el final de una oración o incluso una abreviatura al final de una oración. Elipsis (una serie de puntos (...)) puede ocurrir tanto dentro de las oraciones como en los límites de las oraciones. Los signos de exclamación y los signos de interrogación pueden aparecer al final de una oración, pero también entre comillas o entre paréntesis, o incluso, aunque con poca frecuencia, dentro de una palabra, como en la empresa de Internet *Yahoo!*. Sin embargo, las convenciones para el uso de estos dos signos de puntuación también varían según el idioma; en español, ambos pueden reconocerse inequívocamente como delimitadores de oraciones por la presencia de '¡' o '¿' al comienzo de la oración.

El problema del preprocesamiento de texto se pasó por alto o se idealizó en los primeros sistemas de NLP; la tokenización y la segmentación de oraciones fueron frecuentemente descartadas por carecer de interés. Esto fue posible porque la mayoría de los sistemas se diseñaron para procesar textos pequeños, monolingües, que ya se habían seleccionado y preprocesado manualmente. Al procesar textos en un solo idioma con convenciones ortográficas predecibles, fue posible crear y mantener algoritmos construidos a mano para realizar tokenización y segmentación de oraciones. Sin embargo, la reciente explosión en la disponibilidad de grandes corpus no restringidos en muchos idiomas diferentes, y la demanda resultante de herramientas para procesar dichos corpus, ha obligado examinar los numerosos desafíos que plantea el procesamiento de textos sin restricciones. El resultado ha sido un movimiento hacia el desarrollo de algoritmos robustos, que no dependen de la buena formación de los textos que se procesan. Muchas de las técnicas creadas a mano se han reemplazado por enfoques de corpus que se pueden entrenar, que utilizan el aprendizaje automático para mejorar su rendimiento.

Dado que los errores en la etapa de segmentación de texto afectan directamente a todas las etapas de procesamiento posteriores, es esencial comprender y abordar completamente los problemas relacionados con la selección de documentos, la tokenización y la segmentación de oraciones y cómo afectan el procesamiento posterior. Muchos de estos problemas dependen del idioma: la complejidad de la tokenización y la segmentación de oraciones y las decisiones de implementación específicas dependen en gran medida del idioma que se procesa y las características de su sistema de escritura. Para un corpus en un idioma particular, las características del corpus y los requisitos de la aplicación también afectan el diseño y la implementación de algoritmos de tokenización y segmentación de oraciones. En la mayoría de los casos, dado que la segmentación de texto no es el objetivo principal de los sistemas NLP, no se puede considerar simplemente como un paso de "preprocesamiento" independiente, sino que debe estar estrechamente integrado con el diseño y la implementación de todas las demás etapas del sistema.

3.3 Análisis de texto

Como mencionamos anteriormente, el procesamiento del lenguaje abarca distintos análisis de la estructura del texto. Existen tres niveles en los que coinciden los autores -el análisis sintáctico, semántico y pragmático-, pero algunos incluyen también el análisis morfológico.

- *Análisis morfológico*. Consiste en determinar la forma, clase o categoría gramatical de cada palabra de una oración. Se extrae por ejemplo raíces, rasgos flexivos, unidades léxicas compuestas, entre otros.
- *Análisis sintáctico*. Se analiza la estructura sintáctica de la frase mediante una gramática de la lengua en cuestión.
- *Análisis semántico*. Se extrae el significado de la frase, y la resolución de ambigüedades léxicas y estructurales.
- *Análisis pragmático*. Se analiza el texto más allá de los límites de la frase, por ejemplo, para determinar los antecedentes referenciales de los pronombres.

Además se pueden incluir otros niveles de conocimiento como es la información fonológica, referente a la relación de las palabras con el sonido asociado a su pronunciación; el análisis del discurso, que estudia cómo la información precedente puede ser relevante para la comprensión de otra información; y, finalmente, lo que se denomina conocimiento del mundo, referente al conocimiento general que los interlocutores han de tener sobre la estructura del mundo para mantener una conversación.

3.3.1 Análisis morfológico

Las palabras, por supuesto, no son atómicas, y al separarlas, podemos descubrir información que será útil en etapas posteriores del procesamiento. Su función consiste en detectar cómo se relacionan los morfemas, que son las unidades mínimas que forman una palabra. Algunos de los conceptos analizados en esta etapa pueden ser: reconocimiento de prefijos (anti-) o

sufijos (-ísimo), extracción de raíces (bibliotecas, bibliotecario, bibliotec-), rasgos flexivos (-a femenino, -o masculino, -s plural), partes del discurso (*POS - Part of Speech*), entre otros. Este nivel de análisis mantiene una estrecha relación con el léxico.

El léxico brinda la información que se puede extraer de cada palabra, que se utiliza para el procesamiento. Las palabras que forman parte del diccionario están representadas por una entrada léxica; tendrá más de diferentes entradas si la misma tiene más de un significado o diferentes categorías gramaticales. En el léxico se incluye la información morfológica, la categoría gramatical, irregularidades sintácticas y representación del significado.

Normalmente el léxico sólo contiene la raíz de las palabras con formas regulares, quién establece si el género, número o flexión que componen el resto de la palabra son adecuados es el analizador morfológico.

Una palabra se puede pensar de dos maneras, como una cadena en el texto, por ejemplo, el verbo conjugado *entrega*; o como un objeto más abstracto que es el término de entrada para un conjunto de cadenas. Entonces, el verbo *entregar* nombra el conjunto {entrega, entregarás, entregado}. Una tarea básica del análisis léxico es relacionar las variantes morfológicas con su lema que se encuentra en un diccionario con su información semántica y sintáctica. La *lematización* se usa de diferentes maneras dependiendo de la tarea del sistema de procesamiento del lenguaje natural. En la traducción automática, por ejemplo, se puede acceder a la semántica léxica de cadenas de palabras a través del diccionario de lemas. En los modelos de transferencia, se puede utilizar como parte del análisis lingüístico del lenguaje de origen para obtener la representación morfosintáctica de cadenas que pueden ocupar ciertas posiciones en árboles sintácticos, el resultado de análisis sintácticos. Esto requiere que los lemas estén provistos no sólo de información semántica sino también morfosintáctica. Entonces **entrega** se referencia por el artículo ***entregar* + {3er, singular, presente}**.

En la recuperación de información, el análisis sintáctico y la generación sirven para diferentes propósitos. Para la creación automática de una lista de términos clave, tiene sentido colapsar teóricamente variantes morfológicas bajo un lema. Esto se logra con la técnica *stemming*, una operación de preprocesamiento de texto donde se identifican cadenas morfológicamente

complejas, descompuestas de la forma canónica del lema y afijos; y estos últimos se eliminan. El resultado son los textos como objetos de búsqueda que consisten únicamente en *troncos* para que puedan buscarse a través de una lista de lemas. La generación morfológica también desempeña un papel en la recuperación de información; no en la etapa de preprocesamiento, sino como parte de la coincidencia de consultas: dado que un lema tiene semántica invariable, encontrar una ocurrencia de una de sus variantes morfológicas satisface las demandas semánticas de una búsqueda. Además, teniendo en cuenta que la morfología se usa para crear nuevas palabras mediante la derivación, un texto que utiliza una palabra recién acuñada no se perderá si la cadena fuera una de las muchas salidas de una regla morfológica productiva que opera sobre un lema dado. Los diccionarios de ortografía también hacen uso de la generación morfológica por la misma razón, para tener en cuenta las palabras enumeradas y las palabras "potenciales". Otra aplicación del análisis léxico es el preprocesamiento de texto para el análisis sintáctico, donde el análisis de una cadena en categorías y subcategorías morfosintácticas proporciona la cadena con etiquetas *POS* para la entrada de un análisis sintáctico.

3.3.2 Análisis sintáctico

Una presuposición en la mayoría del trabajo en el procesamiento del lenguaje natural es que la unidad básica del análisis del significado es la oración: una oración expresa una proposición, una idea o un pensamiento, y dice algo acerca de un mundo real o imaginario. Las oraciones no son, sin embargo, sólo secuencias lineales de palabras, por lo que es ampliamente reconocido que para llevar a cabo esta tarea se requiere un análisis de cada oración, que determina su estructura de una forma u otra. El análisis sintáctico tiene como función etiquetar cada uno de los componentes sintácticos que aparecen en la oración y analizar cómo las palabras se combinan para formar construcciones gramaticalmente correctas. El producto de este proceso consiste en generar la estructura correspondiente a las categorías sintácticas formadas por cada una de las unidades léxicas que aparecen en la oración,

con el fin de brindar un resultado deseado del análisis gramatical para la interpretación semántica.

Las gramáticas están formadas por un conjunto de reglas, que tiene como función la composición de estructuras. La forma estándar de representar la estructura sintáctica de una oración gramatical es un árbol de análisis sintáctico, que es una representación de todos los pasos en la derivación de la oración desde el nodo raíz. Esto significa que cada nodo interno en el árbol representa una aplicación de una regla de gramática. El árbol de sintaxis de la oración de ejemplo *"el hombre tripula un barco"* se muestra en la Figura 2. Hay que tener en cuenta que el árbol se dibuja boca abajo, con la raíz del árbol en la parte superior y las hojas en la parte inferior.

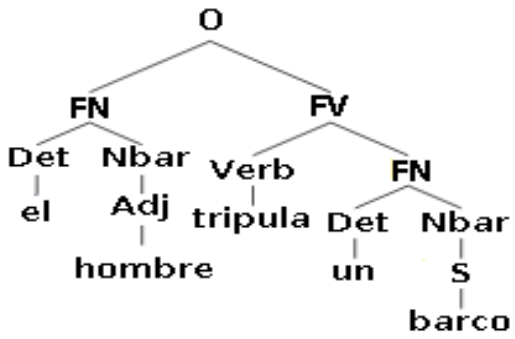


Figura 2. Árbol sintáctico de la frase *"el hombre tripula un barco"*. [26]

Otra representación, que se usa comúnmente en el análisis de texto, es una oración entre corchetes, donde los corchetes tienen la misma etiqueta que los nodos no terminales del árbol de la figura 2:
[O [FN [Det el] [NBar [Adj hombre]]] [FV [Verb tripula] [FN [Det un] [NBar [S barco]]]]], donde O es oración; FN es frase nominal; Det es determinante; NBar es una categoría sintáctica, que al momento de evaluar una declaración se le puede asignar una categoría específica (sustantivo, verbo, adjetivo, preposición, etcétera); Adj es adjetivo; FV es frase verbal; y S es sustantivo.

Si bien la técnica *POS* es utilizada en el análisis morfológico (para identificar los morfemas que constituyen cada palabra), también aplica al sintáctico, debido a que otra de las funciones de esta técnica es la de identificar qué elemento dentro de la oración representa cada palabra (sustantivo, adjetivo, verbo, etcétera).

Un *reconocedor* es un procedimiento que determina si una oración de entrada es o no gramatical de acuerdo con la gramática, incluido el léxico. Un analizador es un algoritmo reconocedor que produce análisis estructurales asociados de acuerdo con la gramática. Un analizador robusto intenta producir resultados útiles, como un análisis parcial, incluso si la entrada no está cubierta por la gramática.

Se puede pensar en una gramática como la inducción de un espacio de búsqueda que consiste en un conjunto de estados que representan etapas de reescrituras sucesivas de reglas gramaticales y un conjunto de transiciones entre estos estados. Al analizar una oración, el analizador reconocedor debe reescribir las reglas de la gramática en alguna secuencia. Una secuencia que conecta el estado *S*, la cadena que consiste en solo la categoría de inicio de la gramática, y un estado que consiste exactamente en la cadena de palabras de entrada, se llama *derivación*. Cada estado en la secuencia se denomina forma sentencial. Si tal secuencia existe, se dice que la oración es gramatical según la gramática.

Los analizadores pueden clasificarse en varias dimensiones de acuerdo con la forma en que llevan a cabo las derivaciones. Una de esas dimensiones se refiere a la invocación de reglas: en una derivación descendente, cada forma sentencial se produce a partir de su predecesor reemplazando un símbolo no terminal A por una cadena de símbolos terminales o no terminales $X_1 \ldots X_d$, donde $A \rightarrow X_1 \ldots X_d$ es una regla gramatical. Por el contrario, en una derivación ascendente, cada forma sentencial se produce reemplazando $X_1 \ldots X_d$ por A dada la misma regla gramatical, aplicando sucesivamente las reglas en la dirección inversa.

Otra dimensión se refiere a la forma en que el analizador trata la ambigüedad, en particular, si el proceso es determinista o no determinista. En el primer caso, solo se puede hacer una única elección irrevocable cuando el analizador se encuentra con una ambigüedad local. Esta elección generalmente se basa en alguna forma de anticipación o preferencia sistemática.

3.3.3 Análisis semántico

Identificar la estructura sintáctica subyacente de una secuencia de palabras es solo un paso para determinar el significado de una oración; se proporciona un objeto estructurado que es más susceptible a una mayor manipulación e interpretación posterior. Son estos pasos subsiguientes los que derivan un significado para la oración en cuestión. El análisis semántico es parte de la fase más compleja del NLP, basándose en el conocimiento acerca de la estructura del texto, se especifica el significado de las palabras, frases y oraciones.

En muchas aplicaciones de NLP los objetivos del análisis apuntan hacia el procesamiento del significado. Desde el punto de vista de una máquina, definir el significado de una oración puede dar lugar a diversas interpretaciones, porque las palabras pueden tener más de un significado, o porque ciertas palabras, como cuantificadores, modales u operadores negativos pueden aplicarse a diferentes tramos de texto, o porque la referencia prevista de los pronombres u otras expresiones de referencia puede no ser clara. A efectos funcionales, para facilitar el procesamiento, la modularidad es una de las propiedades más deseables. Haciendo uso de esta concepción modular es posible distinguir entre significado independiente y significado dependiente del contexto. El primero, tratado por la semántica, hace referencia al significado que las palabras tienen por sí mismas sin considerar el significado adquirido según el uso en una determinada circunstancia. La semántica, por tanto, hace referencia a las condiciones de verdad de la frase, ignorando la influencia del contexto o las intenciones del hablante. Por otra parte, el componente significativo de una frase asociado a las circunstancias en que ésta se da, es estudiado por la pragmática y conocido como significado dependiente del contexto.

En referencia a la estructura semántica que se va a generar, puede ser de interés que exista una simetría respecto a la estructura sintáctica, o por el contrario que no se dé tal correspondencia entre ellas. En el primer caso, a partir del árbol generado por el análisis sintáctico se genera una estructura arbórea con las mismas características, sobre la cual se realizará el análisis semántico. En el segundo caso, en la estructura generada por la sintaxis se

produce un curso de transformaciones sobre las cuales se genera la representación semántica.

3.3.4 Análisis pragmático

Este nivel añade al análisis del significado de la frase información adicional, como por ejemplo, determinar los antecedentes referenciales de los pronombres, en función del contexto donde aparece. Se trata de uno de los niveles de análisis más complejos, cuya finalidad es incorporar al análisis semántico la aportación significativa que pueden hacer los participantes, la evolución del discurso o información presupuesta. Incorpora así mismo información sobre las relaciones que se dan entre los hechos que forman el contexto y entre diferentes entidades.

La semántica se ha formalizado típicamente en lógica de predicados, con la suposición de que una declaración debe interpretarse en términos de su "valor de verdad". Los predicados lógicos son formulaciones precisas de las relaciones entre los objetos en el mundo (así como los estados de sus atributos) y es un enfoque para representar el significado canónico de enunciados declarativos que pueden aparecer en una variedad de formas diferentes. Por lo tanto, con la adición de semántica, un sistema NLP puede resolver ambigüedades que surgen de estructuras de frase incompletas al referirse a un conjunto de predicados que corresponden a hechos conocidos, o mediante el uso de inferencias lógicas para derivar nuevos hechos, con el fin de descubrir la verdad inherente a la expresión de un hablante. Sin embargo, el enfoque de "valor de verdad" es cuestionado por preguntas, solicitudes o cualquier otra declaración que no sea puramente declarativa. Se requiere un análisis más detallado, por ejemplo, para descubrir la verdad que se afirma en la expresión: “*Disculpe, por favor*”. Pero ya sea que se vea en términos de verdad o colecciones de asociaciones, cualquier interpretación del significado *puro* de un enunciado dado está contaminada por el contexto más amplio de cualquier interacción. La semántica se funde con la pragmática, con el reconocimiento de que las personas que se involucran en una conversación se

manipulan entre sí de forma activa, califican según lo que dicen y dependen en gran medida de la historia de la interacción.

La pragmática es, en última instancia, la base a partir de la cual un sistema NLP puede decirnos por qué se emite un enunciado dado.

La pragmática surge con las ambigüedades en los niveles sintáctico o semántico, y entra en juego el contexto y el propósito del enunciado considerado para el análisis. Uno de los problemas más comunes a la hora de interpretar las entradas de lenguaje natural en un sistema informático implica la resolución de referencias ambiguas en frases nominales. Por ejemplo, si un sistema NLP encuentra la referencia definitiva "*la llave*", puede que no haya ninguna indicación en la oración actual sobre qué *"llave"* se está mencionando. Puede ser una referencia a algo que se introdujo en el discurso anteriormente, o esta oración puede contener la primera mención de *"llave"*, siendo un objeto en particular. Sin esta información, el sistema puede asociar atributos improbables o incluso muy incompatibles con tal objeto. Un problema similar puede surgir con frases nominales indefinidas. Una referencia indefinida específica, como "*una llave*" no identifica ninguna *llave* en particular y puede no haber sido referenciada anteriormente en el diálogo.

Quizás aún más difícil de interpretar es una referencia a un sustantivo indefinido no específico ya que en *"una rosa es una rosa"* no expresa que ninguna rosa real sea nombrada; en cambio, es cualquier instancia dada de la clase genérica *"rosa"* sobre la cual se está predicando algo. Finalmente, y tal vez el problema más grave para la desambiguación de un frase nominal, es el caso de pronombres como *"eso"*. Este tipo de referencia es la clásica que necesita apoyo contextual para interpretar correctamente.

Un enfoque para el problema de la frase nominal es usar reglas que busquen incoherencias o contradicciones específicas en un contexto. Cada vez que se encuentra una nueva declaración, las reglas se verifican para estas propiedades. Por ejemplo, suponga que las siguientes declaraciones se hacen a un camarero que toma pedidos: *"El agua está bien. Me gustaría con hielo"*. Dentro de un sistema NLP, esta situación podría ser parecida a la siguiente regla: Si *X* está *"con" Y*, entonces *X* no es *Y*. Dada esta regla, el sistema no se equivocaría al concluir que, en la segunda oración, se refiere al hielo (por ejemplo, "*me gustaría el hielo con hielo*").

Otra técnica es el uso de *scripts* -representaciones de secuencias prototípicas de eventos que restringen los posibles roles desempeñados por acciones que ocurren en un contexto dado-. A medida que se ejecutan las acciones, se asignan a los roles definidos en el script; se dice que completan o coinciden los *slots* que representan los eventos que se espera que se ejecuten en un orden determinado. Por ejemplo, se considera un fragmento de una secuencia de comandos que tiene que ver con la solicitud de bebidas. Supongamos que después de que ya hayan ocurrido cinco acciones, la sexta es la de ordenar una bebida, representada por la ranura nº 6 a continuación:

Slot nº 6: (Pide una bebida): <x>

Slot nº 7: (<x> con / sin hielo): [Verdadero/Falso]

La desambiguación del referente en "*Me gustaría con hielo*" probablemente se manejaría por el hecho de que el slot nº 7 *espera* una referencia sobre si se quería o no hielo con la bebida ordenada en el slot nº6.

Una herramienta general, que podría ser utilizada por diferentes técnicas, es la de *listas históricas*. Esto implica mantener una lista detallada de enunciados previos, de modo que cuando surge una referencia ambigua (como en el caso del pronombre), la lista se escanea en busca de pistas contextuales relevantes sobre su posible referente. Una heurística útil en este sentido es calcular la distancia (en palabras) desde una referencia actual a algún referente previo, inequívoco. En el ejemplo de ordenar agua con hielo, el sustantivo *"agua"* se encuentra a una distancia bastante corta del pronombre *"eso"* y, por lo tanto, podría determinarse como su referente.

Las listas históricas también se pueden usar para reconocer el rol que una referencia ambigua podría tener en una tarea estructurada jerárquicamente. Una tarea que está concebida jerárquicamente puede tener secuencias opcionales de elementos particulares; siempre que se hayan cumplido todas los subobjetivos de la tarea; puede no importar en qué orden se produjeron. Un cálculo simple de la distancia de un pronombre ambiguo puede no encontrar el referente apropiado, pero un enfoque de tarea jerárquica podría.

3.4 Otras técnicas

***Presencia y frecuencia de términos (Terms presence and frequency)*:** Estas características son palabras individuales y n-gramas y su correspondiente conteo de frecuencia. Ésta nos da, o bien una representación binaria (0 si no aparece o 1 si lo hace), o usa un valor que representa la frecuencia de aparición de dicho término, para indicar su importancia relativa [19].

***Palabras y frases de opinión (Opinion words and phrases)*:** éstas son palabras comúnmente utilizadas para expresar opiniones incluyendo *bueno* o *malo*, *querer* u *odiar*. Por otro lado, algunas frases expresan opiniones sin usar palabras de opinión. Por ejemplo: *me costó un ojo de la cara*.

***Negaciones (Negations)*:** La aparición de palabras negativas puede cambiar la orientación de la opinión, tal como *no bueno* es equivalente a *malo*.

***Bolsa de palabras (BoW - Bag of Words)*:** es una técnica que consiste en recibir un texto y devolver un valor numérico correspondiente a cada palabra, reflejando la cantidad de ocurrencias que tiene la misma en dicho texto. Comparando BoW's de distintos textos, podemos saber fácilmente cuales son las palabras más comunes entre ellos y cuales las más específicas.

Aplicamos este concepto, tomando como ejemplo los siguientes textos:

1. *Me gusta comer.*
2. *Me agrada comer y viajar.*
3. *Viajar es mi pasión y mi hobby.*

	me	gusta	comer	agrada	y	viajar	es	mi	pasión	hobby
Texto 1	1	1	1							
Texto 2	1		1	1	1	1				
Texto 3					1	1	1	2	1	1

Tabla 2. Tabla ejemplo BoW.

Podemos observar que el término **gusta** aparece únicamente en el primer texto, **agrada** solo en el segundo, y **es mi pasión hobby** solo en el tercero.

***Frecuencia de término - Frecuencia de documento inversa (TF-IDF - Term Frequency - Inverse Document Frequency)*:** La frecuencia de término por sí sola, es básicamente la salida del BoW. TF mide la importancia local del término, es decir, que tan importante es cada palabra dentro de cada texto. A mayor cantidad de ocurrencias, se supone que tiene más relevancia.

Ahora bien, la segunda parte de éste concepto (IDF) analiza qué tan específica es la palabra, tomando en cuenta el total de los textos. Para que un término se considere específico de un texto, no debería aparecer seguido en el resto, por lo que su frecuencia debería ser baja, lo que deriva en que su frecuencia de documento inversa sería alta. Para calcular el valor de la IDF se sigue la siguiente fórmula:

$$\text{idf(T)} = \log(\ \#(\text{textos}) \ / \ \#(\text{textos conteniendo el término T})\ )$$

donde #(X) significa "Cantidad de X"

Finalmente, TF-IDF es el producto de estas dos frecuencias, es decir, el valor obtenido de IDF para cada término, multiplicado por su frecuencia TF. Los valores aplicados a los textos utilizados como ejemplo serían los siguientes:

	me	gusta	comer	agrada	y	viajar	es	mi	pasión	hobby
Texto 1	0.18	**0.48**	0.18							
Texto 2	0.18		0.18	**0.48**	0.18	0.18				
Texto 3					0.18	0.18	**0.48**	**0.95**	**0.48**	**0.48**

Tabla 3. Tabla ejemplo TF-IDF.

Reconocimiento de entidad con nombre (NER - Named Entity Recognition): NER etiqueta los elementos atómicos en la oración en categorías como "persona" o "ubicación"; a cada palabra se le asigna una etiqueta con el prefijo del comienzo o el interior de una entidad.

3.5 Aplicaciones y ejemplos

En los siguientes casos de alto nivel, donde a menudo se utiliza NLP, el objetivo primordial es tomar la entrada del lenguaje en bruto y usar la lingüística y los algoritmos para transformar o enriquecer el texto de tal manera que ofrezca un mayor valor.

- Descubrimiento investigativo. Se identifica patrones y pistas en correos electrónicos o informes escritos para ayudar a detectar y resolver crímenes.
- Experiencia en el tema. Se clasifica el contenido en temas significativos para que pueda tomar medidas y descubrir tendencias.
- Análisis de redes sociales. Se hace un seguimiento de la conciencia y el sentimiento sobre temas específicos y se identifica personas influyentes.
- Categorización del contenido. Se obtiene un resumen de documentos basado en el lenguaje, que incluye búsqueda e indexación, alertas de contenido y detección de duplicaciones.
- Descubrimiento de tópicos y modelado. Se captura con precisión el significado y los temas en las colecciones de texto, y se aplica análisis avanzados al texto, como la optimización y la previsión.
- Extracción contextual. Se extrae automáticamente información estructurada de fuentes basadas en texto.
- Análisis de los sentimientos. Se identifica el estado de ánimo o las opiniones subjetivas dentro de grandes cantidades de texto, incluido el sentimiento promedio y la minería de opinión.
- Conversión de voz a texto y de conversión de texto a voz. Se transforma los comandos de voz en texto escrito, y viceversa.
- Resumen del documento. Se realiza una generación automática de sinopsis de grandes cuerpos de texto.
- Máquina traductora. Se traduce automáticamente texto o voz de un idioma a otro.

También existen muchas aplicaciones comunes y prácticas de NLP en nuestra vida cotidiana. Más allá de conversar con asistentes virtuales como *Alexa* o *Siri*, hay algunos ejemplos más:

- En los correos electrónicos en la carpeta de spam hay similitudes en las líneas de asunto, se utiliza el filtrado de spam bayesiano, una técnica estadística de NLP que compara las palabras en el correo no deseado con las ya seleccionadas para identificarlo.
- Cuando se navega por un sitio web y se utiliza la barra de búsqueda integrada, o seleccionando etiquetas de tema, entidad o categoría sugeridas, se hace uso de métodos NLP para búsqueda, modelado de temas, extracción de entidades y categorización de contenido.

Capítulo 4

Minería de Opinión

4.1 Introducción

Una **opinión**, según la definición de la RAE, es un juicio o valoración que se forma una persona respecto de algo o de alguien. Se identifican por ser un concepto personal subjetivo en relación con un tema determinado, permitiendo un intercambio de las mismas frente a otros individuos. Pueden ser regulares o comparativas. Las primeras, refieren una valoración a una sola entidad en particular o un aspecto de la misma, por ejemplo, *"las hamburguesas de Mostaza son muy ricas"*, lo que expresa un sentimiento positivo sobre el aspecto del sabor de las hamburguesas de Mostaza. Mientras que las comparativas son aquellas que evidencian una comparación de dos o más entidades en algunos de sus aspectos compartidos, señalando las similitudes y/o diferencias de las mismas; por ejemplo, *"las hamburguesas de Mostaza saben mejor que las de McDonald's"*, que compara Mostaza y McDonald's en función de sus gustos y expresa una preferencia por Mostaza.

Las opiniones son importantes en la mayoría de las actividades que realizan las personas y son influyentes en sus comportamientos. Las creencias y percepciones de la realidad que cada individuo pueda tener, y las elecciones que hagan, están, en gran medida, condicionadas a cómo otros ven y evalúan el mundo. Por esta razón, cuando se necesita tomar una decisión, a menudo se busca las opiniones de los demás. En el mundo real, las empresas y las organizaciones siempre quieren encontrar opiniones del consumidor o del público sobre sus productos y servicios. Los consumidores también desean conocer las opiniones de otros usuarios de un producto antes de comprarlo, y las opiniones de los demás sobre los candidatos políticos antes de tomar una decisión de voto en una elección política. En el pasado, cuando un individuo necesitaba opiniones, le preguntaba a sus amigos y familiares. Cuando una

organización o empresa necesitaba opiniones públicas o de los consumidores, realizaba cuestionarios, encuestas de opinión y grupos focales. La adquisición de opiniones públicas y de los consumidores ha sido durante mucho tiempo un gran negocio para las empresas de marketing, relaciones públicas y campañas políticas.

Con el crecimiento explosivo de las redes sociales, por ejemplo, reseñas, debates en foros, blogs, microblogs, comentarios y publicaciones en sitios de redes sociales en la Web, individuos y organizaciones utilizan cada vez más el contenido en estos medios para la toma de decisiones. Hoy en día, si uno quiere comprar un producto, ya no se limita a pedirle opiniones a su entorno porque hay muchas reseñas de usuarios y discusiones en foros públicos en la Web sobre el producto. Para una organización, puede que ya no sea necesario realizar cuestionarios, encuestas de opinión y grupos focales para recopilar opiniones públicas porque existe una gran cantidad de dicha información disponible públicamente. Sin embargo, encontrar y monitorear sitios de opinión en la Web y destilar la información contenida en ellos rehace una tarea formidable debido al crecimiento de diversos sitios. Cada sitio normalmente contiene un gran volumen de texto de opinión que no siempre se descifra fácilmente en blogs extensos y publicaciones en foros. El lector promedio tendrá dificultades para identificar sitios relevantes y extraer y resumir las opiniones en ellos. Por supuesto, los documentos obsoletos no solo existen en la web, llamados datos externos; muchas organizaciones también tienen sus datos internos, por ejemplo, comentarios de los clientes recopilados de correos electrónicos y centros de llamadas o resultados de encuestas realizadas por las organizaciones. En resumen, la abundancia de datos, junto con la necesidad de poderosas herramientas de análisis de datos, se ha descrito como una situación rica en datos pero pobre en información. La gran cantidad de datos, de rápido crecimiento, recopilada y almacenada en grandes y numerosos repositorios de datos, ha superado ampliamente la capacidad *humana* para la comprensión sin poderosas herramientas. Desafortunadamente, el procedimiento manual de entrada de conocimiento es propenso a sesgos y errores, y es extremadamente costoso y lento. La creciente brecha entre los datos y la información exige el desarrollo sistemático de herramientas de

minería de datos que pueden convertir las *tumbas* de datos en conocimiento de gran valor.

Es por ello que la Minería de Datos genera gran interés en la actualidad, por la necesidad de obtener información útil y conocimiento que pueda ser utilizado para la creación de diversas aplicaciones a partir del crecimiento exponencial que están sufriendo los datos. Esta tarea se lleva a cabo mediante herramientas adicionales de análisis, aparte de las que proveen las bases de datos -como consultas y transacciones-, para un análisis más profundo sobre los mismos, permitiendo obtener patrones importantes. La minería de datos se puede ver como resultado de la evolución natural de la tecnología de la información. El desarrollo temprano de los mecanismos de recopilación de datos y creación de bases de datos fue condicionante para el posterior desarrollo de mecanismos efectivos para el almacenamiento y la recuperación de datos.

La Minería de Datos es una fase del proceso de Extracción del Conocimiento en Bases de Datos (KDD - *Knowledge Discovery from Databases*). Éste último consiste en una secuencia iterativa de los siguientes pasos [14]:

1. Limpieza de datos, para eliminar ruido y datos inconsistentes.

2. Integración de datos, donde se pueden combinar múltiples fuentes de datos. Una tendencia popular en la industria de la información es realizar la limpieza de datos y la integración de datos como un paso de preproceso, donde los datos resultantes se almacenan en un almacén de datos.

3. Selección de datos, donde los datos relevantes para la tarea de análisis se recuperan de la base de datos.

4. Transformación de datos, donde los datos se transforman y consolidan en formas apropiadas para la minería mediante la realización de operaciones de resumen o agregación. En ocasiones, la transformación y consolidación de datos se realiza antes del proceso de selección de datos, particularmente en el caso del almacenamiento de datos. La reducción de datos también se puede realizar para obtener una representación más pequeña de los datos originales sin sacrificar su integridad.

5. Minería de datos, un proceso esencial donde se aplican métodos inteligentes para extraer patrones de datos.

6. Evaluación de patrones, para identificar los patrones verdaderamente interesantes que representan el conocimiento basados en medidas de interés.

7. Presentación del conocimiento, donde las técnicas de visualización y representación del conocimiento se utilizan para presentar el conocimiento minero a los usuarios.

Los pasos 1 a 4 son diferentes formas de preprocesamiento de datos, donde los datos se preparan para la minería. El paso de minería de datos puede interactuar con el usuario o una base de conocimiento. Los patrones interesantes se presentan al usuario y pueden almacenarse como nuevos conocimientos en la base de conocimiento.

El **Análisis de Sentimientos** (SA), también conocido como **Minería de Opinión** (OM), se define como el estudio computacional de opiniones, sentimientos, emociones y subjetividad expresadas en textos hacia una entidad, ya sea a una persona, tema, producto, empresas, eventos [12] [32]. Como se menciona en [19], algunos autores sostienen que tienen nociones levemente diferentes. La Minería de Opinión extrae y analiza la opinión de las personas sobre una entidad, mientras que el Análisis de Sentimientos identifica el sentimiento expresado en un texto y luego lo analiza. Por lo tanto, el objetivo de Análisis de Sentimientos es determinar opiniones, identificar los sentimientos que expresa un escritor mediante su texto, y luego clasificar su polaridad como se muestra en la Figura 3. En este ensayo, utilizamos los términos Análisis de Sentimientos y Minería de Opinión indistintamente.

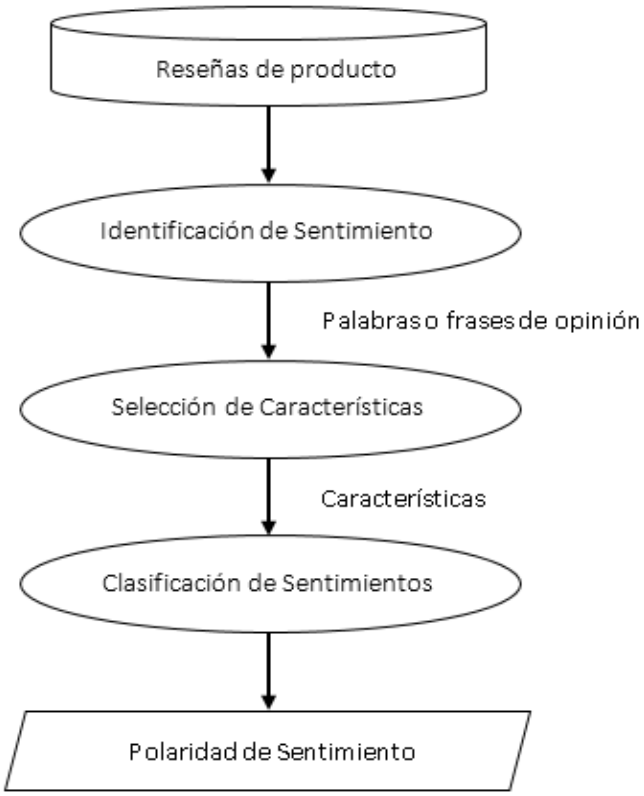


Figura 3. Proceso de SA en reseñas de producto. [19]

El inicio y el gran crecimiento de la Minería de Opinión es igual al de las redes sociales en la Web, mencionadas anteriormente, porque se tiene la posibilidad de obtener grandes volúmenes de información sobre lo que las personas piensan en ese momento sobre una temática en particular, además del bajo costo que conlleva.

Así mismo cabe destacar la rapidez y espontaneidad en que se obtiene la información ya que se realiza en el mismo momento en que se minan las opiniones. En comparación con los métodos tradicionales para obtener información sobre la opinión pública de un tópico, además de la lentitud en la recopilación y análisis posterior de la misma, la Minería de Opinión nos permite realizar un seguimiento del impacto, comparando las métricas de antes y después del acontecimiento de un evento o noticia, ya que una persona puede cambiar su forma de pensar o ver las cosas.

En [32] se señala que desde principios de 2000, el Análisis del Sentimiento se ha convertido en una de las áreas de investigación más activas en el procesamiento del lenguaje natural. De hecho, se ha extendido desde la informática hasta las ciencias de la gestión y las ciencias sociales debido a su

importancia para las empresas y la sociedad en general. En los últimos años, las actividades industriales que rodean el Análisis del Sentimiento también han prosperado. Muchas grandes corporaciones han construido sus propias capacidades internas. Los sistemas de análisis de sentimiento han encontrado sus aplicaciones en casi todos los negocios y el dominio social.

Las aplicaciones de análisis de sentimiento se han extendido a casi todos los dominios posibles, desde productos, diversos servicios, atención médica, artículos de noticias hasta eventos sociales y elecciones políticas. En los debates políticos, por ejemplo, podríamos descifrar las opiniones de las personas sobre ciertos candidatos a elecciones o partidos políticos. Los resultados de las elecciones también pueden predecirse a partir de publicaciones políticas. Los sitios de redes sociales y los sitios de microblogs se consideran una muy buena fuente de información porque las personas comparten y discuten sus opiniones sobre distintos tópicos libremente. Como se indica en [32], muchas grandes corporaciones han desarrollado sus propias aplicaciones para el análisis de sentimientos, por ejemplo, Microsoft, Google, Hewlett-Packard, entre otras. Estas utilidades prácticas e intereses industriales han proporcionado fuertes motivaciones para la investigación en análisis de sentimiento.

4.1.1 Diferentes niveles de análisis

Como hemos visto, existen diferentes niveles de granularidad del Procesamiento de Lenguaje Natural. En función de estos, se describen tres niveles de clasificación principales: nivel de documento, nivel de oración y nivel de entidad y aspecto.

4.1.1.1 Nivel de documento

Este nivel de clasificación tiene como objetivo clasificar un documento de opinión completo como una valoración o sentimiento positivo o negativo. Considera el documento completo como una unidad de información básica, que habla de una entidad, tema o evento. Por ejemplo, dada una revisión del

producto, se determina si la revisión expresa una opinión general positiva o negativa sobre el producto.

Este nivel de clasificación tiene algunas deficiencias a la hora de aplicarlo:

- En muchas aplicaciones, el usuario necesita saber detalles adicionales, por ejemplo, qué aspectos de las entidades son del agrado y desagrado de los consumidores. En los documentos de opinión, se proporcionan tales detalles, pero la clasificación del sentimiento del documento no los extrae.
- Supone que cada documento expresa opiniones sobre una sola entidad, por ejemplo, un único producto. Por lo tanto, no es aplicable a los documentos que evalúan o comparan varias entidades, como las discusiones en foros, blogs y los artículos de noticias, ya que muchas publicaciones de este tipo evalúan múltiples entidades y las comparan. La clasificación de sentimiento a nivel de documento no realiza tareas tan detalladas, que requieren un procesamiento en profundidad del lenguaje natural.

4.1.1.2 Nivel de oración

Este nivel tiene como objetivo clasificar el sentimiento expresado en cada oración. El primer paso es identificar si la oración es subjetiva u objetiva. Si la oración es subjetiva, el nivel de oración determinará si la misma expresa opiniones positivas, negativas o neutrales. Esta última clase, generalmente significa que no hay opinión. No existe una diferencia fundamental entre las clasificaciones de nivel de documento y oración porque las oraciones son solo documentos cortos o en contraposición, un documento contiene múltiples oraciones, por ende, generalmente múltiples opiniones.

La clasificación del texto en el nivel del documento o en el nivel de la oración no proporciona los detalles necesarios sobre todos los aspectos de la entidad que se necesitan en muchas aplicaciones; para obtener estos detalles, tenemos que ir al nivel de aspecto.

4.1.1.3 Nivel de entidad y aspecto

Este nivel tiene como objetivo clasificar el sentimiento con respecto a los aspectos específicos de las entidades. Tanto el nivel de documento como el

análisis de nivel de oración no descubren qué es exactamente lo que le gusta y lo que no le gusta a la gente. El nivel de aspecto realiza un análisis más detallado. En lugar de analizar las presentaciones del lenguaje (documentos, párrafos, oraciones, cláusulas o frases), el nivel de aspecto analiza directamente la opinión en sí misma.

Se basa en la idea de que una opinión consiste en un sentimiento, positivo o negativo, y un objetivo de opinión. Al tener en cuenta la importancia de los objetivos de opinión, también nos ayuda a comprender mejor el problema del Análisis de Sentimientos. Por ejemplo, por más que la frase *"aunque el servicio no es tan bueno, todavía me encanta este restaurante"* claramente tiene un tono positivo, no podemos decir que esta frase sea completamente positiva. De hecho, la oración es positiva sobre el *restaurante*, siendo este objetivo enfatizado, pero negativa sobre su *servicio* que no se enfatiza. En muchas aplicaciones, los objetivos de opinión son descritos por las entidades y/o sus diferentes aspectos. Por ejemplo, la frase *"la calidad de la llamada del iPhone es buena, pero su duración de la batería es corta"* evalúa dos aspectos, la calidad de la llamada y la duración de la batería, del iPhone, que se define como entidad. El sentimiento sobre la calidad de las llamadas de iPhone es positivo, pero el sentimiento sobre la duración de la batería es negativo. Estos dos aspectos mencionados son los objetivos de opinión.

Como comentamos en las dos secciones anteriores, clasificar los textos de opinión a nivel de documento o de oración, a menudo es insuficiente para las aplicaciones porque no identifican objetivos de opinión ni asignan sentimientos a dichos objetivos. Incluso si suponemos que cada documento evalúa una sola entidad, un documento de opinión positivo sobre la misma no significa que el autor tenga opiniones positivas sobre todos los aspectos de la entidad. Del mismo modo, un documento de opinión negativo no significa que el autor sea negativo sobre todas las características de la entidad. Para un análisis más completo, debemos descubrir los aspectos y determinar si el sentimiento es positivo o negativo en cada aspecto.

4.1.2 Campos relacionados con el análisis de sentimiento

Existen nuevos campos relacionadas con SA. Estos campos incluyen detección de emociones (*Emotion Detection - ED*), recursos de construcción (*Building Resources - BR*) y aprendizaje de transferencia (*Transfer Learning - TL*).

4.1.2.1 Detección de emociones

Este campo tiene como objetivo extraer y analizar las emociones, mientras que las mismas sean explícitas o implícitas en las oraciones. El análisis del sentimiento a veces se considera una tarea de NLP para descubrir opiniones sobre una entidad; y debido a que existe cierta ambigüedad acerca de la diferencia entre opinión, sentimiento y emoción, se define la opinión como un concepto transicional que refleja la actitud hacia una entidad. El sentimiento refleja emociones mientras que la emoción refleja actitudes. En [19] se menciona que existen ocho emociones básicas y prototípicas que son alegría, tristeza, ira, miedo, confianza, disgusto, sorpresa y anticipación. La detección de emociones puede considerarse una tarea específica de SA. La ED se preocupa por detectar varias emociones del texto, mientras que SA se preocupa principalmente por especificar opiniones positivas o negativas. Como una tarea de Análisis de Sentimientos, ED se puede implementar utilizando el enfoque de aprendizaje automático (*Machine Learning - ML*) o el enfoque basado en Lexicon, siendo este último el enfoque utilizado con mayor frecuencia.

4.1.2.2 Recursos de construcción

Este campo tiene como objetivo crear léxica, diccionarios y corpus en los que las expresiones de opinión se anoten según su polaridad. La construcción de recursos no es una tarea de SA, pero podría ayudar a mejorarla y también la ED. Los principales desafíos que enfrenta el trabajo en esta categoría son la ambigüedad de las palabras, la plurilingüística, la granularidad y las diferencias en la expresión de opinión entre los géneros textuales.

4.1.2.3 Transferencia de aprendizaje

La transferencia de aprendizaje extrae conocimiento del dominio auxiliar para mejorar el proceso de aprendizaje en un dominio objetivo. Por ejemplo, transfiere conocimientos de documentos de Wikipedia a tweets. La transferencia de aprendizaje se considera una nueva técnica de aprendizaje entre dominios ya que aborda los diversos aspectos de las diferencias de dominio. Se utiliza para mejorar muchas tareas de minería de textos, como la Clasificación de Texto, Análisis de Sentimientos, etcétera. En Análisis de Sentimientos, la transferencia de aprendizaje se puede aplicar para transferir la clasificación de sentimiento de un dominio a otro o construir un puente entre dos dominios.

La diversidad entre varias fuentes de datos es un problema para la modelización conjunta de múltiples fuentes de datos. Esta modelización es importante para transferir el aprendizaje.

4.2 Técnicas utilizadas para el Análisis de Sentimientos

Las técnicas utilizadas para el análisis de sentimiento se pueden dividir en el enfoque de aprendizaje automático, el enfoque basado en léxico y el enfoque híbrido [3]. En el enfoque de aprendizaje automático (*Machine Learning*) se aplican algoritmos de clasificación (como por ejemplo árboles de decisión, modelos probabilísticos, entre otros) y utiliza características lingüísticas. El enfoque basado en léxico se basa en un léxico de sentimiento, una colección de términos de sentimiento conocidos y precompilados. Se divide en un enfoque basado en diccionario y un enfoque basado en corpus que utiliza métodos estadísticos o semánticos para encontrar la polaridad sensorial. El enfoque híbrido combina ambos enfoques y es muy común que los léxicos de sentimiento jueguen un papel clave en la mayoría de los métodos.

Los métodos de clasificación de texto que utilizan el enfoque ML se pueden dividir en métodos de aprendizaje supervisados y no supervisados. Los métodos supervisados hacen uso de una gran cantidad de documentos de

entrenamiento etiquetados. Los métodos no supervisados se usan cuando es difícil encontrar dichos documentos.

El enfoque basado en léxico depende de encontrar el léxico de opinión que se usa para analizar el texto. Hay dos métodos en este enfoque: el enfoque basado en diccionario, que consiste de buscar palabras clave de opinión, y luego busca en el diccionario sus sinónimos y antónimos. Y el enfoque basado en corpus, el cual comienza con una lista inicial de palabras de opinión, y luego encuentra otras en un gran corpus para ayudar a encontrar más palabras de opinión con orientaciones específicas del contexto. Esto podría hacerse mediante el uso de métodos estadísticos o semánticos.

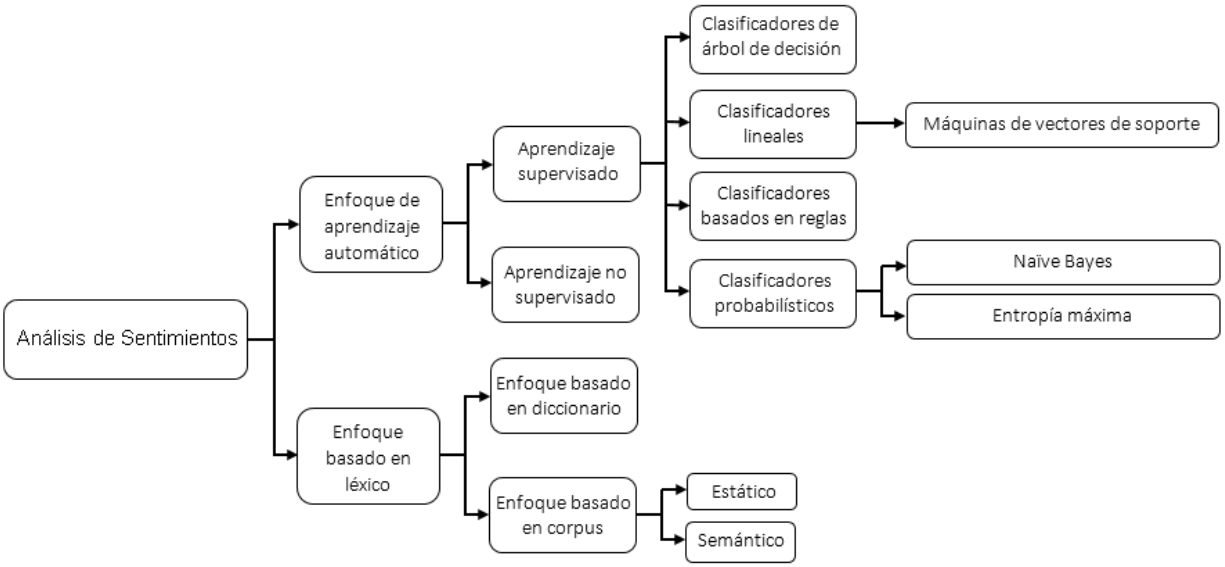


Figura 4. Técnicas de clasificación de sentimientos. [19]

4.2.1 Enfoque de aprendizaje automático

El enfoque de aprendizaje automático utiliza algoritmos de ML para resolver el Análisis de Sentimientos como un problema de clasificación de texto regular que hace uso de características sintácticas y/o lingüísticas.

Clasificación del texto, definición del problema: Sea *D* un conjunto de registros de entrenamiento $D = \{X_1, X_2, ..., X_n\}$ donde cada registro está etiquetado con una clase. El modelo de clasificación está relacionado con las características en el registro subyacente a una de las etiquetas de clase.

Luego, para una instancia dada con clase desconocida, el modelo se utiliza para predecir la etiqueta de dicho registro. El problema de clasificación *difícil (hard)* es cuando solo se asigna una etiqueta a una instancia. El problema de clasificación *suave (soft)* es cuando un valor probabilístico de etiquetas se asigna a un registro.

4.2.1.1 Aprendizaje supervisado

Los métodos de aprendizaje supervisado dependen de la existencia de documentos de entrenamiento ya etiquetados. Existen varios tipos de clasificadores de aprendizaje supervisado. A continuación, describimos brevemente algunos de los estos clasificadores, haciendo hincapié en los algoritmos que utilizamos en el desarrollo de nuestra investigación.

4.2.1.1.1 Clasificadores probabilísticos

Los clasificadores probabilísticos usan modelos denominados *de mezcla* para la clasificación. Éstos son modelos probabilísticos utilizados para representar la presencia de subpoblaciones dentro de una población general, sin necesidad de que los datos posean un identificador para indicar la pertenencia a cada subgrupo. Este tipo de clasificadores también se denominan clasificadores generativos, que pretenden dar cuenta de la variabilidad en el conjunto de observación y permiten su descripción a través de funciones de probabilidad. Describen cómo se generan los datos, en términos de un modelo probabilístico. Una de las cosas más importantes que hay que entender sobre estos modelos es que son capaces de algo más que predecir o clasificar; es decir, maximizar $P(Y|X = x)$. Al estimar $P(Y, X)$ y poder muestrear pares X, Y, se puede usar un modelo generativo para imputar datos faltantes, comprimir el conjunto de datos o generar datos no vistos.

Elegimos estudiar dos de los clasificadores probabilísticos más conocidos y que presentamos en las siguientes secciones.

4.2.1.1.1.1 Clasificador NaÏve Bayes (NB)

El clasificador Naïve Bayes [3] [8] [11] [19] [35] [36] [37] [38] [50] es el clasificador más simple y más utilizado, asume que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de

cualquier otra característica, dada una clase variable. Por ejemplo, una fruta puede ser considerada como una naranja si es de color naranja, redonda y de alrededor de 7 cm de diámetro. Un clasificador de Bayes considera que cada una de estas características contribuye de manera independiente a la probabilidad de que esta fruta sea una naranja, independientemente de la presencia o ausencia de las otras características. Los supuestos de independencia a menudo no tienen un impacto en la realidad, es casi imposible que se obtenga un conjunto de predictores que sean completamente independientes. Por lo tanto, se los considera ingenuos.

El modelo Naïve Bayes es fácil de construir y particularmente útil para conjuntos de datos muy grandes. También funciona bien en la predicción de múltiples clases. Funciona bien en caso de variables de entrada categóricas en comparación con variables numéricas. Para la variable numérica, se supone una distribución normal, pero si la variable categórica tiene una categoría (en el conjunto de datos de prueba), que no se observó en el conjunto de datos de entrenamiento, entonces el modelo asignará una probabilidad de 0 (cero) y no podrá hacer una predicción. Esto a menudo se conoce como “frecuencia cero”.

El modelo funciona con la extracción de características de BoWs, que ignora la posición de la palabra en el documento. Es una lista de probabilidades que incluye: las *probabilidades de cada clase* en el conjunto de datos de entrenamiento y las *probabilidades condicionales* de cada valor de entrada dado cada valor de clase. Utiliza el Teorema de Bayes, también conocido como teorema de la probabilidad condicionada, para predecir la probabilidad que un conjunto de características determinado pertenezca a una etiqueta en particular.

$$\mathrm{P}(\mathrm{label} \mid \mathrm{features}) = (\mathrm{P}(\mathrm{label}) * \mathrm{P}(\mathrm{features} \mid \mathrm{label})) / \mathrm{P}(\mathrm{features})$$

P(label) es la probabilidad previa de una etiqueta o la probabilidad de que una característica aleatoria establezca la etiqueta. *P(features|label)* es la probabilidad previa de que un conjunto de características determinado se clasifique con una etiqueta. *P(features)* es la probabilidad previa de que se produzca un conjunto de características determinado. Dada la suposición

Naïve, que establece que todas las características son independientes, la ecuación se podría reescribir de la siguiente manera:

$$P(\text{label} \mid \text{features}) = (P(\text{label}) * P(f_1 \mid \text{label}) * \ldots * P(f_n \mid \text{label})) / P(\text{features})$$

A modo de ejemplo, planteamos el siguiente problema, observado en [37]:

"Supongamos que un ingeniero está buscando agua en un terreno. A priori, se sabe que la probabilidad de que haya agua en dicha finca es del 60%. No obstante, el ingeniero quiere asegurarse mejor y decide realizar una prueba que permite detectar la presencia o no de agua. Dicha prueba tiene una fiabilidad del 90%, es decir, habiendo agua, la detecta en el 90% de los casos. También, cuando realmente no hay agua, la prueba predice que no hay agua en el 90% de los casos.

Por tanto, pudiendo hacer uso de dicha prueba ¿qué es más probable, que haya agua o que no?"

En el enunciado se plantea que *a priori* tenemos una probabilidad (60% de que haya agua) que se ve afectada por otra probabilidad (el 90% de acierto de la prueba). Según el Teorema de Bayes, la probabilidad que ocurra un hecho en particular, en este caso que haya agua, habiendo sucedido otro que influye en el anterior, que la prueba diga que sí hay agua, se define con la siguiente fórmula:

$$P(A \mid B) = P(B \mid A) * P(A) / P(B)$$

Por tanto, para resolver el problema, por un lado, hay que calcular la probabilidad de que hubiese agua sabiendo que la prueba ha detectado agua (*P(Agua|Prueba+)*); y por otro lado, se debe calcular la probabilidad de que no hubiese agua sabiendo que la prueba no ha detectado agua (*P(NoAgua|Prueba-)*). También se debe tener en cuenta que la probabilidad de que la prueba salga positiva, es la suma de las probabilidades de todos los casos posibles donde pueda salir la prueba positiva. Se debe hacer igual con *P(Prueba-)*:

- $P(\text{Prueba+}) = P(\text{Prueba+}|\text{Agua}) * P(\text{Agua}) + P(\text{Prueba+}|\text{NoAgua}) * P(\text{NoAgua}) = 0{,}9 * 0{,}6 + 0{,}1 * 0{,}4 = 0{,}58$
- $P(\text{Prueba-}) = P(\text{Prueba-}|\text{Agua}) * P(\text{Agua}) + P(\text{Prueba-}|\text{NoAgua}) * P(\text{NoAgua}) = 0{,}1 * 0{,}6 + 0{,}9 * 0{,}4 = 0{,}42$
- $P(\text{Agua}|\text{Prueba+}) = (P(\text{Prueba+}|\text{Agua}) * P(\text{Agua})\ ) /\ P(\text{Prueba+}) = (0{,}9 * 0{,}6) /\ 0{,}58 =$ **0,93**
- $P(\text{NoAgua}|\text{Prueba-}) = (P(\text{Prueba-}|\text{NoAgua}) * P(\text{NoAgua})) /\ P(\text{Prueba-}) = (0{,}9 * 0{,}4) / 0{,}42 = 0{,}86$

Entonces, hay que quedarse con el resultado que refleje más probabilidad. En este caso, es más probable que haya agua. En el caso que se apliquen nuevas pruebas, identificándose como P_1, P_2, P_3 y P_4, se debe calcular la probabilidad de que haya agua sabiendo que todas las pruebas predijeron que hay agua. Es decir: *P(Agua|P_1+,P_2+,P_3+,P_4+)*. Tras aplicar el Teorema de Bayes, se llega a la conclusión de que:

$$P(A \mid b_1,b_2,b_3,b_4) = P(A) * (P(b_1 \mid A) * P(b_2 \mid A) * P(b_3 \mid A) * P(b_4 \mid A))$$

Si se generaliza la anterior fórmula, se obtiene:

$$P(A \mid b_1,b_2, \dots, b_n) = P(A) * (P(b_1,b_2, \dots, b_n \mid A) = P(A) * \prod_{i=1}^{n} P(a_i|A)$$

Para resolver el enunciado anteriormente planteado, se debe volver a calcular la probabilidad que haya agua y de que no haya agua, y nuevamente quedarnos con el mayor. Entonces, la fórmula final es:

$$\text{Solución} = \text{argmax}_{i=1}^{n} P(c_i) * \prod_{j=1}^{m} P(a_j|c_i)$$ (Fórmula de NaÏve Bayes)

4.2.1.1.1.2 Clasificador de entropía máximo (ME)

El clasificador Máxima Entropía [2] [3] [11] [19] [38] [39] [50] es un clasificador probabilístico que pertenece a la clase de modelos exponenciales. A diferencia del clasificador Naïve Bayes, ME no supone que las características sean condicionalmente independientes entre sí, el modelo usa la optimización

basada en búsquedas para hallar ponderaciones para las *características* que maximizan la probabilidad de los datos de entrenamiento. El clasificador ME se puede usar para resolver una gran variedad de problemas de clasificación de texto, como detección de idioma, clasificación de tema, Análisis de Sentimiento, entre otros.

ME se utiliza cuando la variable en cuestión es nominal, lo que significa que pertenece a cualquiera de un conjunto de categorías que no se pueden ordenar de manera significativa, y para la cual existen más de dos categorías. Este clasificador es una solución a los problemas de clasificación que utilizan una combinación lineal de las características observadas y algunos parámetros específicos del problema para estimar la probabilidad de cada valor particular de la variable dependiente.

En el procesamiento del lenguaje natural, los clasificadores ME sirven, comúnmente, como una alternativa a los clasificadores Naïve Bayes porque no presuponen la independencia estadística de las variables aleatorias que sirven como predictores. Sin embargo, el aprendizaje en dicho modelo es más lento que para un clasificador de Naïve Bayes, y por lo tanto puede no ser apropiado dada una gran cantidad de clases para aprender. En particular, el aprendizaje en un clasificador Naïve Bayes es una simple cuestión de contar el número de co-ocurrencias de características y clases, mientras que ME estima las distribuciones de probabilidad a partir de datos. El fundamento principal es que dicha distribución debe ser lo más uniforme posible.

La clasificación de texto con ME, comienza con pesos informativos (previos) mínimos y se optimiza para encontrar pesos que maximicen la probabilidad de los datos. El clasificador ME, toma en cuenta las correlaciones entre palabras, a diferencia de Naïve Bayes. Por ejemplo, el adjetivo "*faltante*" debería dar pesos más altos a los sustantivos, pero un clasificador Naive Bayes podría dar el mismo peso a un adjetivo si su frecuencia relativa fuera la misma que la de un sustantivo dado. ME convierte los conjuntos de características etiquetados en vectores usando codificación. Este vector codificado se usa para calcular los pesos de cada característica, que luego se pueden combinar para determinar la etiqueta más probable para un conjunto de características. Este clasificador se parametriza mediante un conjunto de *X{weights}*, que se utiliza para combinar las características conjuntas que se generan a partir de

un conjunto de características mediante una *X{encoding}*. En particular, la codificación mapea cada par *C{(featureset,label)}* a un vector. La probabilidad de cada etiqueta se calcula usando la siguiente ecuación:

$$P(fs|label) = dotprod(weights; encode(fs; label)) / sum(dotprod(weights; encode(fs; l))forlinlabels)$$

4.2.1.1.2 Clasificadores lineales

Estos clasificadores logran identificar a qué clase o grupo pertenece un objeto tomando una decisión basada en el valor de una combinación lineal sobre su conjunto de características. Entonces, si $\underline{X} = \{x_1 \dots x_n\}$ es la frecuencia de las palabras del documento normalizada, $\underline{A} = \{a_1 \dots a_n\}$ es un vector de coeficientes lineales con la misma dimensión que el espacio de característica, y *b* es un escalar; la salida del clasificador lineal se define como $p = \underline{A}.\underline{X} + b$. El predictor *p* es un hiperplano de separación entre diferentes clases. Estos clasificadores también se conocen como clasificadores discriminativos. Las funciones discriminativas mapean directamente la observación al valor de una etiqueta de clase. Si su objetivo principal es la predicción, los modelos discriminativos, que estiman directamente *P(Y|X)*, son empíricamente superiores a los generativos porque atacan el problema directamente. Sin embargo, se obtienen poco conocimiento sobre los datos y cómo se generan.

4.2.1.1.2.1 Clasificadores de Máquinas de Vectores de Soporte (SVM)

SVM [1] [2] [7] [11] [19] [40] [41] [43] [50] es un método que realiza tareas de clasificación mediante la construcción de hiperplanos en un espacio multidimensional que separa los casos de diferentes etiquetas de clase. El objetivo principal de las SVM es determinar los separadores en el espacio de búsqueda que mejor puedan separar las diferentes clases. Dado un conjunto de ejemplos de entrenamiento (de muestras) se pueden etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra. Intuitivamente, una SVM es un modelo que representa a los puntos de muestra en el espacio, separando de “forma óptima” las clases en espacios lo más amplios posibles mediante un hiperplano de separación definido como el

vector entre los diferentes puntos, de las diferentes clases, más cercanos al que se llama *vector soporte*. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de los espacios a los que pertenezcan, pueden ser clasificadas en algunas de las clases.

En ese concepto de “separación óptima” es donde reside la característica fundamental de las SVM: este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo. Por eso, a veces se les conoce a las SVM como *clasificadores de margen máximo*. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado.

Los datos de texto son ideales para la clasificación SVM, debido a la escasa naturaleza del texto, en el que pocas características son irrelevantes, pero tienden a correlacionarse entre sí y generalmente se organizan en categorías linealmente separables. SVM puede construir una superficie de decisión no lineal en el espacio de características original, mapeando las instancias de datos de forma no lineal a un espacio de producto interno, donde las clases se pueden separar linealmente con un hiperplano.

En la Figura 5, hay 2 clases, *x* y *o*, y hay 3 hiperplanos A, B y C. El hiperplano A proporciona la mejor separación entre las clases, porque la distancia normal de cualquiera de los puntos de datos es la más grande, por lo que representa el margen de separación máximo.

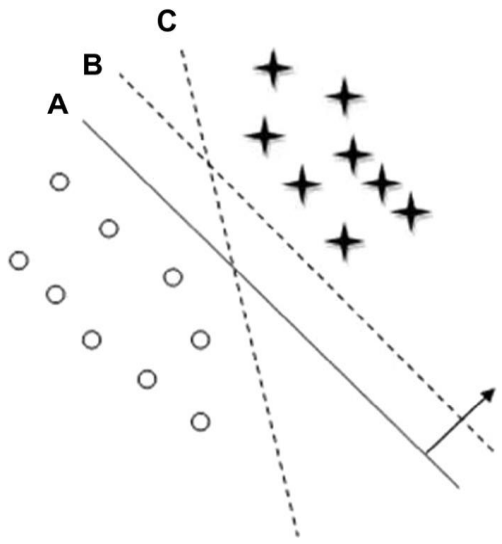


Figura 5. Uso de SVM en un problema de clasificación. [19]

Desafortunadamente los universos a estudiar no se suelen presentar en casos ideales de dos dimensiones como en el ejemplo anterior, sino que un algoritmo SVM debe tratar con a) más de dos variables predictoras, b) curvas no lineales de separación, c) casos donde los conjuntos de datos no pueden ser completamente separados, d) clasificaciones en más de dos categorías.

SVM permite utilizar las llamadas *funciones Kernel* (no lineales). Estas funciones resuelven el problema de clasificación trasladando los datos a un espacio donde el hiperplano de solución es lineal y, por tanto, más sencillo de obtener. Así, con una serie de datos de prueba se tendrá caracterizada la clasificación, es decir, la técnica SVM ha sido *entrenada*. Y tras este entrenamiento, se consigue un modelo en base al que podremos clasificar cualquier otro caso existente en el futuro, modelando los datos con una curva:

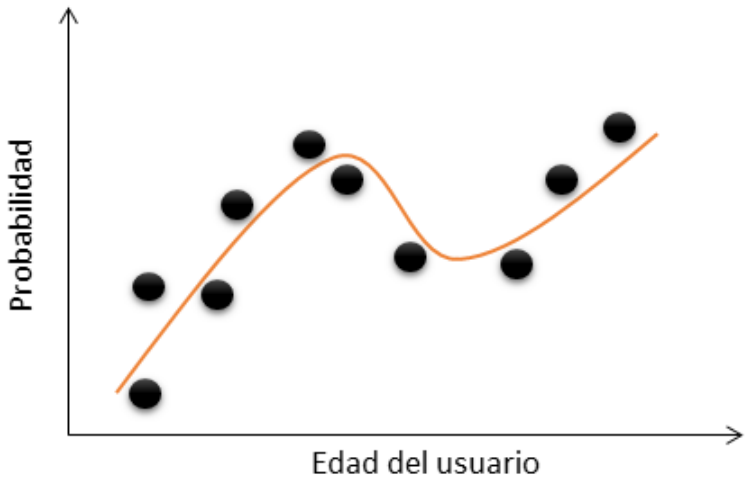


Figura 6. [40]

4.2.1.1.3 Clasificadores de árbol de decisión

El clasificador de árbol de decisión proporciona una descomposición jerárquica del espacio de datos de entrenamiento en el que se usa una condición del valor del atributo para dividir los datos. La condición o predicado es la presencia o ausencia de una o más palabras. La división del espacio de datos se realiza de forma recursiva hasta que los nodos hoja contienen un número mínimo de registros que se utilizan para fines de clasificación.

Hay otros tipos de predicados que dependen de la similitud de los documentos para correlacionar conjuntos de términos que pueden usarse para

una mayor partición de documentos. Los diferentes tipos de divisiones son: *división de atributo único (Single Attribute split)* que utiliza la presencia o ausencia de palabras o frases particulares en un nodo particular en el árbol para realizar la división; *división de atributos múltiples basada en la similitud (Similarity-based multi-attribute split)* utiliza documentos o clústeres de palabras frecuentes y la similitud de los documentos con estos grupos de palabras para realizar la división; y *división discriminatoria basada en atributos múltiples (Discriminat-based multi-attribute split)* utiliza discriminantes, por ejemplo el discriminante Fisher, que proyecta una línea con una dirección útil para la clasificación [34], para realizar la división.

4.2.1.1.4 Clasificadores basados en reglas

En los clasificadores basados en reglas, el espacio de datos se modela con un conjunto de reglas. El lado izquierdo representa una condición en el conjunto de características expresado en forma disyuntiva normal (es una estandarización de una fórmula lógica que es una disyunción de cláusulas conjuntivas), mientras que el lado derecho es la etiqueta de clase. Las condiciones están en la presencia del término. La ausencia del término rara vez se utiliza porque no es informativa en datos dispersos.

Existen diferentes criterios para generar reglas, la fase de entrenamiento construye todas las reglas dependiendo de estos criterios. Los dos criterios más comunes son el soporte y la confianza. El soporte es el número absoluto de instancias en el conjunto de datos de entrenamiento que son relevantes para la regla. La confianza se refiere a la probabilidad condicional de que se satisfaga el lado derecho de la regla si se satisface el lado izquierdo.

Tanto los árboles de decisión como las reglas de decisión tienden a codificar reglas en el espacio de características, pero el árbol de decisiones tiende a lograr este objetivo con un enfoque jerárquico. La principal diferencia entre estas dos técnicas es que el árbol es una partición jerárquica estricta del espacio de datos, mientras que los clasificadores basados en reglas permiten superposiciones en el espacio de decisión.

4.2.1.2 Débil, semi y sin supervisión de aprendizaje

El objetivo principal de la clasificación de texto es etiquetar los documentos en un cierto número de categorías predefinidas. Para lograr eso, se usa una gran cantidad de documentos de entrenamiento etiquetados para el aprendizaje supervisado, como se indicó anteriormente. En la clasificación de texto, a veces es difícil crear estos documentos de entrenamiento, pero es fácil recopilar los documentos sin etiqueta. Los métodos de aprendizaje no supervisados superan estas dificultades.

El concepto de débil y semi-supervisión se puede aplicar de diversas formas. Como se describe en [19] se planteó un método que divide los documentos en oraciones, y se categorizó cada oración usando listas de palabras clave de cada categoría y medida de similitud de oración. También se nombró una estrategia que funciona al proporcionar supervisión débil en el nivel de las características en lugar de las instancias, de la que obtuvieron un clasificador inicial al incorporar información previa extraída de un léxico de sentimiento existente en el aprendizaje del modelo del clasificador de sentimientos.

4.2.2 Enfoque basado en léxico

Hasta aquí, queda claro que las palabras y frases que transmiten sentimientos positivos o negativos son fundamentales para el análisis de sentimientos. Las palabras de opinión se emplean en muchas tareas de clasificación de sentimientos. Las palabras de opinión positiva se utilizan para expresar algunos estados o cualidades deseadas, mientras que las palabras de opinión negativa se usan para expresar algunos estados o cualidades no deseadas. Algunos ejemplos de palabras de sentimientos positivos son: “hermosas”, “maravillosas” y “sorprendentes”, mientras que algunos ejemplos de palabras de sentimientos negativos son: “malas”, “terribles” y “pobres”. Las palabras de opinión se pueden dividir en dos tipos, tipo base y tipo comparativo. Todas las palabras de ejemplo mencionadas anteriormente son del tipo base. Las palabras de sentimiento del tipo comparativo (que incluyen el tipo superlativo) se usan para expresar opiniones comparativas. Ejemplos de

estas palabras son: “mejor”, “peor”, etcétera, que son formas comparativas de sus adjetivos o adverbios base, por ejemplo: “bueno” y “malo”. A diferencia de las palabras del tipo base, las palabras del tipo comparativo no expresan una opinión regular sobre una entidad, sino una opinión comparativa sobre más de una entidad, como el ejemplo antes mencionado, *“las hamburguesas de Mostaza saben mejor que las de McDonald’s”*. Esta oración no expresa una opinión diciendo que cualquiera de las dos hamburguesas es buena o mala. Solo dice que comparado con McDonald’s, Mostaza sabe mejor.

Además hay frases de opinión y expresiones idiomáticas, modismos, por ejemplo, *“me salió un ojo de la cara”* al expresar que algo le costó caro, que juntas se llaman léxico de opinión. Hay tres enfoques principales para compilar o recolectar la lista de palabras de opinión: enfoque manual, enfoque basado en el diccionario y enfoque basado en corpus. El enfoque manual consume mucho tiempo y no se usa solo, por lo general, se combina con los otros dos enfoques automatizados como un control final para evitar los errores que resultaron de los métodos automatizados.

4.2.2.1 Enfoque basado en el diccionario

Una técnica simple en este enfoque es recolectar un pequeño conjunto de palabras de opinión manualmente con orientaciones conocidas, sean positivas o negativas. Luego, este conjunto crece al buscar sus sinónimos y antónimos en los diccionarios conocidos, como por ejemplo WordNet. Las palabras recién encontradas se agregan a la lista de *semillas* y luego comienza la siguiente iteración. El proceso iterativo se detiene cuando no se encuentran palabras nuevas. Una vez que se completa el proceso, se puede llevar a cabo una inspección manual para eliminar o corregir errores, además de asignar una peso de opinión a cada palabra utilizando un método probabilístico.

Observamos que la ventaja de utilizar un enfoque basado en diccionario es que uno puede encontrar fácil y rápidamente una gran cantidad de palabras de opinión con sus orientaciones. Aunque la lista resultante puede tener errores, se puede realizar una comprobación manual para limpiarla, lo que consume mucho tiempo, pero se realiza una sola vez. La principal desventaja es la incapacidad de encontrar palabras de opinión con el dominio y las orientaciones específicas del contexto. Como se expuso anteriormente,

muchas palabras de opinión tienen orientaciones dependientes del contexto. Por ejemplo, para un teléfono con altavoz, sí es silencioso, generalmente es negativo. Sin embargo, para un automóvil, si es silencioso, es positivo.

4.2.2.2 Enfoque basado en corpus

El enfoque basado en corpus ayuda a resolver el problema de encontrar palabras de opinión con orientaciones específicas del contexto. Sus métodos dependen de patrones sintácticos o patrones que ocurren junto con una lista de palabras de opinión para encontrar otras en un gran corpus. Existen restricciones para conectivos como *y*, *o*, *pero*, *cualquiera*, etcétera. También hay expresiones adversas tales como *pero*, *sin embargo*, que se indican como cambios de opinión. Para determinar si dos adjetivos combinados tienen la misma orientación o diferentes orientaciones, el aprendizaje se aplica a un corpus grande. Luego, los enlaces entre los adjetivos forman un gráfico y la agrupación se realiza en el gráfico para producir dos conjuntos de palabras: positivas y negativas.

Usar el enfoque basado en corpus solo, no es tan efectivo como el enfoque basado en el diccionario porque es difícil preparar un enorme corpus para cubrir todas las palabras del lenguaje. Pero este enfoque tiene una gran ventaja que puede ayudar a encontrar palabras de opinión específicas de contexto y sus orientaciones utilizando un corpus de dominio. El enfoque basado en corpus se lleva a cabo mediante el enfoque estadístico o el enfoque semántico como se ilustra en las siguientes subsecciones:

4.2.2.2.1 Enfoque estadístico

Encontrar patrones de co-ocurrencias o palabras de opinión de *semillas* se puede hacer usando técnicas estadísticas. Esto podría hacerse derivando polaridades posteriores utilizando la concurrencia de adjetivos en un corpus, como mencionan en [19]. Es posible usar todo el conjunto de documentos indexados en la web como el corpus para la construcción del diccionario. Esto supera el problema de la falta de disponibilidad de algunas palabras si el corpus utilizado no es lo suficientemente grande.

La polaridad de una palabra puede identificarse mediante el estudio de la frecuencia de ocurrencia de la palabra en un gran corpus anotado de textos. Si

la palabra aparece con mayor frecuencia entre los textos positivos, entonces su polaridad es positiva. Si ocurre con mayor frecuencia entre los textos negativos, entonces su polaridad es negativa. Si tiene frecuencias iguales, entonces es una palabra neutral.

Las palabras de opinión similares suelen aparecen con frecuencia juntas en un corpus. Por lo tanto, es probable que tengan la misma polaridad. De modo que, la polaridad de una palabra desconocida puede determinarse calculando la frecuencia relativa de la coincidencia con otra palabra.

El Análisis semántico latente (LSA) es un enfoque estadístico que se utiliza para analizar las relaciones entre un conjunto de documentos y los términos mencionados en estos documentos con el fin de producir un conjunto de patrones significativos relacionados con los documentos y términos.

4.2.2.2.2 Enfoque semántico

Este enfoque proporciona valores de opinión de forma directa y se basa en diferentes principios para calcular la similitud entre las palabras. Uno de estos principios, aporta valores de sentimiento similares para palabras semánticamente cercanas. WordNet, por ejemplo, proporciona diferentes tipos de relaciones semánticas entre las palabras usadas para calcular las polaridades del sentimiento. WordNet podría usarse también para obtener una lista de palabras de sentimiento al expandir iterativamente el conjunto inicial con sinónimos y antónimos y luego determinar la polaridad del sentimiento para una palabra desconocida por el recuento relativo de sinónimos positivos y negativos de esta palabra.

Como se menciona en [19], el enfoque semántico se usa en muchas aplicaciones para construir un modelo de léxico para la descripción de verbos, sustantivos y adjetivos que se utilizarán en SA. Este modelo consigue describir las relaciones detalladas de subjetividad entre los actores en una oración que expresa actitudes separadas para cada actor. Estas relaciones de subjetividad deben estar etiquetadas con información correspondiente tanto a la identidad del poseedor de la actitud como a la orientación (positiva vs. negativa) de la misma. Los resultados de la utilización de este enfoque, mostraron que la subjetividad del hablante y, en ocasiones, la subjetividad del actor puede identificarse de manera confiable.

Los métodos semánticos se pueden combinar con los métodos estadísticos para realizar el Análisis de Sentimientos. En una investigación nombrada en [19] se utilizó ambos métodos para encontrar la debilidad del producto a partir de las revisiones en la Web. Su buscador de debilidad extrajo las características y agrupó aquellas explícitas mediante el uso de un método basado en el morfema para identificar las palabras particulares de las revisiones. Se agrupó los productos con palabras en los aspectos correspondientes mediante la aplicación de métodos semánticos. Se empleó el método SA basado en oraciones para determinar la polaridad de cada aspecto en oraciones teniendo en cuenta el impacto de los adverbios de grado. Sus resultados afirmaron el buen desempeño del buscador de debilidad.

4.2.3 Herramientas actuales

Hoy en día, como mencionamos, muchas grandes corporaciones han desarrollado sus propias aplicaciones para el análisis de sentimientos, como por ejemplo Microsoft, Google, IBM, entre otras.

Empezando por Microsoft, provee una herramienta llamada *Text Analytics* [56]. Es un servicio basado en la nube que proporciona procesamiento de lenguaje natural avanzado sobre texto sin formato e incluye, para español: análisis de sentimiento y extracción de frases claves. Esta aplicación devuelve una puntuación de sentimiento entre 0 y 1 para cada documento, donde 1 es el más positivo. En cuanto a la extracción de frases clave, identifica rápidamente los puntos principales. Por ejemplo, para el texto "*La comida era deliciosa y había un personal maravilloso*", la aplicación devuelve los principales temas de conversación: "*comida*" y "*maravilloso personal*". La API de Text Analytics acepta datos de texto sin formato. El límite actual es de 5000 caracteres para cada documento.

Google por su lado, presentó Google Cloud Platform [57] que ofrece una gama completa y compleja de productos y servicios en la nube para informática, almacenamiento, redes, big data, aprendizaje automático, operaciones y más. La inteligencia artificial (IA) de Google Cloud proporciona servicios modernos de aprendizaje automático con modelos ya preparados

previamente y está basado en redes neuronales. Según la documentación de la aplicación, muchas de estas librerías o servicios, están en versión *Beta*.

Watson [58] es un sistema informático de inteligencia artificial desarrollado por la empresa estadounidense IBM. La corporación lo describe como "una aplicación de tecnologías avanzadas diseñadas para el procesamiento del lenguaje natural, la recuperación de información, la representación del conocimiento, el razonamiento automático, y el aprendizaje automático al campo abierto de búsquedas de respuestas", que es "para la generación de hipótesis, la recopilación de pruebas masivas, el análisis y la calificación."

Si bien todas las plataformas descritas corresponden a grandes empresas dentro de la ciencia computacional (por lo que es de esperarse que realicen un buen aporte), no fueron utilizadas para ser comparadas con nuestro desarrollo por diversos motivos. Primero y principal, nuestro objetivo fue el de realizar un estudio de NLP y SA, y para ello es necesario tener conocimiento de lo que sucede durante el proceso de transformación de la información. Todos los productos ofrecidos por estas compañías son API's, que reciben datos y devuelven resultados, por lo que un análisis de este tipo de productos, denominados "de caja negra", no es viable. A su vez, no es posible hacer uso del contexto, sólo analizan el texto que reciben, sin aprovechar el procesamiento del lenguaje realizado ni la información que se pueda extraer de las publicaciones o los comentarios, como las reacciones.

Además, los desarrollos propuestos por Google y Microsoft, no son gratuitos. Ofrecen ambos una prueba gratuita y permiten 500 consultas. En el caso de Google, es por 12 meses, con un crédito de $300, mientras que en el caso de Microsoft, es de 30 días y con un crédito de $3150. Te obligan a brindar datos personales como una dirección, un teléfono y datos de una tarjeta de crédito o débito.

Por último, consideramos que su uso no es amigable e intuitivo para el usuario final. Este debería tener conocimientos de programación para poder utilizar las APIs. En cuanto a nuestro objetivo, es brindar un servicio donde esto sea transparente al usuario.

4.3 Performance

Luego de realizar la limpieza de los datos de entrada y la selección de características más relevantes, es momento de la implementación y entrenamiento de un modelo, para obtener resultados. Éstos pueden darse en forma de probabilidad o de clases, y a partir de aquí es necesario realizar un análisis de desempeño o *performance* del modelo, para saber qué tan exitoso fue el entrenamiento.

Diferentes métricas son utilizadas para evaluar distintos algoritmos de aprendizaje automático. Dada la naturaleza de nuestro trabajo, aquí nos centramos solo en aquellas utilizadas para los problemas de clasificación.

Las métricas elegidas para evaluar un modelo de aprendizaje automático son de vital importancia. Dicha elección influencia la manera en la que el desempeño es medido y comparado, por lo que es necesario conocer la finalidad de cada métrica para poder utilizarlas de forma adecuada y no obtener mediciones erróneas o ficticias.

4.3.1 Validación cruzada de K iteraciones

El entrenamiento, testeo y estimación de la performance pueden ser realizados de diversas formas. Por lo general, mientras mayor sea el volumen de datos que recibe un modelo, mayor es la información con la que cuenta para entrenarse, y mejores suelen ser los resultados. Pero en las aplicaciones reales, solo se tiene acceso a un conjunto finito de ejemplos, y por esto es necesario tomar decisiones acerca de cuáles utilizar para entrenar, y cuáles para testear y verificar el modelo resultante.

En un principio podría pensarse en utilizar la totalidad de los datos como entrenamiento, de forma tal de proporcionar la mayor cantidad de información posible. Pero este primer enfoque tiene una contra fundamental: a la hora de realizar el testeo, el modelo puede sufrir un sobreajuste, por lo que los valores de performance obtenidos no serán representativos. El sobreajuste se define como una condición en la cual el algoritmo de aprendizaje puede quedar

ajustado a unas características muy específicas, obteniendo una precisión muy elevada para el conjunto de entrenamiento, pero muy baja para casos externos.

Una alternativa mejor consiste en dividir los datos en conjuntos disjuntos: entrenamiento y testeo. Una de las técnicas comúnmente utilizada para realizar esta división se denomina Validación cruzada de K iteraciones (***K-Fold Cross-validation***). Esta última consiste, como su nombre bien indica, en realizar un número *K* iteraciones sobre el conjunto de datos, los cuales también se dividen aleatoriamente en *K* subconjuntos.

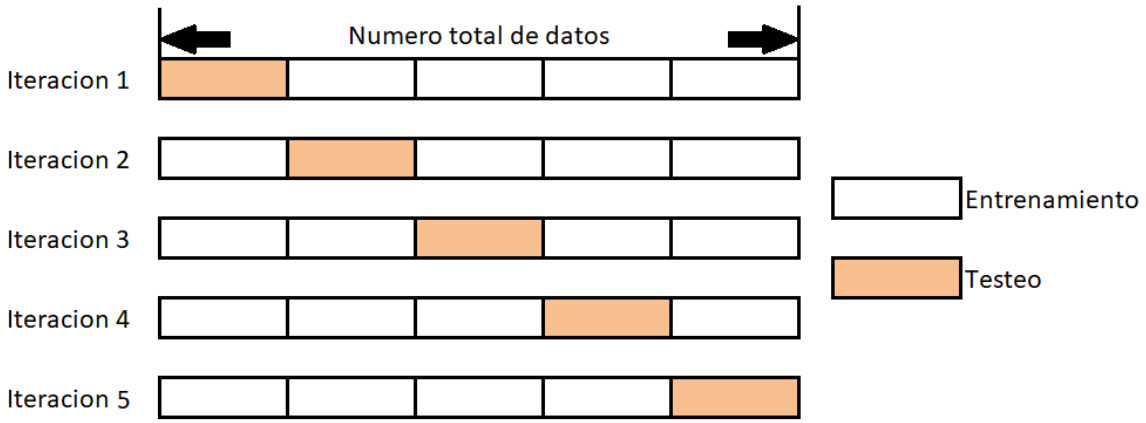


Figura 7. Forma en la que se seleccionan las particiones para la validación cruzada.

En cada una de las iteraciones, se selecciona un grupo distinto para testeo, y los *K*-1 grupos restantes para entrenamiento. De esta forma, eventualmente todos los datos habrán sido utilizados tanto para entrenar como para testear. En cada una de las *K* iteraciones se realiza un cálculo de error E_i, y el resultado final lo obtenemos a partir de realizar la media aritmética de los *K* valores de errores obtenidos, según la fórmula:

$$E = \frac{1}{K} \Sigma_{i=1}^{K} E_i$$

4.3.2 Matriz de confusión

La matriz de confusión es una de las métricas más intuitivas y simples utilizadas para obtener la correctitud y precisión de un modelo. Es utilizada en problemas de clasificación donde la salida puede ser 2 o más clases.

Para facilitar la comprensión de cada concepto, utilizaremos el siguiente ejemplo: supongamos que tenemos un problema de clasificación, donde estamos prediciendo si un comentario es positivo o negativo. Para este caso, la matriz de confusión será una tabla con dos dimensiones: “Real” y “Predicción”, y un conjunto de clases en cada una de ellas.

		Real	
		Pos	Neg
Predicción	Pos	**VP**	**FP**
	Neg	**FN**	**VN**

Tabla 4. Tabla ejemplo matriz de confusión.

VP = Verdadero Positivo

VN = Verdadero Negativo

FP = Falso Positivo

FN = Falso Negativo

La matriz de confusión por sí sola no es una medida de desempeño, pero prácticamente todas las métricas se basan en los valores contenidos en dicha matriz.

Términos asociados con la matriz de confusión:

1. **Verdaderos Positivos (VP):** Son los casos donde coinciden la predicción de una clase positiva con el valor real de ese caso. (Ej: un comentario positivo, etiquetado como “positivo”).

2. Verdaderos Negativos (VN): Se da cuando se predijo que un caso pertenece a la clase “negativo”, y dicha predicción coincide con la clase real del dato. (Ej: un comentario negativo, etiquetado como “negativo”).

3. Falsos Positivos (FP): Éste término es utilizado para identificar a todas aquellas predicciones que tuvieron como resultado la clase “positivo”, pero que

en realidad la clase a la que pertenecen es “negativo”. (Ej: un comentario negativo, etiquetado como “positivo”).

4. Falsos Negativos (FN): Por último, los negativos falsos son aquellos casos en los que el modelo predice que la clase es “negativo”, pero en realidad el dato pertenece a la clase “positivo”. (Ej: un comentario positivo, etiquetado como “negativo”).

Con estos conceptos en mente, es relativamente simple generar una matriz similar a la de la Tabla 4, a partir de una matriz de confusión con más de 2 clases. Para lograrlo se aplica el mismo método descrito en [55], en el que teniendo 3 clases C_1, C_2 y C_3 dispuestos de manera tal que las columnas son las clases reales y las filas las predicciones, los cuatro términos quedan como sigue:

- **VP:** Todos los casos en los que elementos de la clase C_1 fueron etiquetados como C1, es decir, donde coinciden la columna y la fila de C_1.
- **VN:** Todos los casos donde los elementos que no son C_1 (sea C_2 o C_3), no fueron etiquetados como C_1. Es decir, todos los elementos que no se encuentran ni en la columna ni en la fila de C_1.
- **FP:** Todos los elementos que no son C_1, pero fueron etiquetados como tal. Serían todos los elementos de la fila de C_1 (sin contar la columna C_1, obviamente).
- **FN:** Todos los elementos que son C_1, pero fueron etiquetados como C_2 o C_3. Representan la columna de C_1, sin contar la fila C_1.

Finalmente, para obtener estos 4 valores para las clases C_2 y C_3, se repite el procedimiento, simplemente intercambiando las clases. A continuación en las tablas 5 y 6 se muestra a modo de ejemplo cómo interpretar las celdas para las clases C_1 y C_2 respectivamente.

		Real		
		C_1	C_2	C_3
Predicción	**C_1**	**VP**	**FP**	**FP**
	C_2	**FN**	**VN**	**VN**
	C_3	**FN**	**VN**	**VN**

Tabla 5. Tabla ejemplo matriz para la clase C_1.

		Real		
		C_1	C_2	C_3
Predicción	C_1	**VN**	**FN**	**VN**
	C_2	**FP**	**VP**	**FP**
	C_3	**VN**	**FN**	**VN**

Tabla 6. Tabla ejemplo matriz para la clase C_2.

Por supuesto, el escenario ideal sería aquel en el que en la matriz de confusión, los casos de Falso Positivo y Falso Negativo sean 0. Pero la mayoría de los modelos no llegan a predecir de forma perfecta.

¿Cuándo minimizar cada valor?

Como ya sabemos, prácticamente todo modelo posee un grado de error en sus predicciones respecto al valor real. Esto resulta en Falsos Positivos y Falsos Negativos.

No existe una regla de oro que nos diga cuál valor minimizar en todas las situaciones, esto depende pura y exclusivamente de la naturaleza y el contexto de lo que se quiera analizar. A raíz de ello, podríamos querer minimizar uno de éstos dos valores.

Minimizar Falsos Negativos:

Supongamos el siguiente caso de ejemplo, donde tenemos 100 personas, de las cuales solo 5 están enfermas. En este caso queremos clasificar correctamente todos los positivos, ya que, hasta en un mal modelo (que prediga todos los casos como negativos), obtendremos un 95% de precisión. Pero para poder capturar todos los casos en los que las personas están enfermas, podríamos terminar clasificando personas sanas de forma errónea. Éste es un riesgo que es aceptable tomar para el caso de ejemplo planteado, pues es menos peligroso predecir a alguien sano como enfermo (quien va a ser enviado para un examen más exhaustivo), que ignorar a alguien enfermo y etiquetarlo como sano. De este modo, dicho paciente sería eliminado del grupo de análisis y no recibiría atención alguna.

Minimizar Falsos Positivos:

Para éste caso, utilizaremos otro ejemplo, donde el modelo clasifica si un correo electrónico es basura o no.

Supongamos que el modelo clasifica un correo importante como basura (Falso Positivo). Esto es peor que etiquetar un correo basura como importante, o no etiquetarlo como *spam*, porque en dicho caso podríamos verlo y eliminarlo manualmente, y no sería tedioso, si sucede de vez en cuando. Pero perder un correo importante, sí es una situación que es preferible evitar. Por lo que en el caso de la clasificación de correos basura, minimizar los Falsos Positivos es más importante que los Negativos Falsos.

4.3.3 Certeza

Certeza o correctitud (*accuracy*): Es la proporción de predicciones correctas respecto del total de predicciones

		Real	
		Pos	Neg
Predicción	Pos	**VP**	**FP**
	Neg	**FN**	**VN**

Tabla 7. Tabla ejemplo matriz de confusión para calcular certeza.

Certeza = $(\mathrm{VP} + \mathrm{VN}) \ / \ (\mathrm{VP} + \mathrm{FP} + \mathrm{FN} + \mathrm{VN})$

En el numerador, tenemos las predicciones correctas (marcadas en rojo en la tabla), y en el denominador tenemos todas las predicciones realizadas por el modelo.

¿Cuándo utilizar la certeza?

La certeza es una buena métrica cuando las clases de los datos de entrada están aproximadamente balanceadas. (Ej.: 60% de los comentarios son positivos y 40% negativos).

¿Cuándo no utilizar la certeza?

Nunca debe tomarse en cuenta la certeza cuando la mayoría de los datos pertenecen a una sola clase (clases altamente desbalanceadas).

Ejemplo: en el caso de las personas enfermas, de 100 personas solo 5 están afectadas. Suponiendo que se tiene un modelo muy malo, el cual predice todos los casos como negativos, aun cuando éste es completamente inútil prediciendo pacientes enfermos, tendría una certeza del 95%.

4.3.4 Precisión

Precisión (*precision*): Es la proporción de predicciones correctas respecto del total de predicciones para esa clase.

Es decir, de todas las predicciones para la clase *X*, qué porcentaje fue correcto.

		Real	
		Pos	Neg
Predicción	Pos	**VP**	**FP**
	Neg	**FN**	**VN**

Tabla 8. Tabla ejemplo matriz de confusión para calcular precisión.

Precisión = $\mathrm{VP} / (\mathrm{VP} + \mathrm{FP})$

Volviendo al ejemplo de los comentarios, digamos que de 100 comentarios, solo 10 son positivos, y se tiene un modelo que predice todos los comentarios como positivos. Dado que todas las predicciones van a tener como resultado la clase *"positivo"*, el denominador va a ser 100, y el numerador (comentarios positivos reales) 10. Por lo que dicho modelo tendría una precisión del 10%.

4.3.5 Exhaustividad

Exhaustividad (*recall*): Es la métrica que nos permite conocer la proporción de los casos positivos cuya predicción fue acertada. Es decir, de

todos los casos pertenecientes a la clase *X*, qué porcentaje fue correctamente predicho.

		Real	
		Pos	Neg
Predicción	Pos	**VP**	**FP**
	Neg	**FN**	**VN**

Tabla 9. Tabla ejemplo matriz de confusión para calcular exhaustividad.

Exhaustividad = $\mathrm{VP} / (\mathrm{VP} + \mathrm{FN})$

Continuando con el ejemplo donde se tiene 100 comentarios, 10 de los cuales son positivos. Si se tiene un modelo que predice absolutamente todos los casos como positivos, el denominador (VP + FN) sería 10, y el numerador (casos positivos reales) también. Por lo que dicho modelo tendría una exhaustividad del 100%, y una precisión del 10%.

¿Cuándo utilizar la precisión y cuándo la exhaustividad?

Está claro que la exhaustividad nos brinda información acerca del desempeño del modelo respecto a los falsos negativos (cuántos erramos), mientras que la precisión lo hace respecto de los falsos positivos (cuántos acertamos).

Por lo tanto, si nuestra intención es minimizar los falsos negativos, tendremos que lograr que la exhaustividad se aproxime al 100% sin descuidar la precisión. Por el contrario, si nuestro foco está puesto en minimizar los falsos negativos, debemos incrementar la precisión lo máximo posible.

4.3.6 Especificidad

Especificidad (*specificity*): (Opuesto a exhaustividad) Es la proporción de casos que no pertenecen a la clase *X*, y que fueron correctamente predichos como otra clase.

		Real	
		Pos	Neg
Predicción	Pos	**VP**	**FP**
	Neg	**FN**	**VN**

Tabla 10. Tabla ejemplo matriz de confusión para calcular especificidad.

Especificidad = $\mathrm{VN} / (\mathrm{VN} + \mathrm{FP})$

Siguiendo con el mismo ejemplo, (100 comentarios, 10 positivos y un modelo que predice todo como positivo), tendremos como denominador 90, y como numerador 0. Por lo que la especificidad de nuestro modelo sería de 0% (exactamente el opuesto de la exhaustividad).

4.3.7 Valor-F

No siempre se quiere tener por separado la precisión y la exhaustividad cada vez que se realiza un modelo para resolver un problema de clasificación. Por este motivo sería mejor si se obtiene un único valor que represente ambas.

Una primer aproximación sería calcular la media aritmética, (P + E) / 2, donde P es Precisión y E es Exhaustividad, pero como podemos imaginar, existen situaciones en las que éste valor no será de utilidad.

Tomando como caso el ejemplo ya mencionado, se tendría:

		Real	
		Pos	Neg
Predicción	Pos	**10**	**90**
	Neg	**0**	**0**

Tabla 11. Tabla ejemplo matriz de confusión para calcular promedio armónico.

Precisión = 10 / 100 = 10%

Exhaustividad = 10 / 10 = 100%

Ahora, si calculamos la media aritmética entre estos dos valores, obtendremos 55%. Éste valor parecería ser demasiado razonable para un modelo tan malo (ya que todas las predicciones son positivas).

Necesitamos un valor más ponderado, y éste es el promedio armónico. El promedio armónico está dado por la siguiente fórmula:

Promedio armónico = $2xy / x + y$

Este promedio es similar al promedio aritmético cuando *x* e *y* poseen valores similares. Pero cuando *x* e *y* son diferentes, el resultado se aproxima al menor de los valores.

Para el ejemplo:

$$\text{Valor-F} = \text{Promedio armónico}(\text{Precisión}, \text{Exhaustividad})$$

$$\text{Valor-F} = 2 * \text{Precisión} * \text{Exhaustividad} / (\text{Precisión} + \text{Exhaustividad})$$

$$\text{Valor-F} = 2 * 10 * 100 / 110$$

$$\text{Valor-F} = 18.18\%$$

Por lo que si un valor, tanto de la precisión como de la exhaustividad, es realmente bajo, el Valor-F se aproxima a éste, dando un resultado más apropiado para el modelo en lugar de tan solo el promedio aritmético.

Capítulo 5

Desarrollo propuesto

5.1 Introducción

Como ya mencionamos anteriormente, las redes sociales son fuente de una gran cantidad de información generada por los propios usuarios. Por sí solo, todo este material no sería de mucha utilidad, más allá de permitirles a los usuarios interactuar con las redes. Cuando se combinan estos volúmenes de datos con herramientas de recolección y extracción automática de la información, se abren las puertas a un nuevo campo de estudio sobre estadísticas y análisis tanto de sentimientos como de mercado.

También describimos algunos de los obstáculos más comunes que suelen presentarse a la hora de realizar dicho procesamiento, a raíz de las características propias del lenguaje humano y más aún por la informalidad con la que se utiliza en las redes sociales. Para poder abordar dichos obstáculos, la comunidad académica fue desarrollando diversas técnicas, producto de años de estudio en la materia. Algunas de estas técnicas fueron explicadas en los capítulos anteriores, tanto de Procesamiento del Lenguaje Natural, como de Análisis de Sentimientos.

Teniendo en cuenta que la mayor parte de la bibliografía hoy en día está en idioma inglés, y que los estudios realizados fueron basados en textos en ese mismo idioma, decidimos plantearnos como objetivo, en este trabajo de tesis, el análisis y la comparación de diversas técnicas de procesamiento de lenguaje natural y clasificación aplicado sobre textos en español. Para ello, tomamos como corpus comentarios en español provenientes de la red social Facebook.

Para poder llevar a cabo este estudio, fue necesario el diseño e implementación de una aplicación que permite:

- Recolectar comentarios conectándose con la API de Facebook y almacenarlos en una base de datos.

- Etiquetarlos de manera personal (por el motivo detallado en la sección 1.2).

- Analizar el corpus, realizando un preprocesamiento del mismo, que comprende tanto la limpieza de texto como la obtención de palabras propensas a tener mayor contenido de opinión.

- Entrenar y aplicar modelos basados en las técnicas de clasificación.

- Evaluar y comparar el rendimiento de los diversos modelos.

Cada uno de estos puntos será abordado con mayor nivel de detalle en la sección 5.4 de éste capítulo.

5.2 Servicio utilizado

5.2.1 ¿Que es una API?

Las siglas API vienen del inglés: *Application Programming Interface*, que en español significa Interfaz de Programación de Aplicaciones. Consiste básicamente en una capa de abstracción que ofrece cierta biblioteca o servicio web para que un conjunto de funciones, métodos y procedimientos sean utilizados por diversos sistemas y aplicaciones.

Se define como una capa de abstracción, ya que el software cliente (que es quien consume estas funciones) sabe *qué hace*, pero no *cómo lo hace*. Es decir, no tiene conocimiento de la implementación de las mismas, sino que solo conoce la forma de realizar la petición, la interfaz de entrada (parámetros de entrada), los datos devueltos y el formato o estructura en la que son provistos.

Una de las principales características de las API's consiste en que los servicios proporcionados son de uso general, de esta manera se amplía la cantidad y diversidad de sistemas que pueden sacar provecho, evitando el trabajo de implementar ciertas funcionalidades.

5.2.2 API Graph - Facebook

La API Graph de Facebook es la principal forma en la que las aplicaciones pueden ingresar y extraer datos desde y hacia la plataforma de Facebook. Está

basada en el protocolo de comunicación HTTP mediante el cual las distintas aplicaciones se comunican con la API para consultar datos, publicar nuevas historias, administrar anuncios, subir fotos, entre tantas otras tareas.

El nombre API Graph proviene del concepto de “grafo social”, es decir, una representación de la información de Facebook que está compuesta por:

- **Nodos:** objetos individuales con identificador único, por ejemplo, un usuario, una foto, una página o un comentario.
- **Aristas:** conexiones entre una colección de objetos y un objeto único, por ejemplo, las fotos de una página o los comentarios de una foto.
- **Campos:** datos sobre un objeto, por ejemplo, el cumpleaños de un usuario o el nombre de una página.

5.2.3 Tokens de acceso

Para realizar cualquier tipo de interacción con la API, es necesario poseer un ***Token de Acceso***, el cual permiten realizar generalmente dos funciones: Acceder a la información de un usuario sin necesidad de proporcionar su contraseña e identificar la aplicación, el usuario y los tipos de datos a los que se les permite el acceso.

Los ***Tokens*** implementan el protocolo OAuth 2.0 [51], lo que permite que tanto los usuarios como las páginas les otorguen autorizaciones, que son utilizadas posteriormente por las aplicaciones para acceder a información específica.

Los identificadores de usuario y aplicación vienen codificados en el propio token, por lo que puede hacerse un seguimiento de cuáles son los datos a los que el usuario dio permiso de acceso a la aplicación.

Token de Acceso

Un token de acceso es una cadena que identifica a un usuario, aplicación o página, y que la aplicación puede utilizar para realizar llamadas a la API Graph. El token incluye información acerca de su caducidad y de la aplicación que lo generó. Para efectos de las comprobaciones de privacidad, la mayoría de las

llamadas a la API en Facebook deben incluir un token de acceso. Según el caso de uso, hay distintos tipos de token de acceso que se pueden usar:

Token de acceso de Usuario: es el más común. Se necesita cada vez que la aplicación solicita a una API que lea, modifique o escriba los datos de Facebook de una persona en nombre de esta. Se obtienen cuando se inicia la sesión y requieren que la persona conceda permiso a la aplicación.

Token de acceso a la aplicación: es necesario para modificar y leer la configuración de la aplicación. También se puede utilizar para publicar acciones de Open Graph[3]. Se genera usando una clave secreta acordada previamente entre la aplicación y Facebook, y más tarde se utiliza durante las llamadas que cambian la configuración general de una aplicación. El token de acceso a la aplicación se obtiene mediante una llamada de servidor a servidor.

Token de acceso a la página: es similar a los de usuario, salvo por el hecho de que proporcionan permiso para que las API lean, escriban o modifiquen los datos pertenecientes a una página de Facebook. Estos tokens de acceso son únicos para cada página, administrador y aplicación. Para obtener un token de acceso a la página, primero se debe obtener un token de acceso de usuario y solicitar el permiso manage_pages. Luego, se obtiene el token de acceso a la página mediante la API Graph.

Token de cliente: es un identificador que se puede insertar en aplicaciones binarias nativas para celulares o en aplicaciones para computadoras y sirve para identificar la aplicación. No está diseñado para ser un identificador secreto porque se inserta en las aplicaciones. El token de cliente se utiliza para acceder a las API del nivel de la aplicación, aunque solo a un subconjunto muy limitado.

Un aspecto importante acerca de los tokens de acceso es que son portátiles. Una vez que se obtiene uno, se puede utilizar para realizar llamadas

[3] Es una plataforma basada en aspectos sociales de los usuarios y que cualquier empresa puede usarlo. Por ejemplo, Spotify si iniciaste sesión con Facebook usa el Open Graph para publicar lo que estás escuchando, de tal manera que socializas un contenido de algo que estás haciendo.

desde un cliente móvil, un navegador web o un servidor a los servidores de Facebook. Para el desarrollo de nuestro trabajo, obtuvimos un token de acceso sin caducidad, tal como se describe en [46].

5.2.4 Datos disponibles desde la API

Las operaciones de lectura comienzan por un **nodo**. Por ejemplo, para obtener los datos del nodo correspondiente a la página del portal de noticias "Todo Noticias", la petición necesaria sería la siguiente:

```
GET https://graph.facebook.com/28963119862
```

La cual devuelve los siguientes campos en formato JSON:


```
{
  "name": "TN Todo Noticias",
  "id": "28963119862"
}
```


Todo **nodo** tiene **perímetros** o **aristas**, que normalmente devuelven colecciones de otros nodos.

Para acceder al perímetro, es necesario especificar el identificador del nodo, y el nombre del perímetro en la ruta de acceso. Siguiendo con nuestro ejemplo anterior, podemos acceder al perímetro **feed** con la siguiente petición:

GET https://graph.facebook.com/28963119862/feed

Y la respuesta JSON sería la siguiente:


```
{
  "data": [
    {
      "message": "En la novena jornada del juicio por el crimen de Fernando Pastorizzo, declararon los padres de Nahir Galarza entre contradicciones y llantos. Crece la expectativa por el testimonio de la acusada Por Sebastian Domenech",
      "created_time": "2018-06-22T01:56:46+0000",
      "id": "28963119862_10157476229169863"
    },
    {
      "message": "TREMENDO El audio viral de Simeone que aniquila a Messi, Sampaoli y la AFA por la derrota con Croacia",
      "created_time": "2018-06-21T22:32:43+0000",
      "id": "28963119862_10157475814189863"
    }
  ]
}
```


En la que se puede observar una colección de publicaciones realizadas por el usuario TodoNoticias en su muro.

De igual manera que con el perímetro ***feed***, a través de la API es posible obtener gran cantidad de datos. Prácticamente cualquier información que podamos obtener navegando y utilizando la aplicación de facebook, también es obtenible mediante una petición a la API. Además de los datos propios de una página (perfil, aplicación, etcétera), es posible consultar por métricas o estadísticas de la misma obtenidas en base a algún parámetro. Por ejemplo: el número total de personas que indicaron que le gusta la página o de personas que compartieron historias sobre ella. Algunos ejemplos concretos son:

- **page_content_activity_by_age_gender_unique:** Número de personas que están hablando de la página, por edad y sexo del usuario.
- **post_activity:** Número de historias generadas acerca de una publicación de la página.
- **page_post_engagements:** Número de veces que las personas interactuaron con las publicaciones al indicar que les gustan, comentarlas, compartirlas, etcétera.
- **page_actions_post_reactions_total:** Total de reacciones diarias en las publicaciones de la página por tipo.
- **post_reactions_by_type_total:** Total de reacciones a una publicación en particular, por tipo.

5.2.5 Datos recolectados

Como vimos, la API de Facebook deja a nuestra disposición prácticamente cualquier información accesible por un usuario común, pero a fin de simplificar el procesamiento de la misma, en el presente trabajo nos limitamos a recolectar únicamente los siguientes datos:

- Publicaciones del muro de la página Todo Noticias.
- Comentarios realizados a dichas publicaciones.
- Cantidad de *likes* de cada comentario.

Las publicaciones son el objeto principal sobre el cual el usuario manifiesta sus sentimientos de distintas formas, ya sea mediante un comentario textual o una reacción predefinida.

Los comentarios son el corpus que utilizaremos para trabajar, sometiéndolo a procesos de limpieza de texto, extracción de características y como entrada para las técnicas de aprendizaje automático.

La cantidad de *likes* que posee cada comentario nos serviría para ponderar el impacto de la polaridad de un comentario sobre su publicación. Ya que si un comentario recibe numerosos *likes*, se entiende que un conjunto de usuarios opina de la misma manera, por lo que de ésta forma, estamos tomando en cuenta su opinión, a pesar de que no la expresaron textualmente.

Estructura de los datos obtenidos

Como vimos en la sección 5.2.2, toda la información disponible desde la API Graph está compuesta por elementos de 3 tipos: Nodos, Aristas o Campos.

Los datos obtenidos para el presente trabajo, poseen la siguiente estructura:

Nodo: "TodoNoticias"{
  Arista: "feed" (Colección de publicaciones) {
    "data": [
      {
      Campo: "message" (Mensaje de cada publicación),
      Campo: "created_time" (fecha de publicación),
      Nodo: "id" (identificador único de la publicación)
      },
      ……
    ]
  }
}

Luego, utilizando el nodo "id" de cada publicación, realizamos una consulta a la API consultando por la arista "comments", para así obtener la colección de comentarios realizados sobre esa publicación. Ésta colección posee una estructura muy similar a "feed", la cual dentro del campo "data" devuelve una lista con el id, el mensaje del comentario y la fecha.

5.3 Arquitectura de la aplicación

El desarrollo del software para el presente trabajo fue implementado respetando una arquitectura multinivel (a menudo llamada *arquitectura de n-niveles*). Su nombre proviene de la separación tanto física como lógica del manejo de los datos, el procesamiento y la presentación. La división más simple que puede aplicar para un desarrollo en 3 capas: capa de presentación, capa de lógica de negocio y capa de datos.

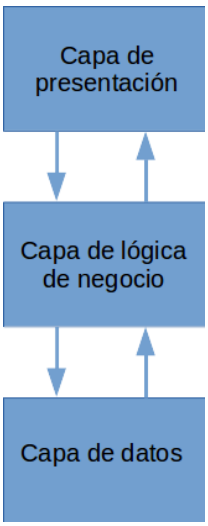


Figura 8. Disposición y comunicación de las distintas capas de la aplicación.

Capa de presentación: El nivel más alto de la aplicación es la interfaz gráfica. Su principal función es capturar la interacción del usuario y comunicarle la información de una manera entendible y fácil de usar.

Capa de lógica de negocio: Esta capa controla la funcionalidad de la aplicación, procesa las peticiones del usuario y envía las respuestas a la capa de presentación. Se denomina capa de lógica de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Capa de datos: Aquí la información es almacenada y recuperada desde la base de datos. Se comunica únicamente con la capa de lógica de negocio.

5.3.1 Capa de datos

Como sabemos, la capa de datos es la responsable de persistir, almacenar y recuperar toda la información. Para ello es necesario un motor de base de datos que se encargue de realizar todas estas tareas.

Motor de base de datos

El motor de base de datos utilizado en nuestro desarrollo es MySQL. Nuestra elección se basó principalmente en la robustez, eficiencia y popularidad de ésta tecnología. Según una encuesta realizada en el 2017 [44], MySQL es utilizado por aproximadamente el 50% de los desarrolladores. Estos motivos, en conjunto con nuestra experiencia previa utilizando esta tecnología, fueron determinantes a la hora de su elección.

Además del motor de base de datos, es necesario plantear una organización de la información, y para ello es necesario realizar un modelado de datos. El modelado de datos es un proceso utilizado para analizar y definir los requerimientos de la capa de lógica de negocio y que tiene como resultado una estructura la cual garantiza que toda la información relevante para el sistema será persistida y almacenada.

Modelo de datos

Debido a que el centro de nuestro trabajo es el procesamiento de texto, y el análisis de técnicas de clasificación, el modelo de datos se mantiene relativamente simple. A nivel de base de datos contamos con 3 tablas: Posts, Comments y Models.

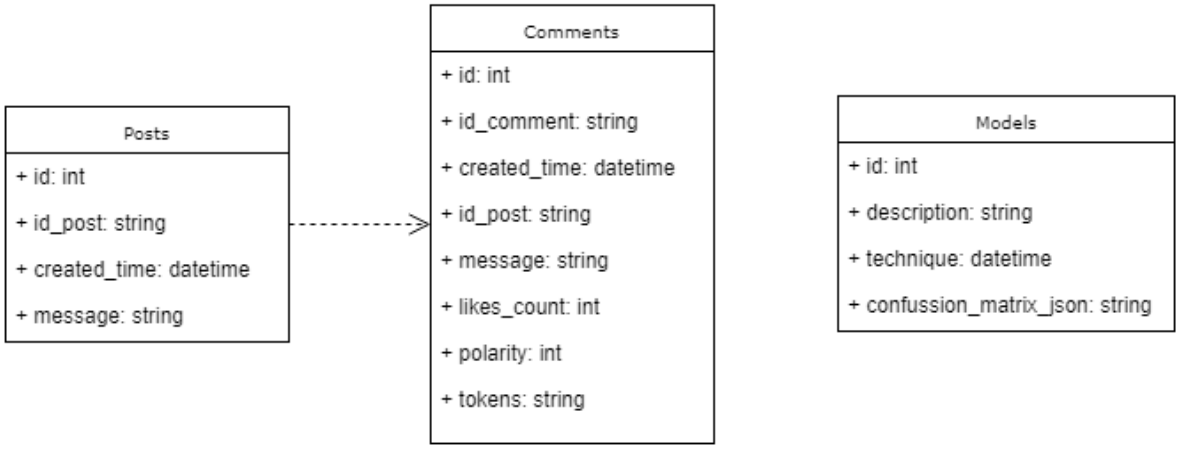


Figura 9. Detalle de tablas y campos persistidos en la base.

De las publicaciones (tabla Posts) se almacena:

- **id_post:** identificador único de una publicación obtenido con la API.
- **created_time:** fecha en la que fue creada la publicación.
- **message:** texto de la publicación.

Para los comentarios (tabla Comments), se persiste:

- **id_comment:** identificador único de un comentario obtenido con la API.
- **created_time:** fecha de creación del comentario.
- **id_post:** identificador correspondiente a la publicación en la cual fue redactado dicho comentario.
- **message:** texto del comentario.
- **likes_count:** cantidad de “me gusta” que posee el comentario.
- **polarity:** la polaridad que le fue asignada, ya sea de forma manual o automática.
- **tokens:** lista de palabras consideradas como más relevantes, producto del preprocesamiento del texto.

La información persistida referente a los modelos (tabla Models) es la siguiente:

- **description:** texto que identifica a qué ensayo pertenecen los datos de dicho registro
- **technique:** siglas de la técnica utilizada para entrenar dicho modelo

- **confussion_matrix_json:** matriz de confusión almacenada en formato JSON

5.3.2 Capa de lógica de negocio

En esta sección describiremos todas las tecnologías utilizadas para el procesamiento de la información.

PYTHON

El lenguaje de programación utilizado para el desarrollo de la aplicación fue Python. Éste es un lenguaje simple pero potente, con excelentes funcionalidades para el procesamiento de información lingüística. A simple viste puede apreciarse que es un lenguaje altamente legible, tanto es así, que leyendo el código es relativamente simple entender lo que hace, aún si el lector no es un programador.

Elegimos Python porque posee una curva de aprendizaje poco pronunciada, su sintaxis y semántica son transparentes, y posee un buen manejo de cadenas de caracteres. Al ser orientado a objetos, permite que los datos y los métodos sean encapsulados y reutilizables. Es un lenguaje ampliamente utilizado en la industria, en la investigación científica y en el campo educativo alrededor del mundo.

Otra de las ventajas de utilizar Python como lenguaje de programación, es la gran cantidad y diversidad de librerías que posee y que son desarrolladas para éste. Entre ellas, la más importante y popular es la librería de procesamiento de lenguaje natural NLTK (Natural Language Toolkit).

NLTK (Natural Language Toolkit)

NLTK provee clases básicas para la representación de datos relevantes al procesamiento del lenguaje natural, interfaces estándar para realizar tareas tales como etiquetado *part-of-speech*, análisis sintáctico, y clasificación de textos; e implementaciones estándar para cada tarea que pueden ser combinadas para resolver problemas complejos. Además posee una documentación extensa y una comunidad muy activa.

NLTK fue diseñado con cuatro objetivos principales:

Simplicidad: Para proveer un *framework*[4] intuitivo con funcionalidades considerables, brindándole al usuario conocimiento práctico, sin que éste se vea atascado en tareas tediosas asociadas con el procesamiento de lenguaje natural.

Consistencia: Para proveer una herramienta uniforme con interfaces y estructuras de datos consistentes, junto con nombres de métodos fácilmente deducibles.

Extensibilidad: Para proveer una estructura en la que nuevos módulos de software pueden ser fácilmente adicionados, incluyendo implementaciones alternativas y diversos enfoques para la misma tarea.

Modularidad: Para proveer componentes que puedan ser utilizados independientemente sin necesidad de comprender el resto de la herramienta.

Flask

Por otra parte, para la interacción con la capa de presentación, utilizamos un micro-framework web llamado Flask que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código. El término “micro” no significa que exista una carencia de funcionalidades, sino que la finalidad del mismo es mantener su núcleo simple pero extensible. Ésta característica acompaña muy bien nuestras necesidades, ya que nuestro foco está situado en el procesamiento y análisis de la información, y no en el desarrollo de una aplicación web de gran complejidad.

Por defecto Flask no incluye una capa de abstracción de base de datos, validación de formularios, ni nada para lo que existan diferentes librerías ya implementadas. En su lugar, soporta extensiones para agregar dichas funcionalidades a la aplicación como si hubiera sido implementado por uno mismo. Actualmente ya existen numerosas extensiones que proveen integración con bases de datos, mapeo objeto-relacional, validación de formularios, administración de subida de archivos, tecnologías de autenticación y más. Algunos ejemplos de aplicaciones conocidas que son desarrolladas con Flask incluyen a Pinterest, LinkedIn y la propia página web de la propia comunidad de Flask.

[4] Es un conjunto de funciones y clases estructuradas por un solo sistema, que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

Para entender lo simple que resulta Flask, aquí tenemos un ejemplo del código necesario para una aplicación que imprime *"¡Hola Mundo!"* cuando se ingresa a la URL raíz:

```
from flask import Flask
app = Flask(__name__)
@app.route("/")
def hello():
    return "¡Hola Mundo!"
if __name__ == "__main__":
    app.run()
```

5.3.3 Presentación

Para la capa de presentación, utilizamos lo que se denominan *"plantillas"*. Cada página de la aplicación tendrá el mismo diseño, y solo cambiará el contenido de la misma. Es por ello que en lugar de repetir segmentos de código HTML, cada plantilla extenderá una plantilla base, y sobrescribirá secciones específicas de la misma.

```
<!doctype html>
<title>{% block title %}{% endblock %} - Flaskr</title>
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
<nav>
  <h1>Flaskr</h1>
  <ul>
    {% if g.user %}
      <li><span>{{ g.user['username'] }}</span>
      <li><a href="{{ url_for('auth.logout') }}">Log Out</a>
    {% else %}
      <li><a href="{{ url_for('auth.register') }}">Register</a>
      <li><a href="{{ url_for('auth.login') }}">Log In</a>
    {% endif %}
  </ul>
</nav>
<section class="content">
  <header>
    {% block header %}{% endblock %}
  </header>
  {% for message in get_flashed_messages() %}
    <div class="flash">{{ message }}</div>
  {% endfor %}
  {% block content %}{% endblock %}
</section>
```

En este ejemplo podemos ver algunas de las ventajas más importantes que tienen las plantillas en comparación con el código HTML puro.

Lo primero que podemos observar son los tres bloques definidos que serán sobrescritos en las otras plantillas:

1. `{% block title %}{% endblock %}` será reemplazado por el título que corresponda en la pestaña del navegador, según la página que se esté mostrando.

2. `{% block header %}{% endblock %}` es similar a ***title***, pero aquí se muestra el título en el cuerpo de la página.

3. `{% block content %}{% endblock %}` es donde se coloca el contenido específico de cada página.

Lo siguiente que destaca es la sintaxis **`url_for()`**. Ésta es una función que está disponible automáticamente en toda plantilla, y es utilizada para generar URLs de manera dinámica, en lugar de escribirlas manualmente. Es posible generar URLs tanto para archivos (tales como estilos, scripts, etcétera), como así también rutas a las diferentes secciones de la aplicación.

También existen maneras de modificar dinámicamente el código HTML de la página. Esto se logra con instrucciones de control de flujo, tales como:

```
{% if <condición> %}{% else %}{% endif %}
o {% for <elemento> in <colección> %}{% endfor %}
```

Además de las plantillas de Flask, utilizamos Bootstrap3 para el desarrollo de las vistas de nuestra aplicación. Bootstrap es un framework web que combina HTML5, CSS3 y JavaScript para la creación de sitios adaptables.

El diseño web adaptable es una filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando. Hoy en día los sitios son visitados desde diversos dispositivos como tabletas, teléfonos, portátiles, computadoras de escritorio, etcétera. Además, dentro de cada categoría, cada aparato posee características concretas: tamaño de pantalla, resolución, potencia de cómputo, sistema operativo, memoria, entre otras. Ésta tecnología pretende que con un único diseño web, toda aplicación se vea correctamente en cualquier dispositivo.

Para la confección de los gráficos utilizamos Highcharts. Ésta es una librería de gráficos informativos multiplataforma basada en gráficos de vectores escalares implementada en JavaScript. Su principal funcionalidad es ofrecer una manera simple y completa para agregar gráficos interactivos a un sitio web, soportando diversos tipos, como gráficos de barra, de torta, líneas, etcétera.

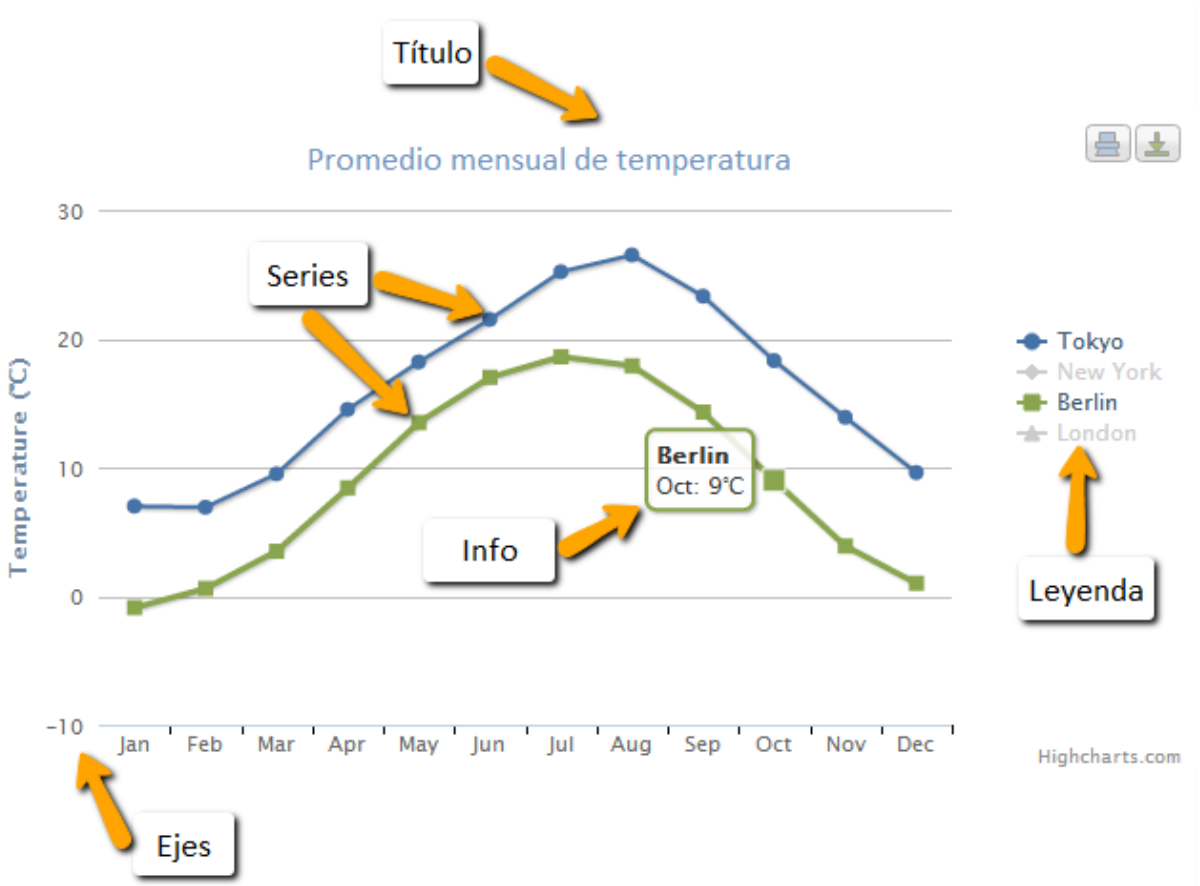


Figura 10. Figura que indica los diferentes componentes de los gráficos. [53]

Algunas características que nos brinda esta librería son:

Compatible: funciona en todos los navegadores modernos, incluyendo iPhone/iPad e Internet Explorer desde la versión 6.

Gratis para uso no comercial: No es necesario pagar para poder hacer uso de ella en un sitio web personal o sin fines de lucro.

Código abierto: Bajo cualquiera de las licencias, está permitido descargar el código fuente y realizar las modificaciones deseadas. Esto habilita las modificaciones personalizadas y una gran flexibilidad.

JavaScript puro: Highcharts está basado únicamente en tecnologías nativas de los navegadores, por lo que no requiere complementos del lado del

cliente (como Flash o Java). Incluso no es necesario instalar nada en el servidor. Highcharts requiere únicamente dos archivos JavaScript para correr: el núcleo highcharts.js y el framework de jQuery (que por lo general también es utilizado en la aplicación).

Numerosos tipos de gráficos: Highcharts soporta gráficos de línea, aérea, columnas, barras, torta y dispersión entre otros.

Dinámico: A través de la API, es posible agregar, eliminar y modificar series y puntos, o modificar ejes en cualquier momento luego de la creación del gráfico. Diversos eventos soportan modificaciones mediante programación. Combinándolos con jQuery y Ajax, es posible generar gráficos en tiempo real, que se actualizan constantemente con valores desde el servidor, mediante datos ingresados por el usuario, y más.

Ejes múltiples: A veces se quiere comparar variables que no poseen la misma escala (por ejemplo temperatura, volumen de lluvia y presión del aire). Highcharts permite asignar un eje Y para cada serie.

Etiquetas de información: Manteniendo el cursor sobre un gráfico, se muestra texto informativo para cada punto o serie. Esto permite obtener información de manera simple y clara, sin sobrecargar la pantalla.

Imprimir y exportar: Habilitando el módulo de exportaciones, los usuarios pueden exportar el gráfico en formatos PNG, JPG, PDF o SVG con tan solo un click, o directamente imprimir desde la web.

Zoom: Ampliando un gráfico, es posible examinar una parte especial más de cerca. La ampliación puede ser en el eje Y, eje X o ambos.

Carga de datos externos: Highcharts toma los datos en un arreglo de JavaScript, que puede ser definido en el objeto local, en un archivo separado o incluso proveer desde un sitio distinto.

Invertir gráfico o eje: En algunas ocasiones queremos invertir el gráfico para que el eje X aparezca en forma vertical, o invertir un eje, de manera tal que los valores mayores comiencen a partir del punto de origen.

Rotación de texto: Todos los textos, incluidos etiquetas de ejes y etiquetas de datos para los puntos y los títulos de los ejes, pueden ser rotados en cualquier ángulo.

5.4 Descripción y funcionamiento

El software desarrollado está compuesto por un conjunto de funcionalidades que nos permiten cumplir con los objetivos planteados.

Recolección y almacenamiento de comentarios

Comenzando por la recolección de la información, implementamos un proceso que se comunica con la API de Facebook y recolecta de la página ingresada como parámetro tanto publicaciones como sus correspondientes comentarios y reacciones. No es posible seleccionar individualmente estos datos, pero si se configura un corte de control. Éste puede ser configurado para alcanzar una cierta cantidad de publicaciones o de comentarios. Como ya vimos, la API responde a las peticiones con datos en formato JSON. Nuestro proceso recorre los datos, extrae la información deseada y luego la almacena en una base de datos local. De esta manera, todo el contenido obtenido queda disponible para el procesamiento posterior.

Figura 11. Captura de cómo se pueden recolectar la información.

Etiquetado de los comentarios

Con los comentarios almacenados en la base de datos, la aplicación posee una funcionalidad que nos permite etiquetarlos de forma manual, asignando una polaridad **positiva**, **neutral** o **negativa** según el criterio del usuario de la aplicación. Además de etiquetar, es posible eliminar comentarios carentes de contenido, como las etiquetas a otras personas, nombres propios, etcétera. A su vez, permite visualizar la proporción de comentarios etiquetados por clase.

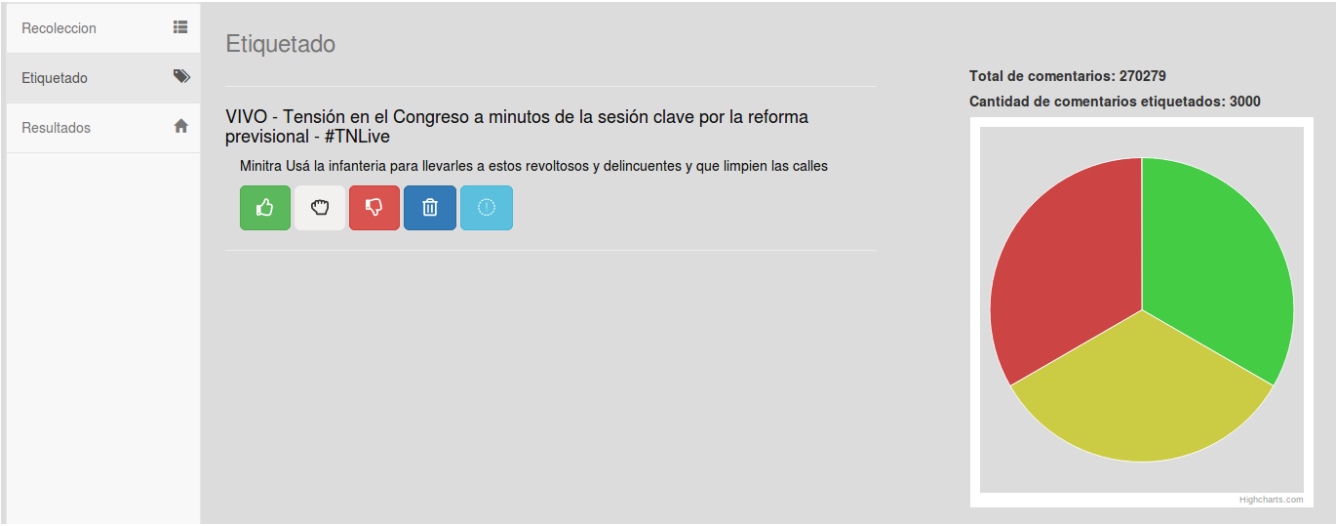


Figura 12. Captura de cómo se etiquetan los comentarios.

Análisis del corpus

La aplicación cuenta con un módulo que realiza el preprocesamiento de comentarios almacenados y está compuesto por varias etapas.

Comenzando por algo tan simple como transformar todas las palabras a minúsculas (a fin de facilitar la comparación y búsqueda de las mismas), y realizar una decodificación del formato **latin-1**, para poder interpretar caracteres especiales. Esta decodificación fue necesaria debido a que, por defecto, se utiliza el formato **utf-8**, el cual reconoce todos los caracteres utilizados en el idioma inglés, pero deja fuera muchos otros utilizados en el español (como la tilde o la letra “ñ”).

A continuación se implementó un algoritmo utilizado para particionar el texto en palabras, tomando como separadores el espacio en blanco y cualquier signo de puntuación (punto, coma, doble punto, punto y coma, guion, signo de admiración, signo de interrogación y paréntesis).

A esta altura del proceso, ya no se posee un texto, sino más bien una colección de tokens para cada comentario, por lo que comienza la etapa de procesamiento individual de cada uno de ellos.

Como primer medida, identificamos todas las ocurrencias de los distintos patrones de risa (*“jaja”*, *“jajaja”*, *“jeje”*, *“jejeje”*, etcétera), y los reemplazamos por un identificador único para facilitar su reconocimiento e interpretación en las

etapas siguientes, en todos los casos, las diferentes combinaciones fueron reemplazadas por el identificador *"inter-risa"*.

Luego, analizamos cada token y verificamos que el mismo no pertenezca al conjunto previamente categorizado como **stopword** (son palabras de uso común que no aportan contenido de opinión. Por ejemplo: *"a"*, *"la"*, *"de"*, *"del"*, *"el"*, *"que"*, *"y"*, *"asique"*, etcétera). La búsqueda se realiza tanto por el singular como por el plural, utilizando la funcionalidad ***singularize(***<*término*>***)*** provista por la librería NLTK. Todas las palabras consideradas como *stopwords* son removidas del conjunto de términos finales para cada comentario, a fin de simplificar su tamaño y tratar de mejorar el rendimiento de las técnicas que serán aplicadas.

En este punto del proceso, para cada comentario se posee una colección de tokens (producto del proceso de limpieza al que fueron sometidos los textos originales, quitando aquellos conocidos por no aportar valor). El tamaño y los elementos de la misma difieren en cada caso, y dependen del texto original. Este conjunto de tokens será el punto de partida para el análisis de cada una de las técnicas estudiadas en esta tesis.

Entrenamiento y aplicación de modelos de clasificación

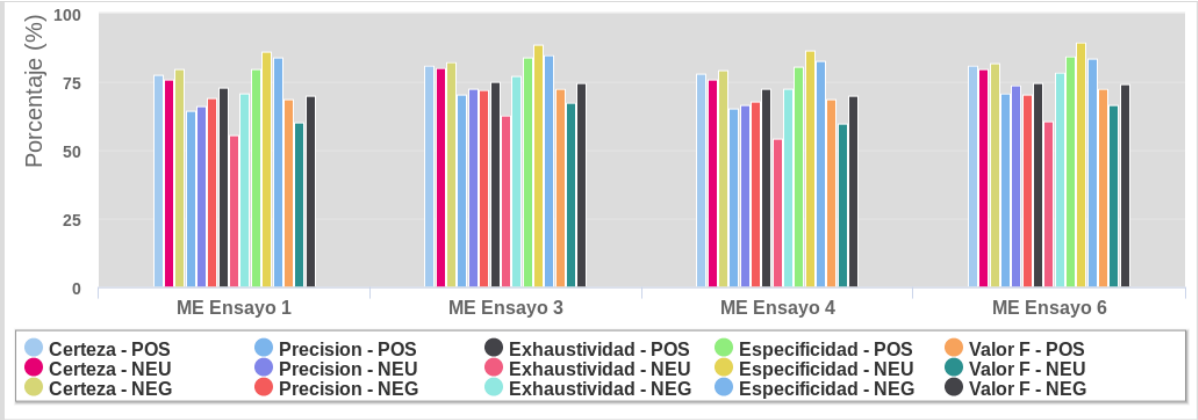
Una vez completo el preprocesamiento y limpieza del corpus, el paso siguiente consiste en aplicar las técnicas elegidas para la construcción de modelos predictores que a futuro serán utilizados para realizar la sugerencia de publicaciones mediante el análisis de sentimiento. Los detalles de la implementación para este paso se describen en la sección 5.5.

Evaluación y comparación del rendimiento de los modelos

Por último, la aplicación permite la visualización de gráficos de barras, uno por cada técnica utilizada, en los que se puede seleccionar la/s métrica/s a evaluar. El eje *y* corresponde al valor en porcentaje de las mismas, mientras que el eje *x* se muestran los diferentes ensayos realizados.

A continuación se muestra un ejemplo de cómo son presentados los datos en la pantalla del sistema, donde se puede visualizar, por clase, las métricas de todos los ensayos realizados y debajo del gráfico, los valores de las correspondientes matrices de confusión. Cabe aclarar que es posible filtrar las

métricas mediante las opciones en la parte inferior, y elegir puntualmente la métrica y la clase que se desea mostrar.



Ensayo	Positivos	Neutrales	Negativos	Certeza	Precisión	Exhaustividad	Especificidad	Promedio Armónico
ME Ensayo 1	• POS:**644** • NEU:126 • NEG:112	• 216 • **480** • 169	• 137 • 121 • **624**	• 77.52% • 75.96% • 79.50%	• 64.59% • 66.02% • 68.95%	• 73.02% • 55.49% • 70.75%	• 79.79% • 86.00% • 83.92%	• 68.55% • 60.30% • 69.84%
ME Ensayo 3	• POS:**660** • NEU:116 • NEG:104	• 164 • **542** • 159	• 114 • 88 • **678**	• 81.03% • 79.92% • 82.29%	• 70.36% • 72.65% • 72.05%	• 75.00% • 62.66% • 77.05%	• 84.07% • 88.41% • 84.93%	• 72.61% • 67.29% • 74.46%
ME Ensayo 4	• POS:**640** • NEU:125 • NEG:117	• 209 • **470** • 186	• 131 • 112 • **639**	• 77.86% • 75.96% • 79.23%	• 65.31% • 66.48% • 67.83%	• 72.56% • 54.34% • 72.45%	• 80.54% • 86.56% • 82.66%	• 68.74% • 59.80% • 70.07%
ME Ensayo 6	• POS:**657** • NEU:108 • NEG:115	• 163 • **525** • 177	• 109 • 81 • **690**	• 81.14% • 79.85% • 81.64%	• 70.72% • 73.53% • 70.26%	• 74.66% • 60.69% • 78.41%	• 84.41% • 89.26% • 83.27%	• 72.64% • 66.50% • 74.11%

Figura 13. Captura de cómo son presentados los datos para los ensayos 1, 3, 4 y 6 de ME.

En resumen, el usuario de la aplicación puede elegir una página de Facebook, descargar sus publicaciones y comentarios, etiquetar los mismos y entrenar los modelos. A partir de éstos, podría extenderse la aplicación para realizar una valoración del sentimiento de las publicaciones, y brindar así un sistema de recomendación de las mismas. Entonces, si el usuario está conforme con los resultados obtenidos, en el futuro la aplicación podría realizar una descarga personalizada de publicaciones y sus comentarios, a fin de mostrar sólo aquellas que le resulten de interés.

5.5 Estudio de técnicas para el Análisis de Sentimientos

La primera técnica estudiada mencionada en [3], es la denominada “basada en diccionario”. La misma consiste básicamente en buscar cada token en un diccionario previamente definido (el cual posee pares del tipo <*palabra, valor*>), para luego analizar cada uno de los valores obtenidos y a partir de ellos, poder calcular la polaridad final del texto.

Para poder ejecutar el algoritmo, es necesario realizar la llamada al script correspondiente desde la terminal, como se muestra a continuación:

```
python lexicon.py
```

En una primera instancia utilizamos un diccionario, generado en [42], el cual cuenta con un total de 12000 palabras aproximadamente, cada una acompañada de un valor que representa el peso de su polaridad, acotado en el rango [-1,1]. Debido a la gran diversidad de dialectos que posee el idioma español, muchos de los términos contenidos en dicho diccionario no son utilizados en Argentina. Como contrapartida, luego de contrastar dicha lista con el corpus recopilado, notamos que existían varias palabras (500 aproximadamente) con una frecuencia considerablemente alta, y las mismas no estaban contenidas en dicho documento, por lo que tomamos la decisión de incluirlas de forma manual: con los comentarios ya etiquetados, a cada palabra a agregar se le asignó un valor que es igual al promedio de las polaridades de los comentarios en los que aparece. Es decir, se suman las polaridades de todos los comentarios en los que ese token aparece y luego se divide por la cantidad de ocurrencias.

En éste diccionario resultante se encuentran la mayoría de las palabras consideradas como portadoras de sentimiento u opinión, y las mismas están asociadas a un valor que refleja el grado de positividad o negatividad que aporta. Luego de buscar cada uno de los tokens y asignarles un valor, es momento de calcular el valor global de cada comentario para obtener su polaridad final.

Diversas son las alternativas a la hora de realizar este cálculo, ya que el valor de las palabras de todo un comentario podría verse invertido por el uso de la negación *“no”*, o del sarcasmo, por citar algunos casos. Un ejemplo de

implementación es el propuesto en [50], donde se le da mayor importancia a las palabras mientras más cercanas al final de la frase se encuentren. Pero a fin de no extender el alcance de nuestro trabajo por fuera de los límites propuestos, optamos por un cálculo más directo, en el que al comentario se le asigna el promedio obtenido a partir de los valores individuales de cada palabra. Este valor está normalizado dentro del intervalo [-1, 1], siendo -1 el extremo negativo, y 1 el positivo.

La aplicación cuenta con un módulo que permite ejecutar un algoritmo de machine learning para conseguir un modelo clasificador de comentarios, el cual puede ser utilizado a posteriori para filtrar publicaciones o comentarios.

En esta tesina se estudiaron tres algoritmos de aprendizaje supervisado: Naïve Bayes, ME y SVM. Para realizar el entrenamiento y testeo se implementó un script que ejecuta los tres algoritmos con la misma base de datos para obtener resultados y poder analizarlos luego.. Este script se ejecuta con el comando que se muestra a continuación:

```
python tecnicas.py
```

El proceso de entrenamiento y testeo se realiza utilizando la técnica de validación cruzada de K iteraciones (descrita en la sección 4.3.1). El tamaño de cada conjunto es un parámetro que se puede controlar y así obtener distintos resultados. Usualmente se divide en 5 partes, por lo que la proporción resultante es: 80% de entrenamiento y 20% de testeo.

Previo al entrenamiento, es necesario normalizar el formato en el que los comentarios serán usados para entrenar los modelos. Esta normalización se hizo con la técnica “bolsa de palabras” (*Bag of Words - BoW*) ya descrita en la sección 3.4. En este punto, se le permite al usuario seleccionar cuál será la cantidad mínima de ocurrencias que debe tener cada palabra para que forme parte de esta estructura.

A esta altura del proceso, cada comentario cuenta con un BoW en el que se especifica la presencia o ausencia de ciertos tokens.

Los vectores correspondientes a cada BoW de cada comentario son utilizados tanto para entrenar los modelos como para testear a los mismos. En esta etapa, las técnicas de machine learning estudiadas analizan los patrones que conforman los

BoW de cada comentario, creando una relación entre un comentario y su clase asignada.

Una vez finalizado el proceso de aprendizaje, se aplica el modelo a cada comentario del conjunto de testeo, a fin de comparar si la polaridad otorgada por la técnica coincide con aquella asignada previamente de forma manual. Con estos datos, se generan tantas matrices de confusión como iteraciones posee la validación cruzada. Finalmente se unifican los valores de las mismas, obteniendo una matriz global para todas las iteraciones, a partir de la cual se realiza el estudio de rendimiento.

Capítulo 6

Estudio realizado

6.1 Caso de estudio

Para llevar a cabo nuestra investigación y el estudio de las técnicas de machine learning mencionadas en el capítulo anterior, se recolectaron publicaciones junto con sus comentarios del portal de noticias *Todo Noticias* de Facebook, el cual aporta una gran variedad de noticias diariamente, desde acontecimientos políticos a climáticos, como también de espectáculos, etcétera. Además, como se explicó previamente, en el capítulo 2, es una página pública la cual nos permite acceder a sus publicaciones, en este caso noticias, y a los comentarios de las mismas sin tener que pedir permiso de acceso a los usuarios ni violar su privacidad.

Se obtuvieron 270279 comentarios pertenecientes a 569 publicaciones cuyas fechas varían desde el 16 de Junio del 2017 al 15 de Febrero de 2018. Algunos de los temas más debatidos fueron política, economía y espectáculos de nuestro país, y algunos de noticias internacionales, como Chile o México. También muchas noticias hacían referencia al fallecimiento de la periodista Débora Pérez Volpin.

Por lo detallado en la sección 1.2, se armó una base de datos de 3000 comentarios aleatoriamente, obteniendo un corpus balanceado, de 1000 comentarios por cada clase (positivo, neutral y negativo), confeccionando una base de datos cuyo tamaño resulta razonable para poder armar conjuntos de entrenamiento y testeo de diversos tamaños, permitiendo realizar diversos ensayos con las técnicas de aprendizaje seleccionadas para su estudio en esta tesina (NaÏve Bayes, Máxima Entropía y Máquinas de Vectores de Soporte).

Se plantearon diversos escenarios de prueba, en los que se modificaron distintas variables (el número de iteraciones para la validación cruzada, la cantidad mínima de tokens en comentarios y tamaño del BoW), descritos en la siguiente sección. Luego, se aplicó cada una de las técnicas antes descritas, a fin de comparar las diferentes salidas que produjeron; es decir, las matrices de confusión

en cada uno de los escenarios de entrenamiento. Con estos resultados, se calcularon las distintas métricas, explicadas en el capítulo 4, en la sección Performance, para cada una de las matrices obtenidas.

6.2 Etapa de entrenamiento

A fin de realizar un análisis exhaustivo y observar el comportamiento de las técnicas frente a distintos escenarios, se plantearon una serie de pruebas, variando diversos parámetros en las etapas de preprocesamiento de los datos y la aplicación de los modelos.

Comenzando por la técnica basada en diccionario, introducida en la sección 5.5, partimos de un valor comprendido por el intervalo [-1, 1]. Para interpretar cada una de las tres clases de salida que deseamos (negativo, neutral y positivo), se propusieron dos valores donde se segmenta dicho intervalo. El primer límite divide las clases negativo y neutral, por lo que cualquier comentario con valor perteneciente al intervalo [-1, límite_1) será clasificado como negativo. Todo aquel perteneciente al intervalo [límite_1, límite_2] será neutral, y por descarte, (límite_2, 1] contiene los valores considerados como positivos.

Para la prueba inicial, estos valores fueron configurados en -0,3 y 0,3 (siendo los valores que dividen aproximadamente el intervalo en tres partes iguales). Luego de una serie de ensayos intermedios, se modificaron los límites de los intervalos para lograr una distribución más equitativa de las predicciones en cada clase, logrando así mejores resultados a nivel general en todas las métricas. En la siguiente sección se analizan los resultados obtenidos con estos valores.

Como se explicó en la sección 5.4, se generó una colección de tokens para cada comentario. A raíz de esto, se eligió contemplar dos escenarios: tomar la totalidad de los comentarios etiquetados, y sólo aquellos que poseen un mínimo de 2 tokens, con el objetivo de evaluar el desempeño de las técnicas contemplando texto más extenso. Decidimos realizar el filtro con un mínimo de 2 tokens, ya que la cantidad de comentarios con más de 3 o 4 tokens decrece considerablemente y el conjunto de testeo final sería demasiado pequeño. Los resultados de las métricas se verían afectados significativamente por no contar con una cantidad de casos de

prueba lo suficientemente representativa. En la tabla 12, se observan la cantidad de comentarios etiquetados por clase, según el filtro de tokens que se elija.

Tokens	Positivos	Neutrales	Negativos
>=1	1000	1000	1000
>=2	884	865	889
>=3	721	661	768
>=4	604	420	652

Tabla 12. Cantidad de comentarios etiquetados según el filtro de número de tokens.

Con respecto al BoW, introducido en la sección 3.4, se seleccionaron como palabras aquellas cuyo número mínimo de ocurrencias es 13 o 19 dentro de todo el conjunto de comentarios etiquetados. El primero de estos valores (13) fue elegido en base al mayor número de ocurrencias (123 para la palabra “bien”), a partir del cual, se tomó el 10% aproximadamente, y de ahí el número 13, consiguiendo un BoW formado por 108 tokens. Luego, el valor 19 fue producto de la decisión por parte de los tesistas de disminuir el tamaño del BoW, a fin de comprobar las diferencias obtenidas en los resultados. Para ello se decidió disminuir en un 30% aproximadamente el tamaño del BoW, reduciendo de 108 a 73 tokens, y para ello, fue necesario establecer el valor de corte en 19 ocurrencias de una palabra en todo el conjunto de comentarios. Se hicieron pruebas contemplando mayor cantidad de palabras (decrementando el mínimo de ocurrencias requerido), sin conseguir una mejora sustancial de la performance.

Por último, en gran parte de la bibliografía consultada, se particiona el total de los datos de manera tal que un 80% de los mismos se destina para entrenamiento y el porcentaje restante para testeo. Al utilizar la técnica de cross validation (presentada en la sección 4.3), es necesario dividir los datos en 5 particiones (k = 5) para poder obtener los porcentajes antes mencionados. Tomando estos valores como punto de partida, optamos por analizar la performance de las técnicas cuando esta proporción varía. Para ello, se eligieron dos valores para el parámetro k. Por un lado, con k = 3, se obtiene una proporción de (~66.66%, ~33.33%). Por el otro, con k = 10, una proporción de (90%, 10%), teniendo en cuenta que siempre el mayor valor se corresponde con el conjunto destinado al entrenamiento.

6.3 Evaluación de técnicas

Partiendo de las descripciones anteriores, se plantean cada uno de los ensayos, especificando los valores y las formas en las que se les suministra el conjunto de entrenamiento. Además, para cada ensayo se muestra la matriz de confusión obtenida en cada técnica, y distintos gráficos correspondientes a las diversas métricas de rendimiento.

Lexicon

Para el análisis de la técnica basada en diccionario, se modificaron 2 parámetros: los valores utilizados para discernir la clase a la que pertenece cada caso, y el hecho de filtrar o conservar las *stopwords*. Se realizaron 4 ensayos, con las siguientes configuraciones:

- **Ensayo 1:** Sin filtrar stopwords. Límites: -0.3, 0.3.
- **Ensayo 2:** Sin filtrar stopwords. Límites. -0.009, 0.05.
- **Ensayo 3:** Filtrando stopwords. Límites: -0.3, 0.3.
- **Ensayo 4:** Filtrando stopwords. Límites: -0.009, 0.05.

Las matrices de confusión con los resultados obtenidos para cada ensayo, se exponen a continuación:

		Predicción		
		Pos	**Neu**	**Neg**
Real	**Pos**	**143**	855	2
	Neu	2	**996**	2
	Neg	1	936	**63**

Tabla 13: Lexicon - Ensayo 1.

		Predicción		
		Pos	**Neu**	**Neg**
Real	**Pos**	**628**	268	104
	Neu	122	**628**	250
	Neg	45	242	**713**

Tabla 14: Lexicon - Ensayo 2.

		Predicción		
		Pos	**Neu**	**Neg**
Real	**Pos**	**281**	714	5
	Neu	6	**980**	14
	Neg	6	856	**138**

Tabla 15: Lexicon - Ensayo 3.

		Predicción		
		Pos	**Neu**	**Neg**
Real	**Pos**	**721**	196	83
	Neu	181	**641**	178
	Neg	92	244	**664**

Tabla 16: Lexicon - Ensayo 4.

A partir de estas matrices, se obtuvieron las métricas de performance descritas en la sección 4.3. En la siguiente sección, se analizan en detalle los gráficos generados para cada una de estas métricas.

Modelos automáticos

Dado que la técnica de ME realiza varias iteraciones durante el proceso de aprendizaje, el parámetro correspondiente a la cantidad de dichas repeticiones debe ser provisto a la hora de entrenar. Como es de esperarse, a mayor cantidad de iteraciones, mayor será el tiempo requerido para finalizar el proceso, como así también, mejor será la precisión del modelo obtenido. Para seleccionar la cantidad de iteraciones a utilizar en todos los ensayos y poder encontrar un *trade-off* óptimo entre tiempo de ejecución y precisión del modelo, se realizó una serie de pruebas previas, modificando solo el parámetro de cantidad de iteraciones y comparando tanto tiempo transcurrido como la precisión obtenida en cada caso. El resto de los parámetros para todos los casos fueron los siguientes:

- Comentarios con cualquier cantidad de tokens.
- Conjuntos de entrenamiento y testeo homogéneos respecto a la frecuencia de comentarios en cada clase.
- Tamaño de BoW: 73 tokens.
- 5 folds en el cross validation.

Luego de 6 pruebas con diferente cantidad de iteraciones, se obtuvieron los siguientes datos:

Iteraciones	Minutos	Precisión	Precisión extra	Min para +1%
5	1.28	62.10%	-	-
10	2.77	62.90%	+0.80%	1.86
30	8.70	64.17%	+2.07%	4.66
60	20.35	66.98%	+4.88%	4.14
100	28.58	68.68%	+6.58%	4.84
150	48.18	69.85%	+7.75%	16.75

Tabla 17. Resultados de tiempo, precisión, mejora de precisión y tiempo requerido para la mejora de un 1% de Máxima Entropía según la cantidad de iteraciones.

Lo primero que debemos destacar de la tabla 17 es la relación lineal que posee el tiempo necesario respecto de la cantidad de iteraciones realizadas.

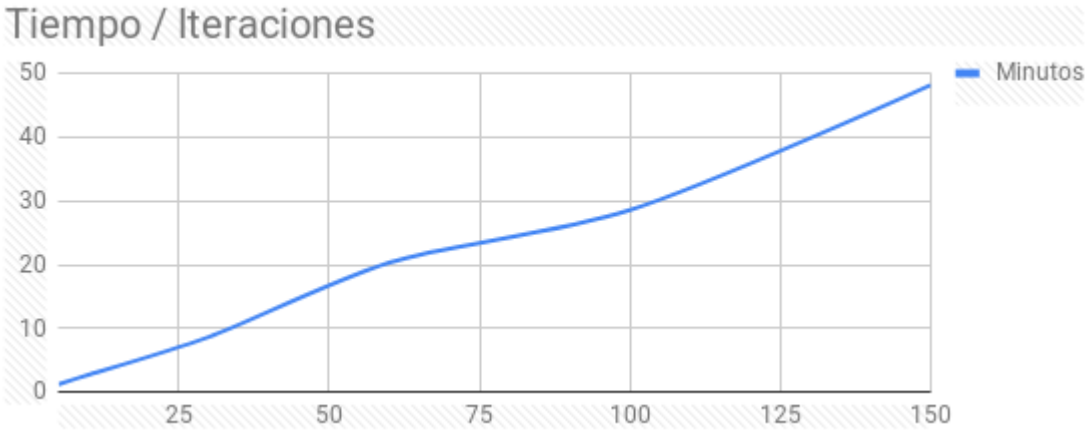


Gráfico 1. Relación tiempo/iteraciones de Máxima Entropía.

Luego, analizamos la relación entre la precisión obtenida y la cantidad de iteraciones. Aquí vemos que conforme aumenta este último valor, la mejora relativa en el rendimiento de la técnica va siendo cada vez menor.

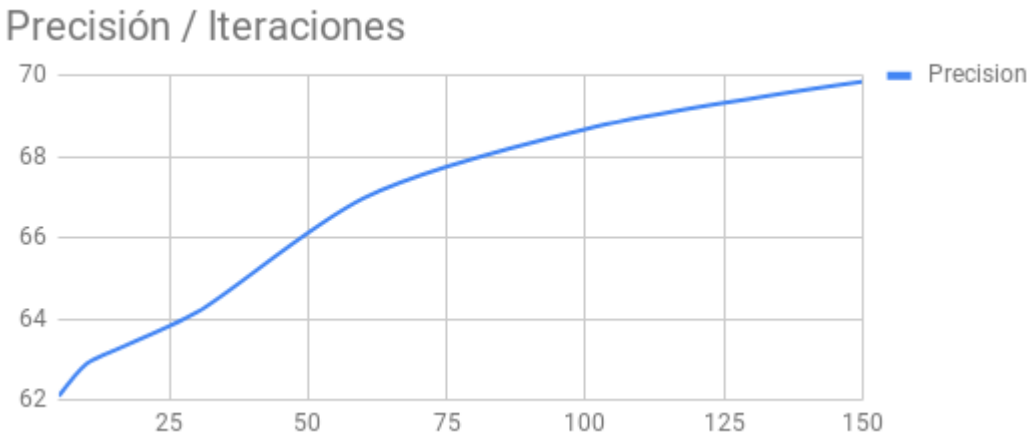


Gráfico 2. Relación precisión/iteraciones de Máxima Entropía

De estos dos análisis, podemos obtener una cantidad de iteraciones a partir de la cual, la relación entre tiempo invertido en el entrenamiento y mejora obtenida comienza un crecimiento exponencial. En el Gráfico 3 podemos ver que ese número de iteraciones es 100, por lo que éste será el valor que utilizaremos de aquí en adelante para todos los ensayos.

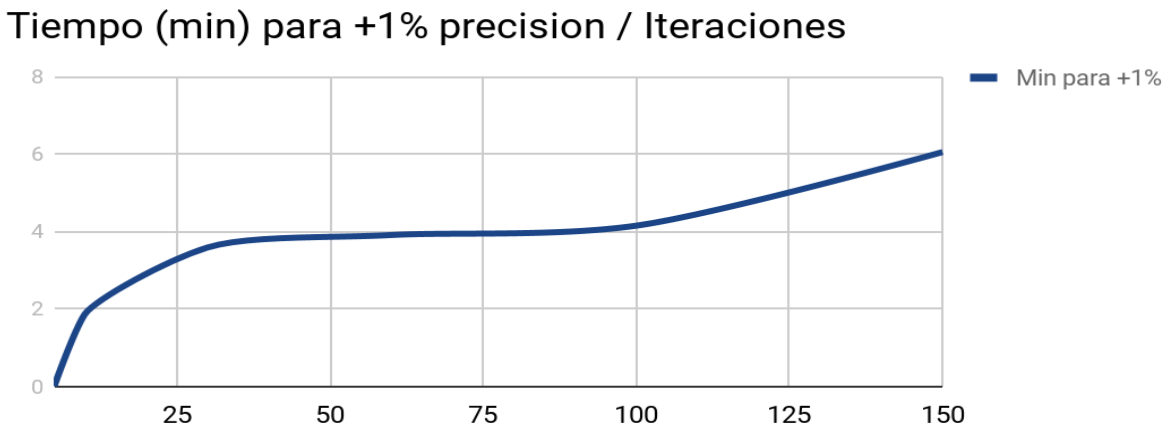


Gráfico 3. Relación entre la cantidad de minutos necesarios de entrenamiento para aumentar en 1% la precisión y la cantidad de iteraciones realizadas, de Máxima Entropía

Para identificar los ensayos que realizamos y la especificación de las variables para cada uno, presentamos la Tabla 18.

N° ensayo	N° tokens	N° Folds	Tamaño BoW
1	>=2	3(66.66%-33.33%)	108
2	>=2	5(80%-20%)	108
3	>=2	10(90%-10%)	108
4	>=2	3(66.66%-33.33%)	73
5	>=2	5(80%-20%)	73
6	>=2	10(90%-10%)	73
7	>=1	3(66.66%-33.33%)	108
8	>=1	5(80%-20%)	108
9	>=1	10(90%-10%)	108
10	>=1	3(66.66%-33.33%)	73
11	>=1	5(80%-20%)	73
12	>=1	10(90%-10%)	73

Tabla 18. Resumen de ensayos.

A continuación se listan todas las matrices de confusión obtenidas para cada ensayo con cada técnica. Cabe aclarar que en todos los ensayos realizados, no se variaron los parámetros internos para los algoritmos. Naïve Bayes particularmente no posee parámetro alguno, en SVM se estableció como kernel el tipo lineal, y en ME se configuró con 100 iteraciones y con el algoritmo 'GIS' (*Generalized Iterative Scaling - Escalado Iterativo Generalizado*).

		Naïve Bayes			Maximum Entropy			SVM		
Ensayo		Pos	Neu	Neg	Pos	Neu	Neg	Pos	Neu	Neg
1	Pos	**611**	130	141	**644**	126	112	**636**	157	89
	Neu	96	**598**	171	216	**480**	169	77	**677**	111
	Neg	68	117	**697**	137	121	**624**	97	221	**564**
3	Pos	**629**	114	137	**660**	116	104	**658**	144	78
	Neu	80	**609**	176	116	**542**	159	73	**694**	98
	Neg	63	97	**720**	116	88	**678**	89	201	**590**
4	Pos	**623**	102	157	**640**	125	117	**636**	157	89
	Neu	112	**544**	209	209	**470**	186	77	**677**	111
	Neg	71	88	**723**	131	112	**639**	97	221	**564**
6	Pos	**636**	90	154	**657**	108	115	**658**	144	78
	Neu	87	**565**	213	163	**525**	177	73	**694**	98
	Neg	57	69	**754**	109	81	**690**	89	201	**590**
7	Pos	**685**	161	153	**598**	306	95	**701**	210	88
	Neu	112	**703**	184	89	**805**	105	78	**808**	113
	Neg	70	164	**765**	97	341	**561**	110	286	**603**
9	Pos	**717**	146	137	**607**	303	90	**736**	185	79
	Neu	87	**724**	189	97	**805**	98	72	**825**	103
	Neg	66	130	**804**	79	318	**603**	96	237	**667**
10	Pos	**684**	137	178	**602**	293	104	**701**	210	88
	Neu	120	**659**	220	94	**793**	112	78	**808**	113
	Neg	68	110	**821**	100	310	**589**	110	286	**603**
12	Pos	**727**	111	162	**619**	288	93	**736**	185	79
	Neu	107	**660**	233	97	**789**	114	72	**825**	103
	Neg	60	94	**846**	76	296	**628**	96	237	**667**

Tabla 19. Matrices de confusión para cada ensayo para cada técnica. (Horizontal: real, Vertical: prediccion).

6.4 Comparación de métricas para las técnicas

En la presente sección, se realiza un análisis detallado a partir de los gráficos correspondientes a cada una de las métricas obtenidas para cada ensayo. Trataremos de hacer énfasis en las similitudes y las diferencias observadas, y los motivos o las causas a las cuales se puedan atribuir.

En general, las métricas aumentan conforme aumenta el número de folds. Puede deberse a que se utilizan mayor cantidad de datos para el entrenamiento.

6.4.1 Lexicon

Certeza

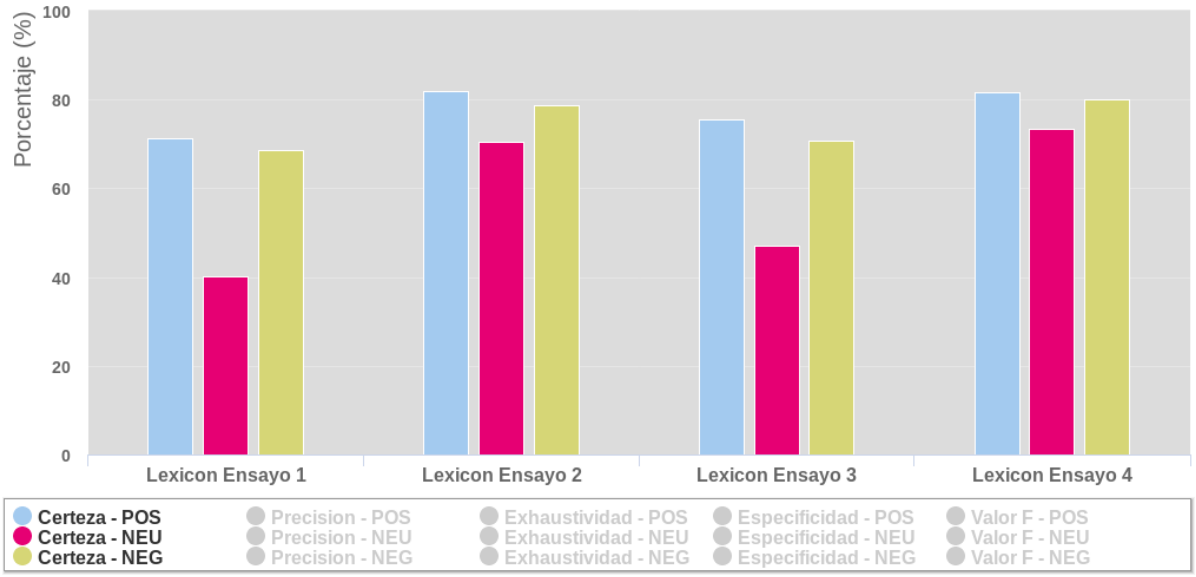


Gráfico 4. Valores de certeza de lexicon para cada ensayo.

Podemos observar que en los ensayos 1 y 3, donde los límites fueron establecidos en 3 intervalos similares, la certeza en la clase ‘neutral’ es considerablemente inferior a las otras dos (~ -30%). Esto se debe a que, como puede observarse en Tabla 13 y Tabla 15, la técnica tiene una clara tendencia a devolver como resultado la clase ‘neutral’. Es de esperarse entonces, que gran parte de estas predicciones sean erróneas. En cambio, en los ensayos 2 y 4 vemos una notable mejoría en la certeza de la clase ‘neutral’, y una diferencia entre las tres clases de no más del 10%.

En cuanto al hecho de filtrar o conservar las stopwords, no se observaron diferencias notables entre los ensayos, apenas de ~4%.

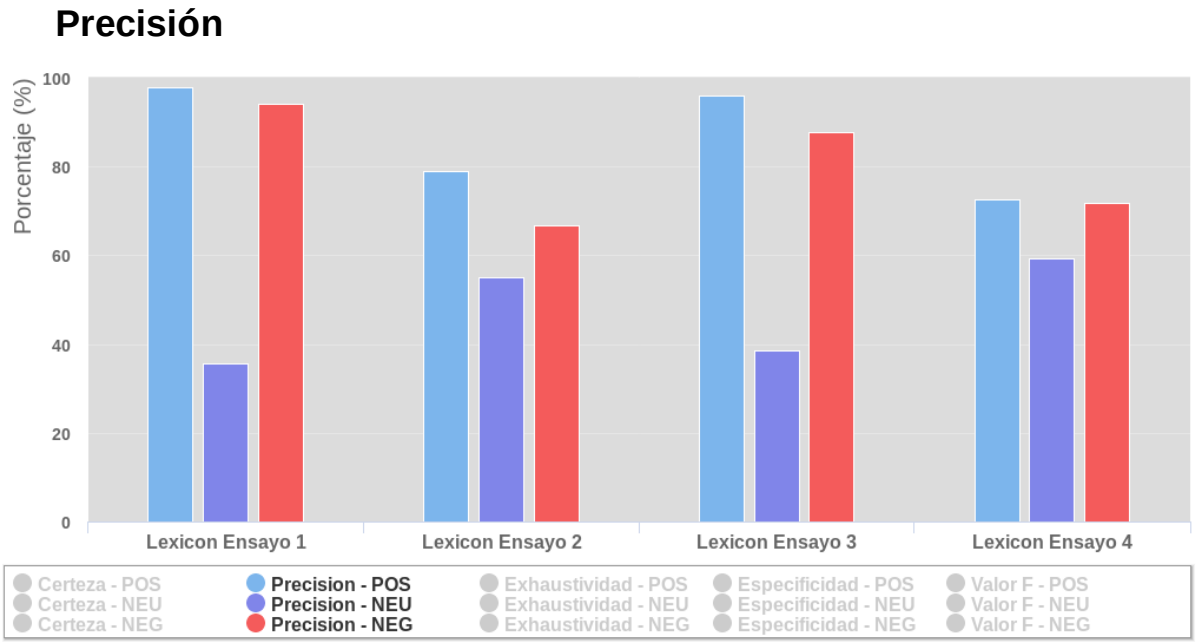


Gráfico 5. Valores de precisión de lexicon para cada ensayo.

En cuanto a la precisión, vemos una diferencia mayor entre la clase ‘neutral’ y del resto, la cual se hace más evidente en los ensayos 1 y 3. Esto se debe a que la precisión toma únicamente las predicciones realizadas para cada clase, y a partir de allí, representa qué porcentaje de las mismas fueron correctas. Podemos deducir entonces que las clases ‘positivo’ y ‘negativo’ (los extremos del intervalo que representa el rango de valores posibles) poseen una mayor precisión dado que, para que una predicción pertenezca a alguna de estas clases, debe poseer una polaridad significativa, es decir, debe estar alejada del centro 0. Así, es más probable que dicho caso sea efectivamente un comentario ‘positivo’ o ‘negativo’.

En el siguiente gráfico, se muestra la distribución de los valores asignados por esta técnica para cada comentario del corpus. Sobre el eje *x* se visualizan cada uno de los 3000 comentarios, primero los negativos, luego los neutrales y finalmente los positivos. Sobre el eje *y* se encuentra el intervalo [-1, 1] de valores posibles. Podemos observar que la mayoría de los valores se encuentran entre -0.3 y 0.3 aproximadamente, corroborando la hipótesis planteada en el gráfico de certeza, en la que se asumía que la técnica tenía una tendencia hacia la clase neutral.

Gráfico 6. Distribución de predicciones de lexicon.

Luego de analizar ambos gráficos, vemos que en los ensayos 2 y 4, (donde los límites se modificaron acercándolos al centro 0, de manera tal de contemplar mayor cantidad de negativos y positivos), las precisiones de las tres clases tienden a

equipararse. Este fenómeno se divide en dos partes: el aumento de la precisión en la clase *neutral*, y el decremento en las clases *positivo* y *negativo*. Primero, la clase *neutral* es más precisa porque, al desplazar los límites hacia el centro, para que un caso sea considerado perteneciente a esta clase, su polaridad debe ser muy próxima al 0, por lo tanto, mayor probabilidad de que realmente sea neutral. Como contraparte, sucede algo similar con las clases *positivas y negativas*, pero a la inversa. Los intervalos de valores considerados para cada una de estas clases se ampliaron, por lo que abarcan mayor cantidad de casos, antes contemplados como neutrales. Estos últimos poseen valores intermedios (ni muy cercanos a -1 o 1, ni muy cercano al 0), por lo tanto son más difíciles de clasificar, y se les atribuye una clase errónea con mayor frecuencia.

Otra diferencia que pudimos observar, en los ensayos 2 y 4, corresponde a la mejora a nivel general entre las 3 precisiones. Si bien en el ensayo 4 (filtrando las stopwords) disminuye levemente la precisión en positivos, tanto en los neutrales como en los negativos aumenta.

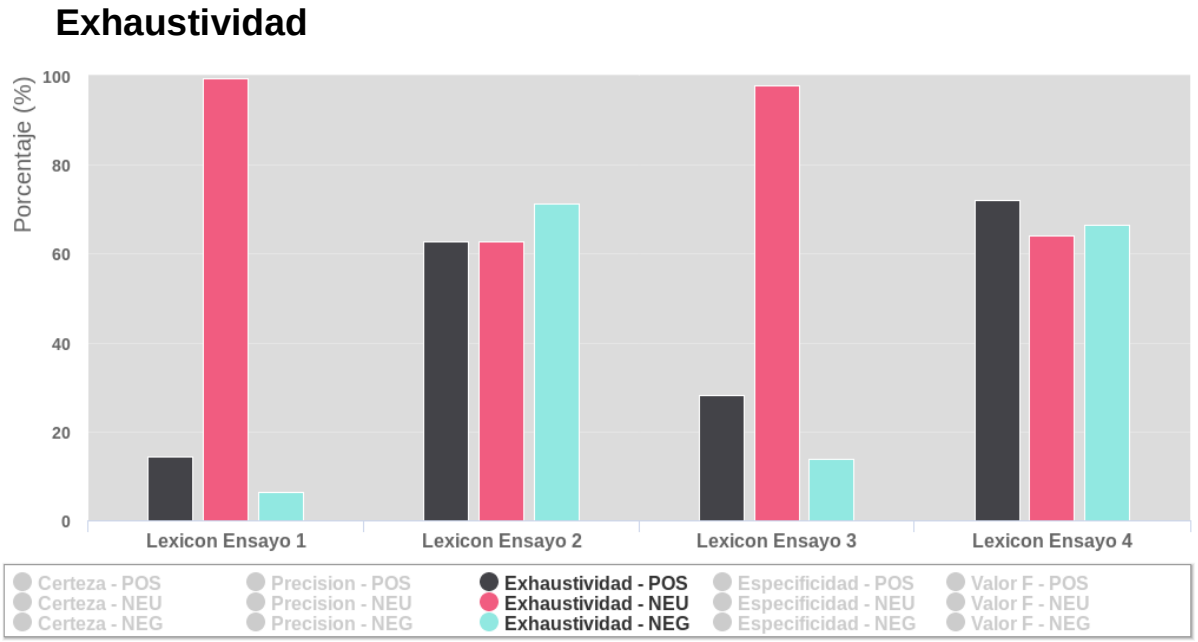


Gráfico 7. Valores de exhaustividad de lexicon para cada ensayo.

Al igual que como se observó en la certeza, aquí es más evidente la tendencia de la técnica por devolver en su mayoría predicciones para la clase *neutral* en los ensayos 1 y 3. Esto se deduce, no solo por el hecho de que la exhaustividad para la clase *neutral* es prácticamente 100% (dado que en un modelo perfecto, los tres

valores serían 100%), sino que para las clases *positivo* y *negativo* se acerca a 0%. Por el contrario, en los ensayos 2 y 4, se observa un balance entre los tres valores, por lo que la mejora es clara.

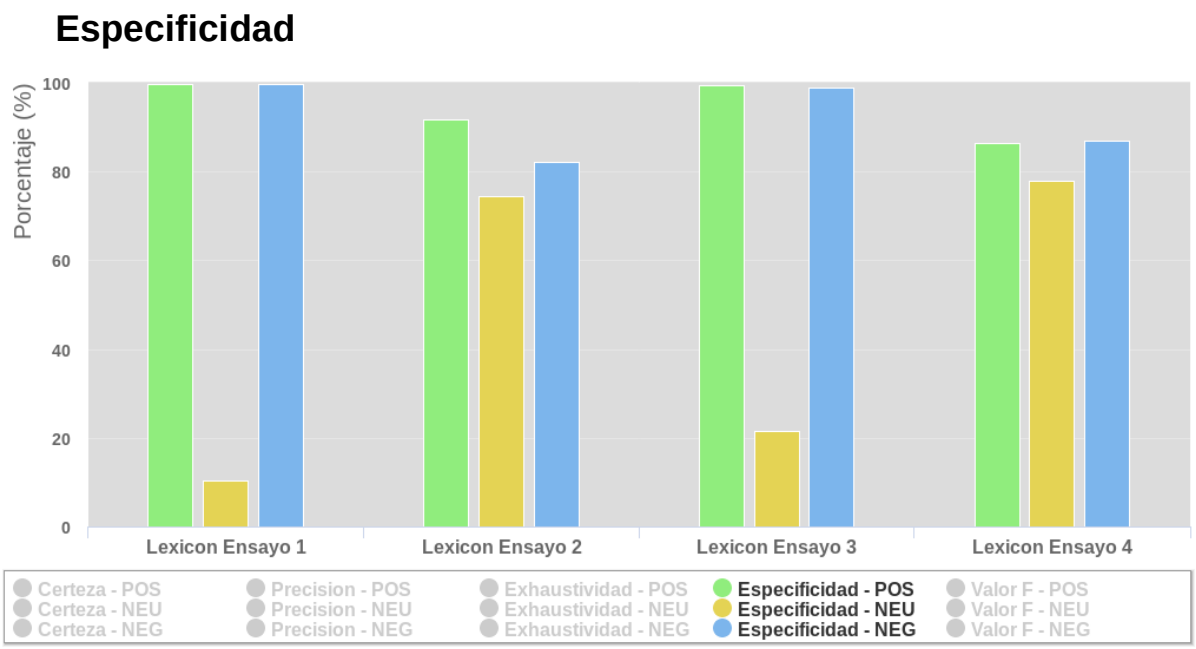


Gráfico 8. Valores de especificidad de lexicon para cada ensayo.

Aquí vuelve a ser evidente la tendencia en los ensayos 1 y 3, dado que poseer una especificidad extremadamente baja en los neutrales da indicio de que muchos de los casos pertenecientes a otras clases, están siendo atribuidos a ésta.

En los ensayos 2 y 4 vemos que la diferencia no es tan notoria, e incluso los valores son elevados (75%-90%) en las 3 clases, por lo que da indicio de que el algoritmo sabe reconocer cuando un caso no pertenece a cierta clase.

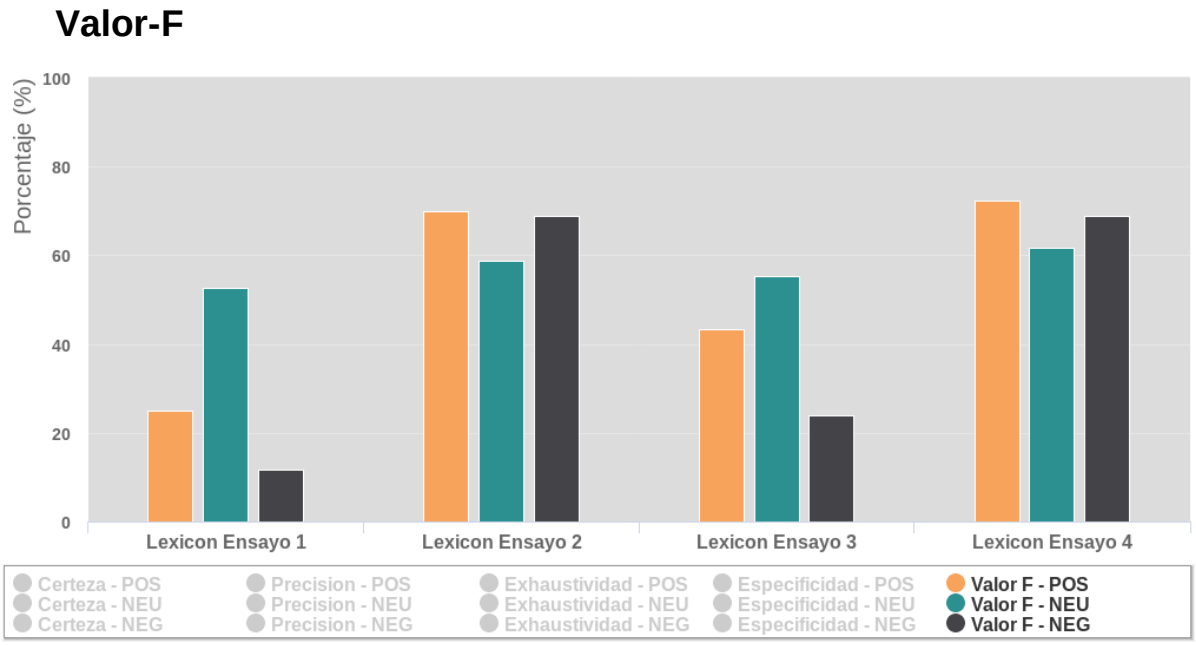


Gráfico 9. Valores de valor-F de lexicon para cada ensayo.

Analizando el gráfico correspondiente al promedio armónico, queda clara la gran ventaja de utilizar ésta métrica en comparación con la media aritmética entre la precisión y la exhaustividad. Tomando como ejemplo el ensayo 1, si promediamos los valores de la precisión y la exhaustividad para cada clase, obtendremos los siguientes resultados: Positivos: 56.12%, Neutrales: 67.66%, Negativos: 50.16%. A primera vista parecen ser valores aceptables, aun cuando tanto la precisión de los neutrales como la exhaustividad de los positivos y negativos son realmente malas. Gracias al promedio armónico, los valores obtenidos reflejan la disparidad entre los valores de las dos métricas, dando indicio de que el modelo obtenido no es aceptable.

Cabe decir que la determinación de los límites resulta crucial a la hora de obtener buenos resultados. Si bien la técnica es altamente personalizable (gracias a la opción de configuración manual de los valores en el diccionario y los límites), ésta se muestra muy sensible respecto de estos valores, y encontrar la configuración óptima resulta costoso.

6.4.2 Naïve Bayes

Luego de un análisis preliminar de los resultados obtenidos con los diferentes folds (3, 5 y 10), se observó una correspondencia lineal (en algunos casos positiva,

en otros negativa) entre la cantidad de folds y las métricas obtenidas. Por ello es que a fines prácticos, para facilitar la comprensión de los gráficos y no sobrecargarlos, se optó por mostrar únicamente aquellos con 3 y 10 folds, siendo estos los extremos donde resultan más evidentes las variaciones en los resultados.

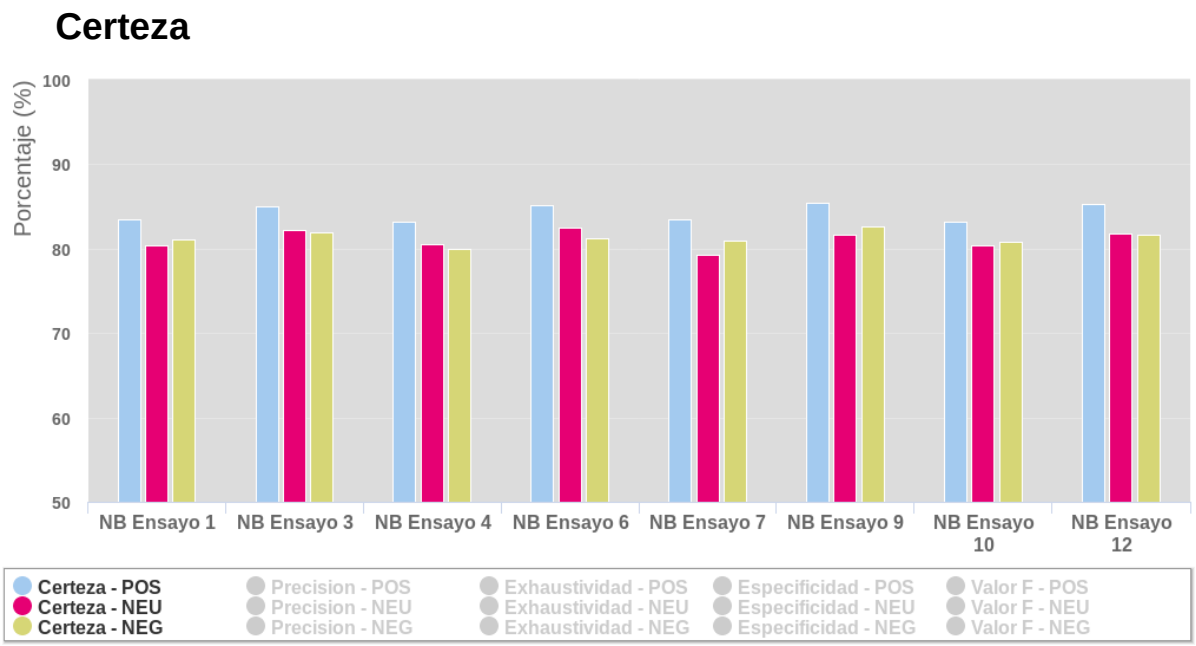


Gráfico 10. Valores de certeza de NB para cada ensayo.

A mayor cantidad de folds, mayor certeza en las tres clases (ensayo 3, 6, 9 y 12). Además se observa una leve diferencia entre los primeros cuatro ensayos del Gráfico 10 con respecto al resto, donde la certeza en los neutrales baja, y en los negativos se incrementa. En cuanto a los positivos, vemos que la certeza es siempre superior, pero no por un gran margen.

Precisión

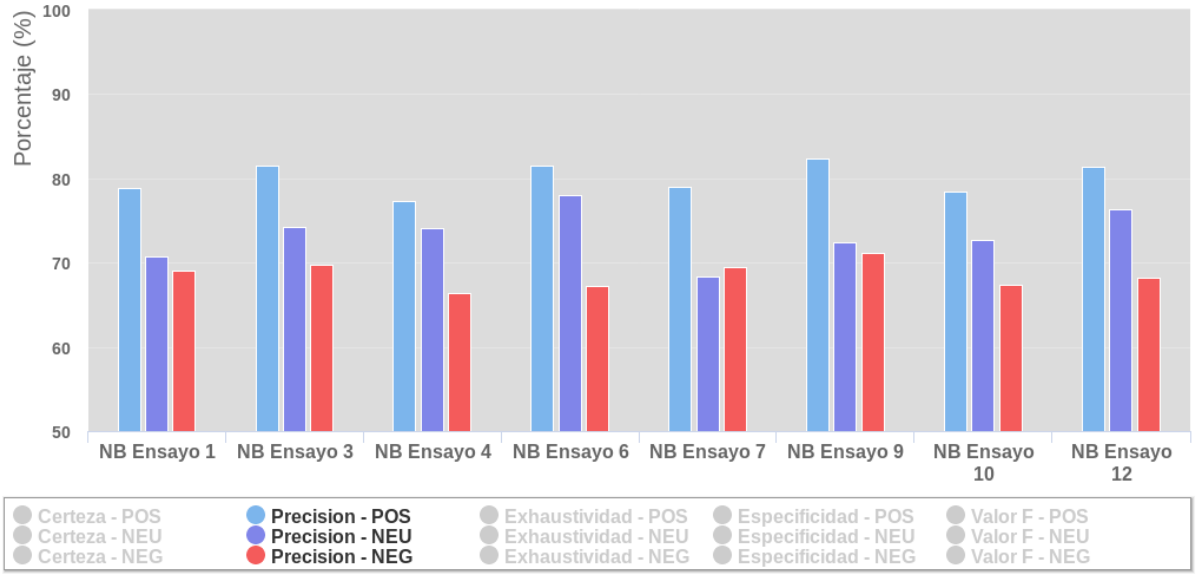


Gráfico 11. Valores de precisión de NB para cada ensayo.

Puede observarse una ventaja de los positivos frente a los neutrales y negativos. Sin embargo, esta diferencia no se mantiene de igual manera en todos los casos, ya que cada clase se comporta distinto frente a los cambios de parámetros. Los positivos se ven afectados únicamente por la variación en los folds, presentando una leve mejoría a mayor número de folds. Los neutrales, en cambio, además de mejorar junto con el aumento de folds, varían de manera favorable a menor tamaño del BoW. Contrariamente, conforme disminuye el tamaño del BoW (ensayos 4-6 y 10-12), decrece la precisión en los negativos.

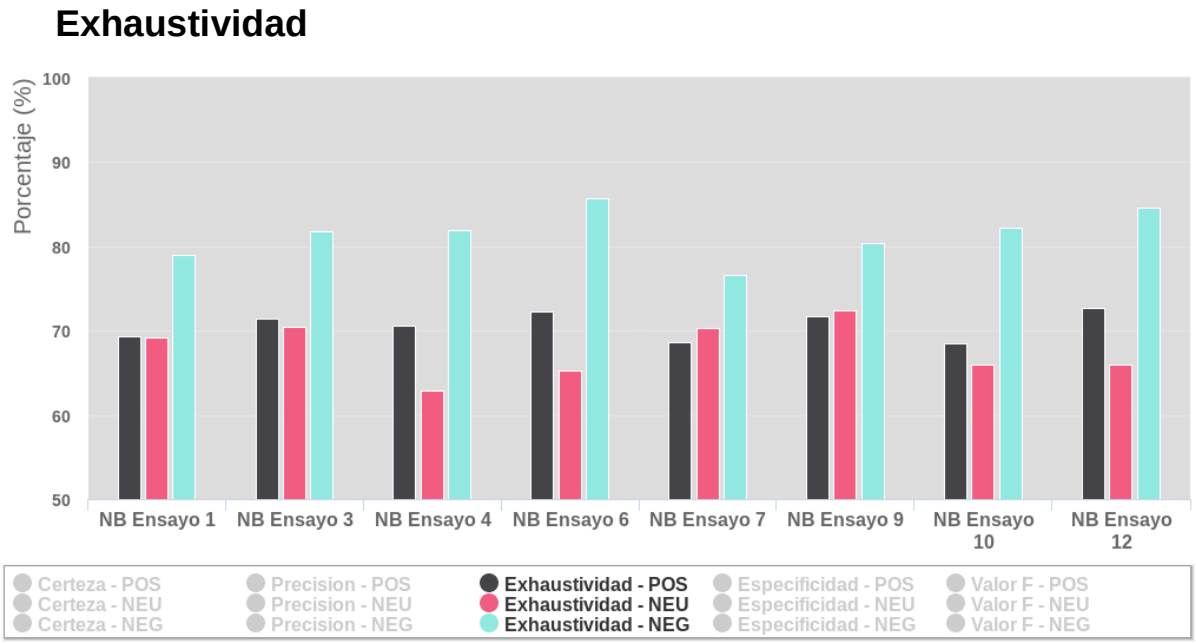


Gráfico 12. Valores de exhaustividad de NB para cada ensayo.

A simple vista resalta la exhaustividad en los negativos por sobre las otras dos, diferencia que es aún mayor con menor tamaño del BoW (4-6 y 10-12). La clase positiva no se ve afectada por ningún parámetro, salvo la cantidad de folds, variación que es común a todos los ensayos, métricas y técnicas. Por último, en clase neutral se observa un decremento en los ensayos donde se optó por un BoW de menor tamaño.

Teniendo en cuenta el incremento de esta métrica para la clase negativa, y el decremento para los neutrales en los ensayos donde se reduce el tamaño del BoW, podemos plantear como hipótesis que, al reducir el tamaño de datos sobre los cuales la técnica puede tomar decisiones (cantidad de tokens en el BoW), ésta tiene una tendencia a predecir más comentarios como negativos. Es decir, la mayor cantidad de confusiones son de neutrales como negativos.

Gráfico 13. Valores de especificidad de NB para cada ensayo.

Al igual que lo observado en el gráfico de exhaustividad, aquí la especificidad en los positivos se ve afectada únicamente por la variación en la cantidad de folds, e incluso esta diferencia es casi imperceptible. Vemos también que reduciendo el tamaño del BoW, aumenta la especificidad en los neutrales y disminuye en los negativos. Esto, acompañado con los análisis de precisión y exhaustividad ya realizados, deja en evidencia la tendencia negativa por parte de la técnica al disminuir el tamaño del BoW.

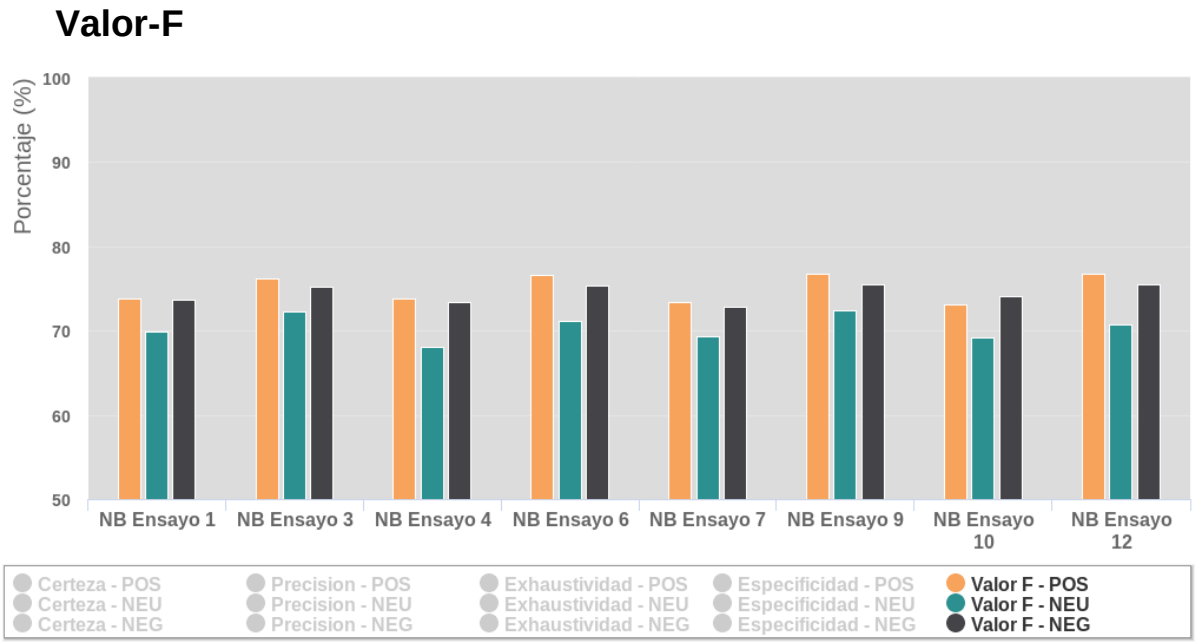


Gráfico 14. Valores de valor-F de NB para cada ensayo.

En cuanto al Valor-F, en líneas generales se mantiene constante (dejando de lado la mejora producto del aumento de folds). Aquí la diferencia más notable es en los neutrales, que se ven beneficiados levemente a mayor tamaño del BoW. Podemos concluir entonces que aumentar la cantidad de características proporcionadas a la técnica para cada caso (tamaño del BoW), beneficia su capacidad de discernir entre las clases neutral y negativa.

6.4.3 Máxima Entropía

Certeza

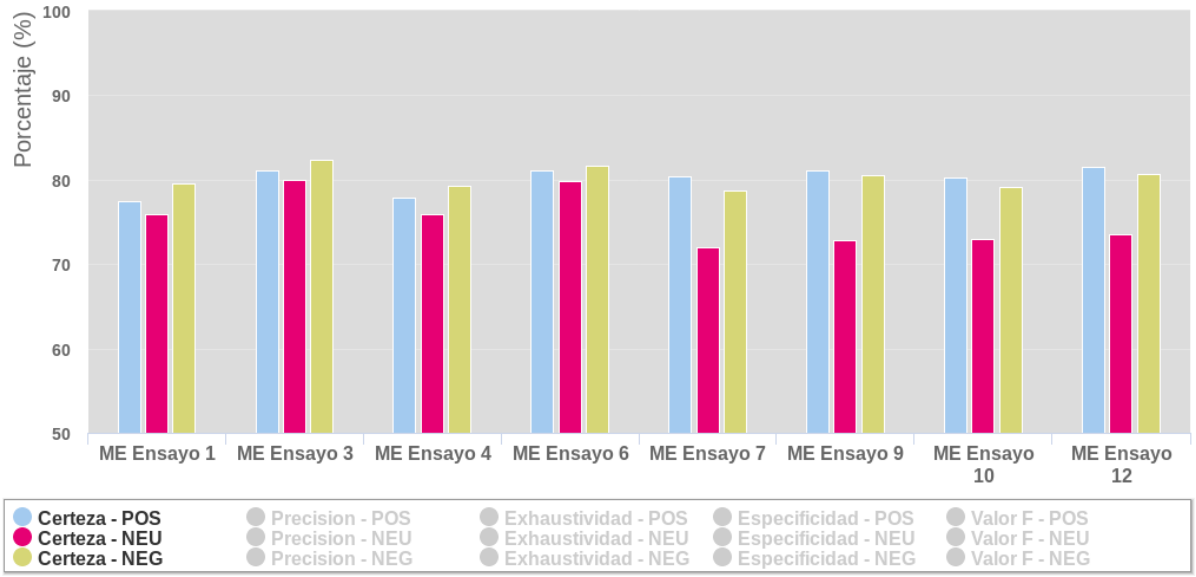


Gráfico 15. Valores de certeza de ME para cada ensayo.

En los ensayos 7, 9, 10 y 12, vemos que las tres clases se mantienen invariantes ante la alteración de los parámetros. No sucede lo mismo en los ensayos 1-6, donde sí se percibe una mejora a nivel general, cuando se incrementa la cantidad de folds.

A nivel de clases, vemos que en positivos existe una leve mejora en los ensayos 7-12. En cuanto a los negativos, vemos que es mayor en los ensayos 1-6, donde sólo fueron considerados los comentarios con un mínimo de 2 tokens. Ahora bien, los neutrales se ven perjudicados al incluir los comentarios con 1 token dentro de los casos de entrenamiento y testeo (ensayos 7-12).

Precisión

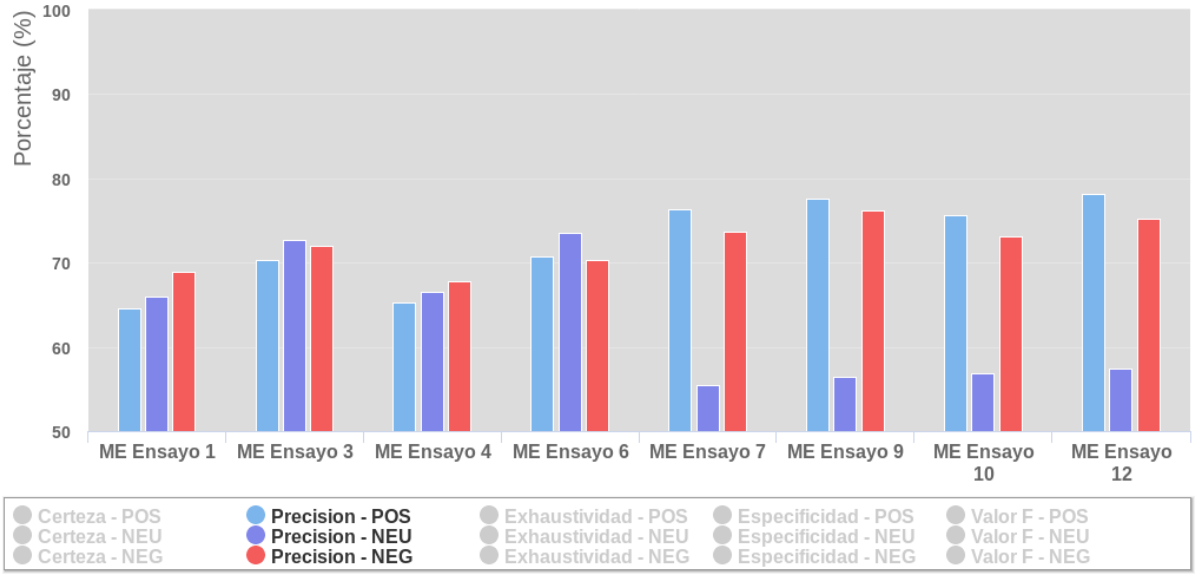


Gráfico 16. Valores de precisión de ME para cada ensayo.

Las precisiones de positivos y negativos presentan una mejoría en los ensayos 7 al 12 respecto a los anteriores. Con los neutrales sucede lo contrario, disminuyendo su precisión hasta en 15 puntos en comparación a los ensayos 1 al 6. Este fenómeno da cuenta que si se consideran todos los comentarios para el entrenamiento y testeo, se perjudica al modelo en líneas generales.

Analizando los ensayos 1 al 6, sólo se observa la típica mejora producto del aumento de folds, ya que en cuanto a la variación del tamaño del BoW, ésta no produce cambios significativos en los resultados.

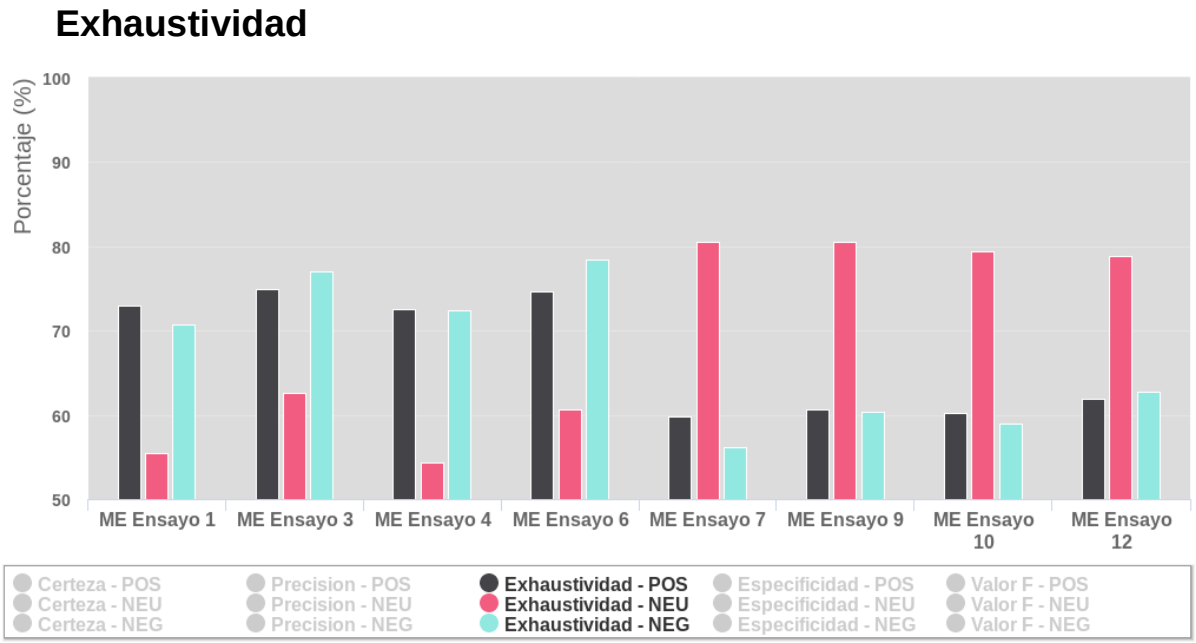


Gráfico 17. Valores de exhaustividad de ME para cada ensayo.

A nivel general, no se perciben cambios sustanciales con respecto al tamaño de BoW. Como en las métricas anteriores, se observa una mejora cuando el número de iteraciones es mayor.

Sí podemos ver una clara diferencia en el comportamiento de la técnica entre los ensayos donde se modificó el filtro por cantidad de tokens (1-6 y 7-12). En el primer caso, se obtienen valores razonables para las clases positivo y negativo, y valores muy bajos para los neutrales. En el segundo caso, podemos suponer que existe una tendencia hacia la clase neutral por parte del modelo, debido al cambio abrupto de los valores, centrándose mayormente en esta última. Para analizar con mayor profundidad este fenómeno, es necesario incluir el gráfico de precisión. Con estos datos, podemos observar que la cantidad de casos que fueron clasificados como pertenecientes a la clase neutral en los ensayos 7, 9, 10 y 12 es mucho mayor que para el resto de las clases.

Gráfico 18. Valores de especificidad de ME para cada ensayo.

Los valores de especificidad en los ensayos con un mínimo de 2 tokens son similares para las tres clases, y tomando en cuenta la misma similitud para la métrica de precisión, podemos ver que los modelos obtenidos en los ensayos 1-6 no poseen una tendencia significativa hacia ninguna clase. No sucede lo mismo en los ensayos 7-12, donde la especificidad de los neutrales cae por debajo del 70%, mientras que en los positivos y negativos alcanza el 90%. Si además tomamos en cuenta lo analizado en las métricas anteriores, vemos que todos los modelos obtenidos en estos últimos ensayos adquirieron una clara tendencia hacia la clase neutral.

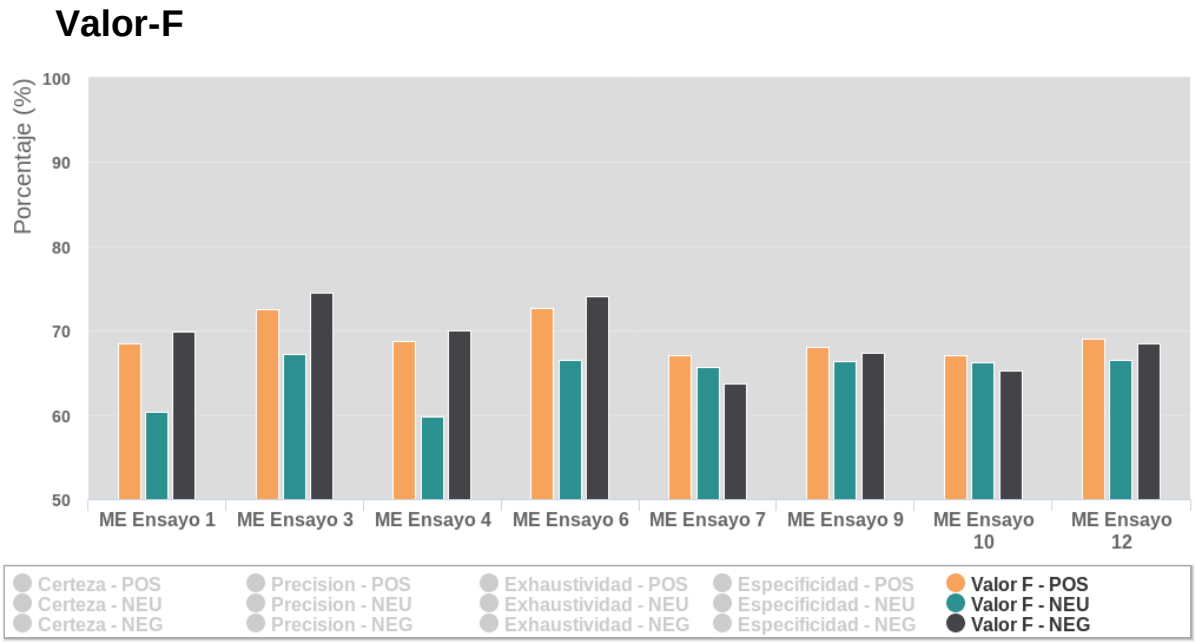


Gráfico 19. Valores de valor-F de ME para cada ensayo.

A simple vista, parecería ser que los ensayos 7-12 tuvieron resultados más consistentes, al ser más equitativos los valores entre clases. Pero no debemos olvidar que esta métrica es producto del promedio entre la precisión y la exhaustividad. Y como vimos, en los ensayos 7-12, estas métricas mostraron grandes diferencias por parte de la clase neutral respecto del resto, producto de la preferencia de los modelos por la clase intermedia.

Si bien los ensayos 1-6 poseen un valor-f aceptable para las clases positivo y negativo, para los neutrales es muy bajo. Esto se debe a que, manteniendo valores equitativos para las precisiones, en estos ensayos la exhaustividad en la clase de neutrales es notablemente inferior.

Podemos concluir entonces, que la técnica de Máxima Entropía enfrenta grandes dificultades a la hora de diferenciar comentarios neutrales, dado que o bien, los ignora (como sucede en los ensayos 1-6), o asume que la mayoría de los comentario son neutrales (ensayos 7-12).

6.4.4 Máquinas de Vectores de Soporte

Certeza

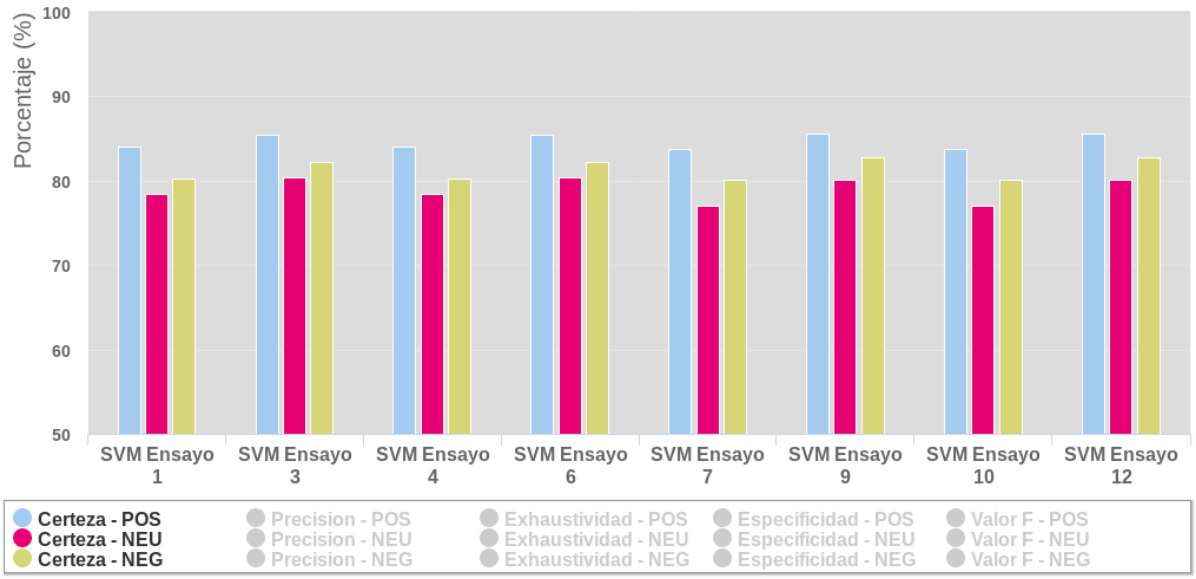


Gráfico 20. Valores de certeza de SVM para cada ensayo.

Ocurre algo similar con la certeza de Naïve Bayes. Hay una mejora en los ensayos con mayor cantidad de iteraciones (ensayos 3, 6, 9 y 12) ya que se entrenan con un conjunto de datos mayor. A su vez también existe una leve mejora cuando se filtran los comentarios que tienen como mínimo dos tokens con respecto a los tienen mínimo uno. Siempre la certeza de la clase positiva se mantiene por encima de las otras dos, seguida por la clase negativa.

Precisión

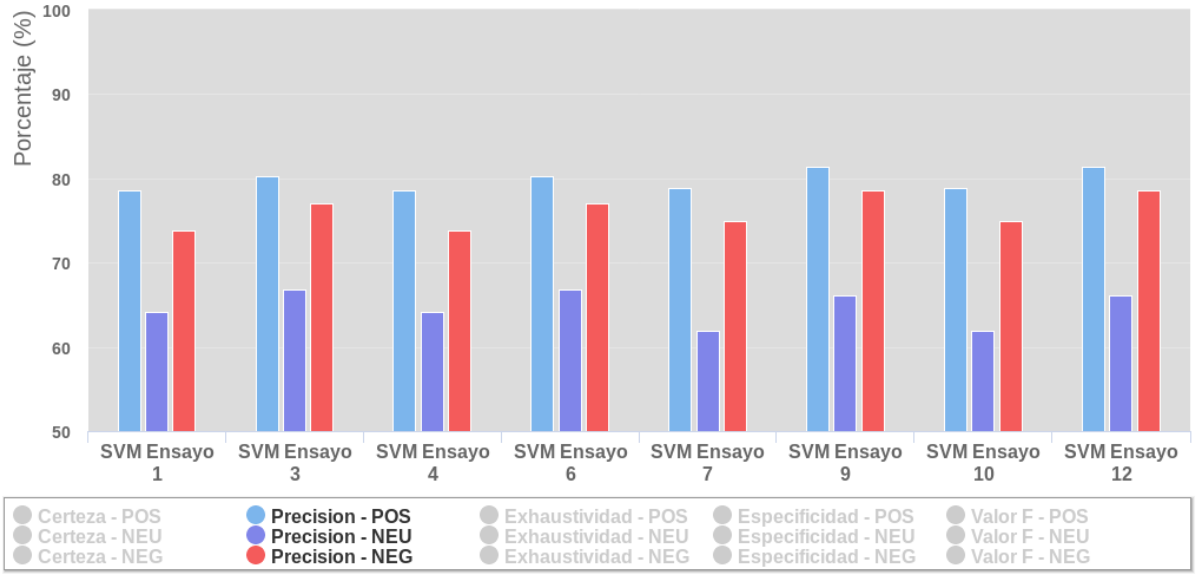


Gráfico 21. Valores de precisión de SVM para cada ensayo.

Se observa de nuevo una leve superioridad de la precisión de los positivos, seguida por los negativos. La precisión de los neutrales es muy baja con respecto a las otras dos. Se puede decir que esto se debe a que la técnica predice como neutrales, comentarios que pertenecen a las otras clases.

Así mismo, la precisión de los neutrales baja, aunque insignificantemente, cuando la cantidad de tokens mínima a evaluar es uno. En este caso, las otras dos precisiones aumentan, nuevamente de manera poco sustancial.

También hay una mejora en todas las precisiones con respecto al tamaño del BoW, cuando este es más chico.

Exhaustividad

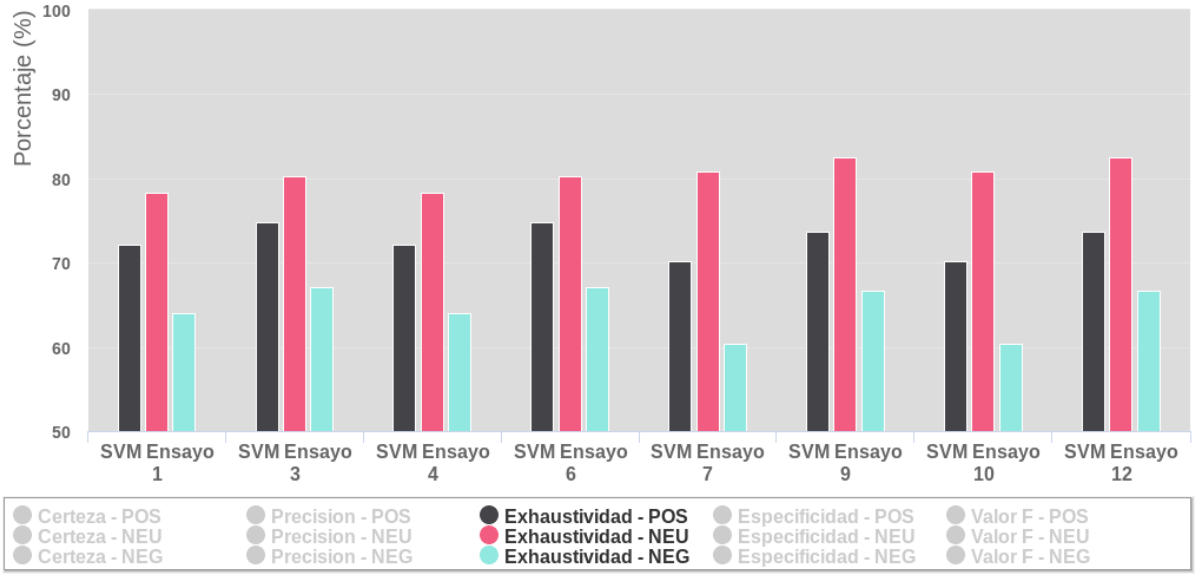


Gráfico 22. Valores de exhaustividad de SVM para cada ensayo.

Se nota primeramente, la ventaja de los neutrales por sobre las otras clases. Esto deja entrever, junto con la baja precisión que evaluó anteriormente, que la técnica tiene una tendencia a etiquetar más neutrales.

Existe un leve incremento en la exhaustividad de todas las clases cuando aumenta el número de iteraciones. Esto ocurre porque tiene un mayor conjunto para entrenarse.

El comportamiento con respecto a la cantidad mínima de tokens tiene una leve mejora en los ensayos 1-6 (donde se utilizaron únicamente comentarios con un mínimo de dos).

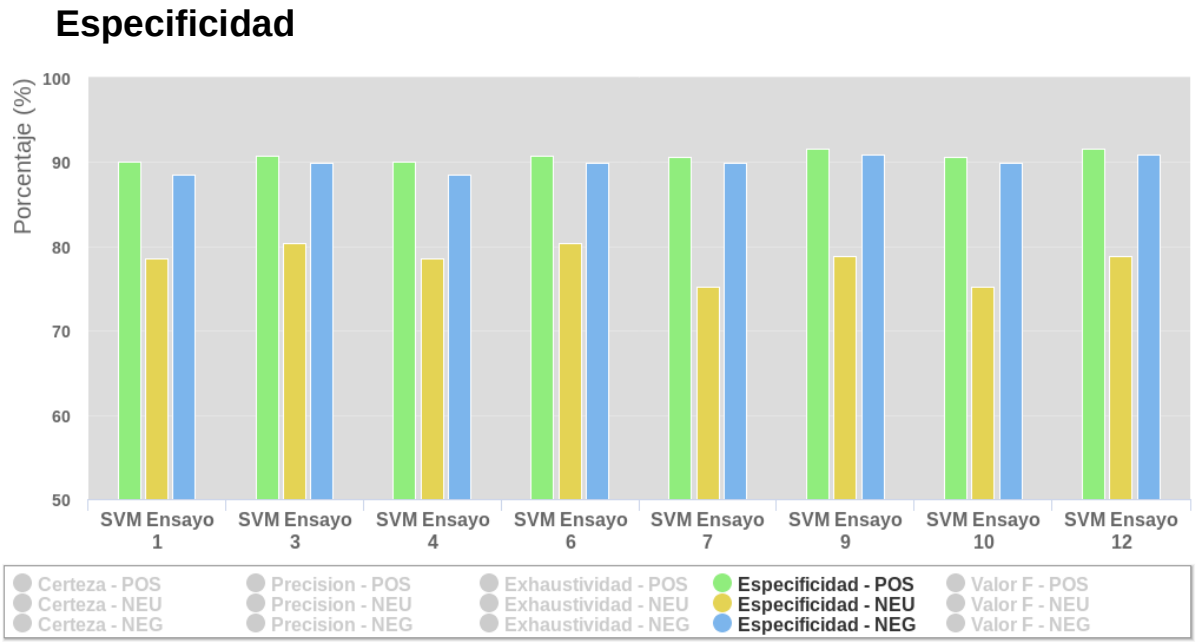


Gráfico 23. Valores de especificidad de SVM para cada ensayo.

Como se describió y concluyó en el Gráfico 22, esta técnica tiene una tendencia a etiquetar casos como neutrales, ya que en este caso, esta última clase es inferior a las otras dos. Presenta un leve decrecimiento cuando la cantidad mínima de tokens es uno y a su vez cuando aumenta el tamaño del BoW, mientras que los valores para las otras dos son parejos y se mantienen iguales.

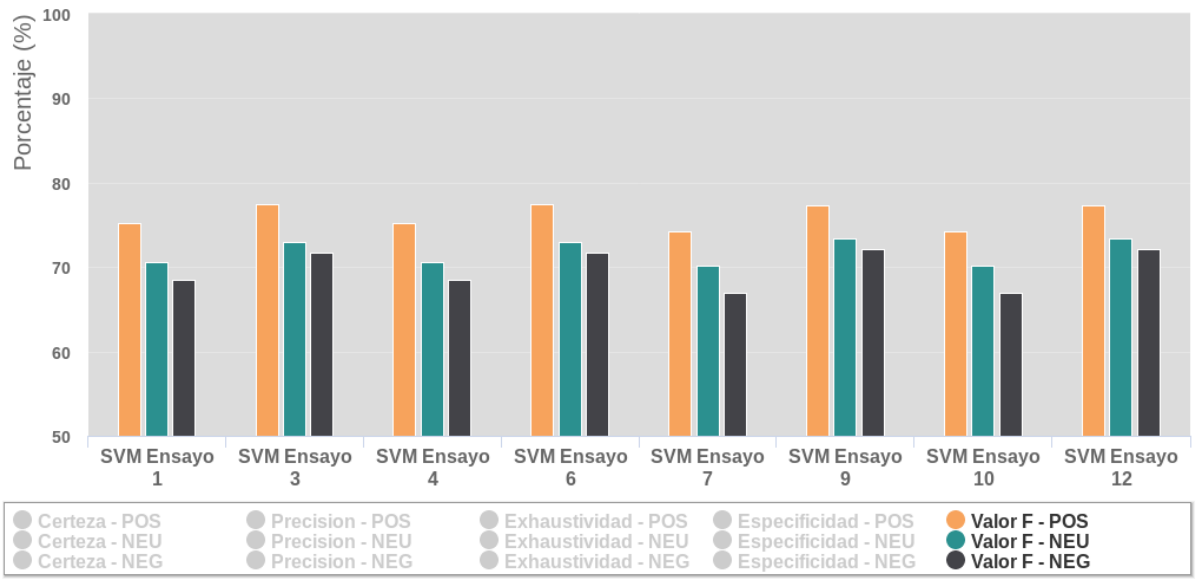


Gráfico 24. Valores de valor-F de SVM para cada ensayo.

En general, existe un leve incremento cuando el número de folds es mayor, que se aprecia mejor en los negativos, cuyo valor se aproxima al de los neutrales. Los positivos se mantienen siempre por sobre las otras dos clases.

Con respecto a la cantidad mínima de tokens, en los ensayos 1-6 la técnica se comporta de manera similar que los segundos, al igual que algunas de las otras métricas. Concluimos entonces que este parámetro no es significativo. Lo mismo sucede con el menor tamaño de BoW. Por último podemos notar diferencias en los ensayos que tienen mayor número de folds, por lo que éste es un valor que sí aporta una mejora a los modelos.

6.4.5 Conclusión final

Con el objetivo de realizar una comparación final entre las técnicas de aprendizaje supervisado, se seleccionaron dos ensayos que difieren en todos los parámetros, el 1 y el 12. A continuación se grafican los resultados de las métricas certeza, precisión, exhaustividad y especificidad para cada clase.

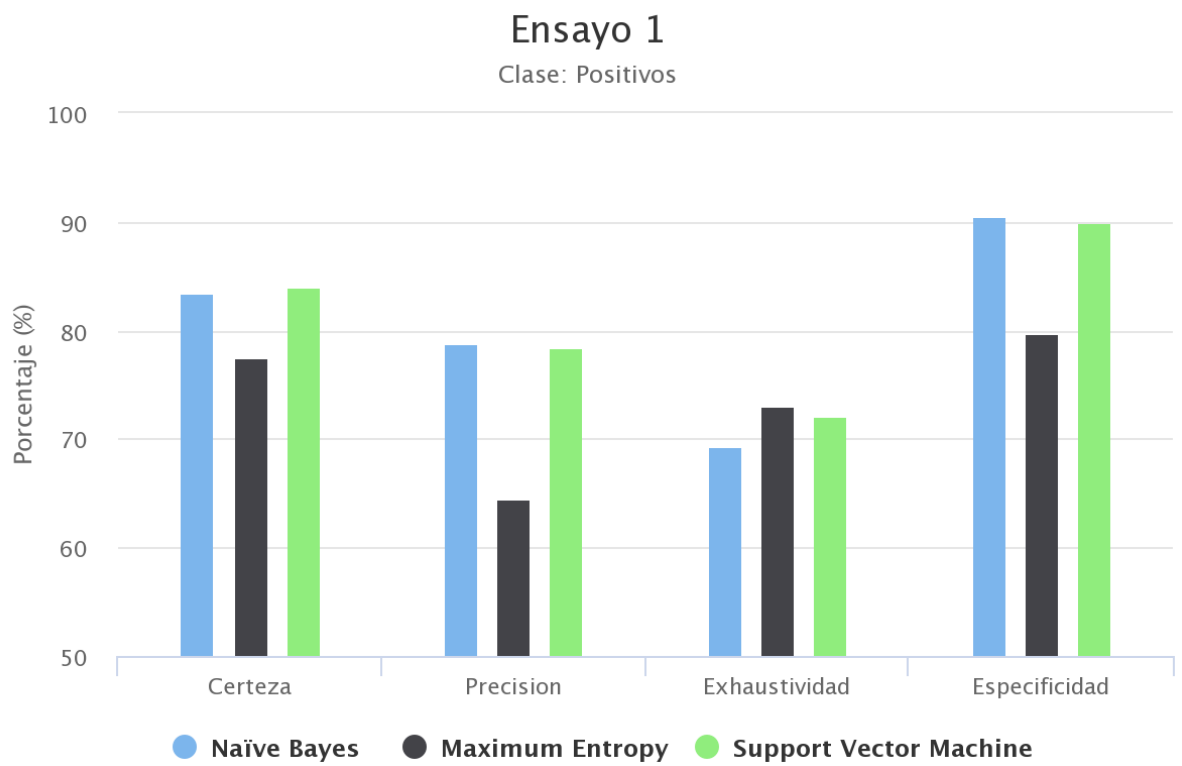


Gráfico 25. Valores de certeza, precisión, exhaustividad y especificidad para la clase positiva de las técnicas en el ensayo 1.

Se observa a simple vista, una inferioridad de ME con respecto a las otras dos técnicas para predecir positivos, ya que tuvo una menor certeza, muy baja precisión y especificidad. NB y SVM, se mantiene muy parejas entre sí en cada métrica.

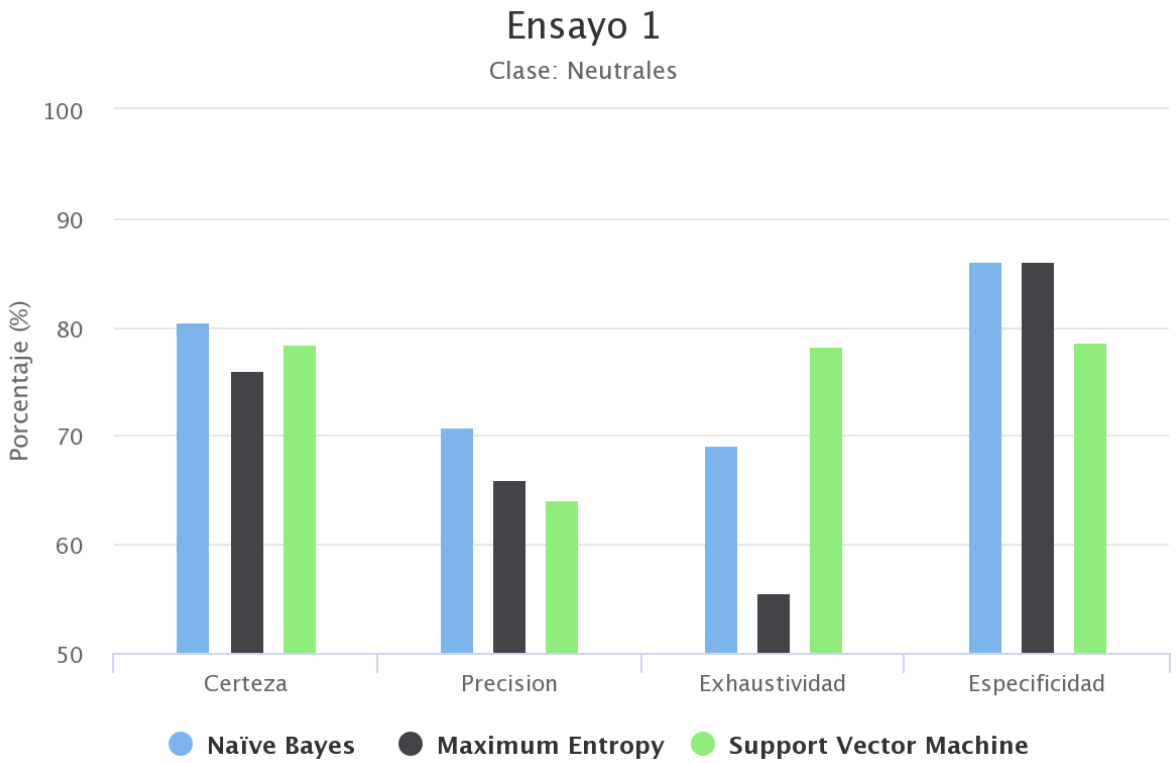


Gráfico 26. Valores de certeza, precisión, exhaustividad y especificidad para la clase neutral de las técnicas en el ensayo 1.

En cuanto a la clase neutral, NB presenta mejores resultados que las otras dos técnicas en la mayoría de las métricas. Tiene mayor certeza, precisión y especificidad que SVM, a diferencia del anterior gráfico, donde tenían valores similares.

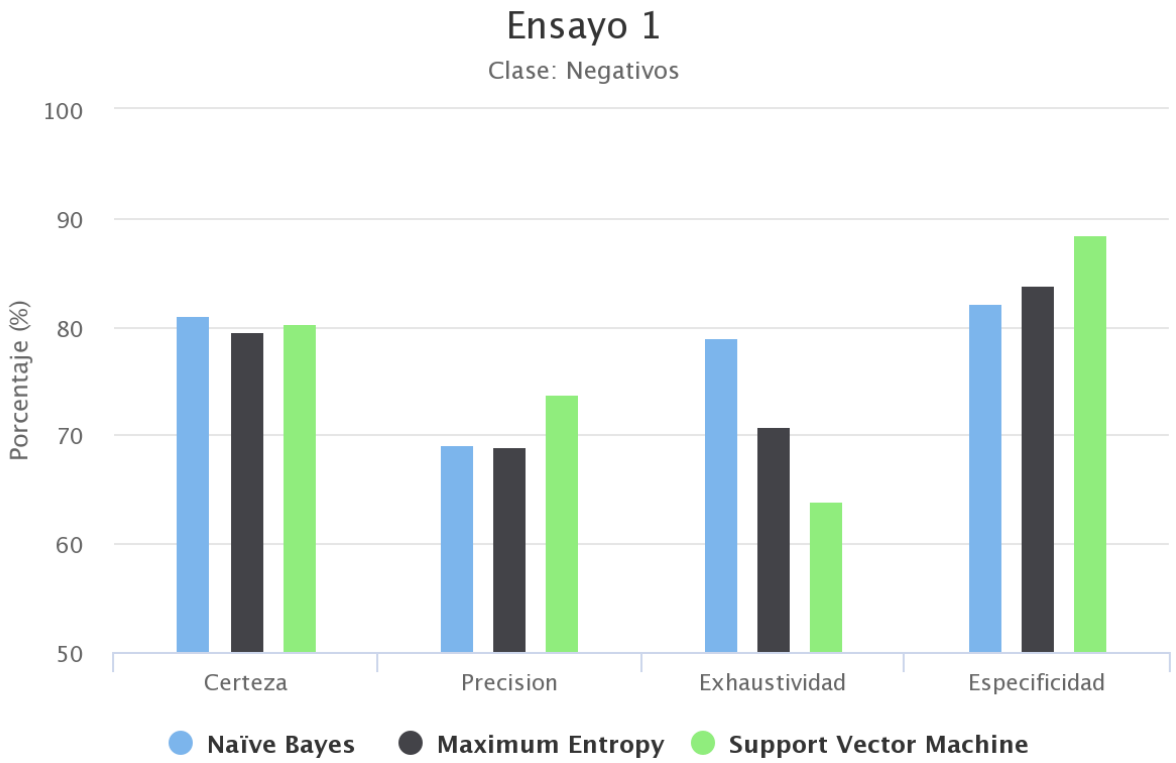


Gráfico 27. Valores de certeza, precisión, exhaustividad y especificidad para la clase negativa de las técnicas en el ensayo 1.

En este gráfico podemos ver que las tres técnicas se comportaron de manera muy similar, exceptuando la exhaustividad, en la que NB tuvo mejores resultados, seguida por ME. SVM se mantuvo por encima de las otras dos en cuanto a precisión y especificidad. NB lo hizo en cuanto a certeza y exhaustividad.

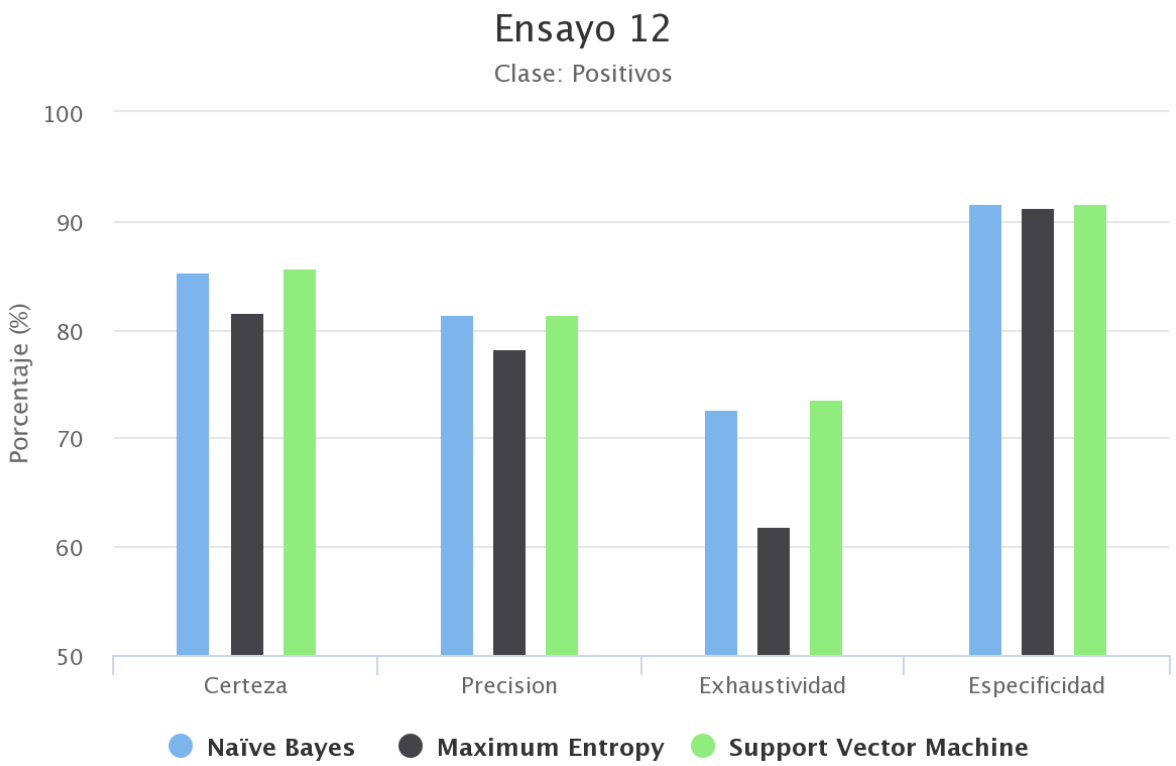


Gráfico 28. Valores de certeza, precisión, exhaustividad y especificidad para la clase positiva de las técnicas en el ensayo 12.

A diferencia del ensayo anterior, aquí ME denota una clara inferioridad únicamente en la exhaustividad, arrojando valores muy similares a las otras dos técnicas para el resto de las métricas. Esto se debe a la tendencia que desarrolla al ser entrenada con la totalidad de los comentarios, como ya se analizó.

NB y SVM presentaron una leve mejoría en todas las métricas, como ya hemos visto, por un mayor número de folds.

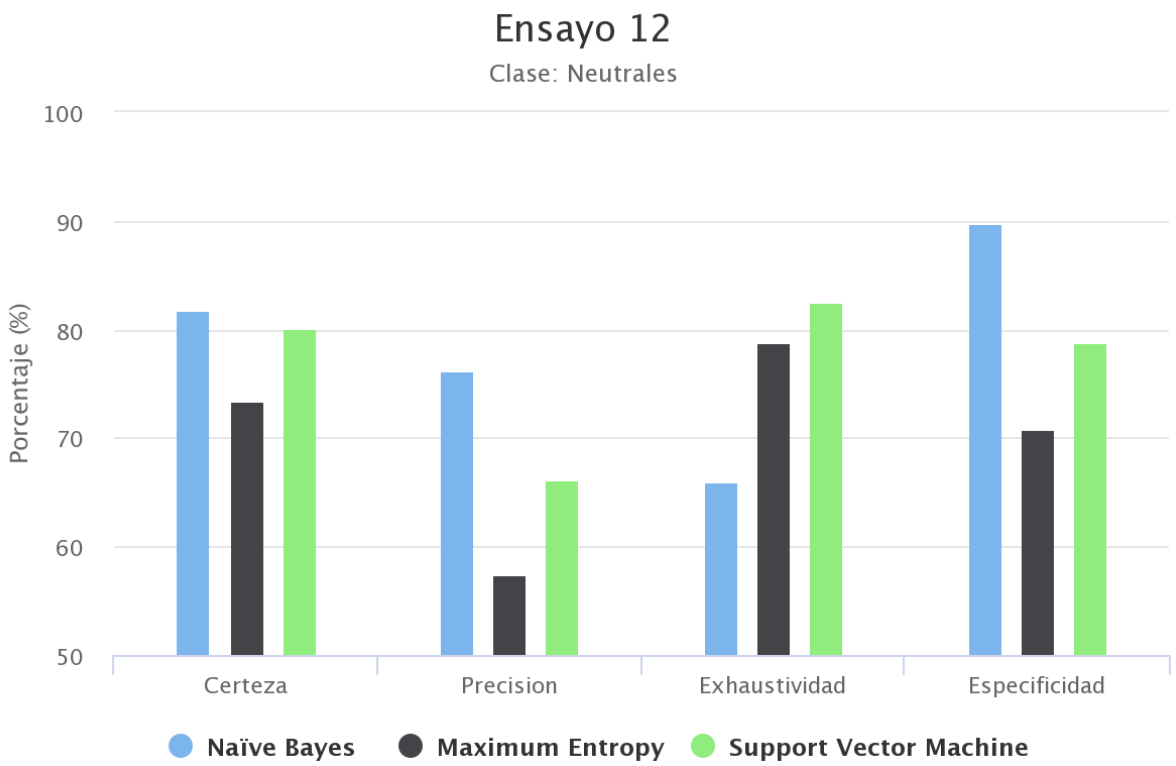


Gráfico 29. Valores de certeza, precisión, exhaustividad y especificidad para la clase neutral de las técnicas en el ensayo 12.

Con respecto a la clase neutral, nuevamente es clara la tendencia de ME a sobre etiquetar neutrales cuando se entrena con todos los comentarios, por la alta exhaustividad y mantenerse inferior en el resto de las métricas. En líneas generales, NB es superior.

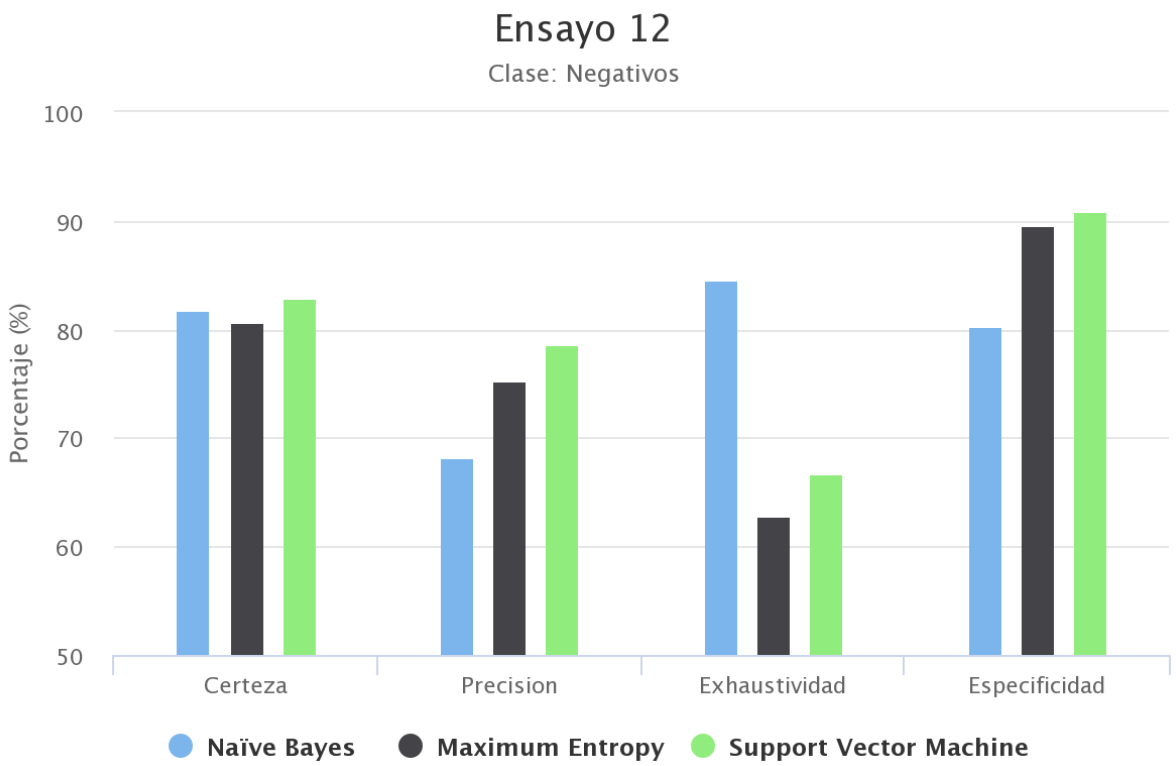


Gráfico 30. Valores de certeza, precisión, exhaustividad y especificidad para la clase negativa de las técnicas en el ensayo 12.

NB presenta una leve tendencia a etiquetar negativos, por la alta exhaustividad y baja precisión y especificidad, mostrando valores muy distintos a las otras dos técnicas.

Valor-F de cada técnica

Como conclusión final, comparando todas las técnicas, se seleccionaron los ensayos donde cada una obtuvo el mejor valor- F, con el propósito de comparar el desempeño general de las mismas.

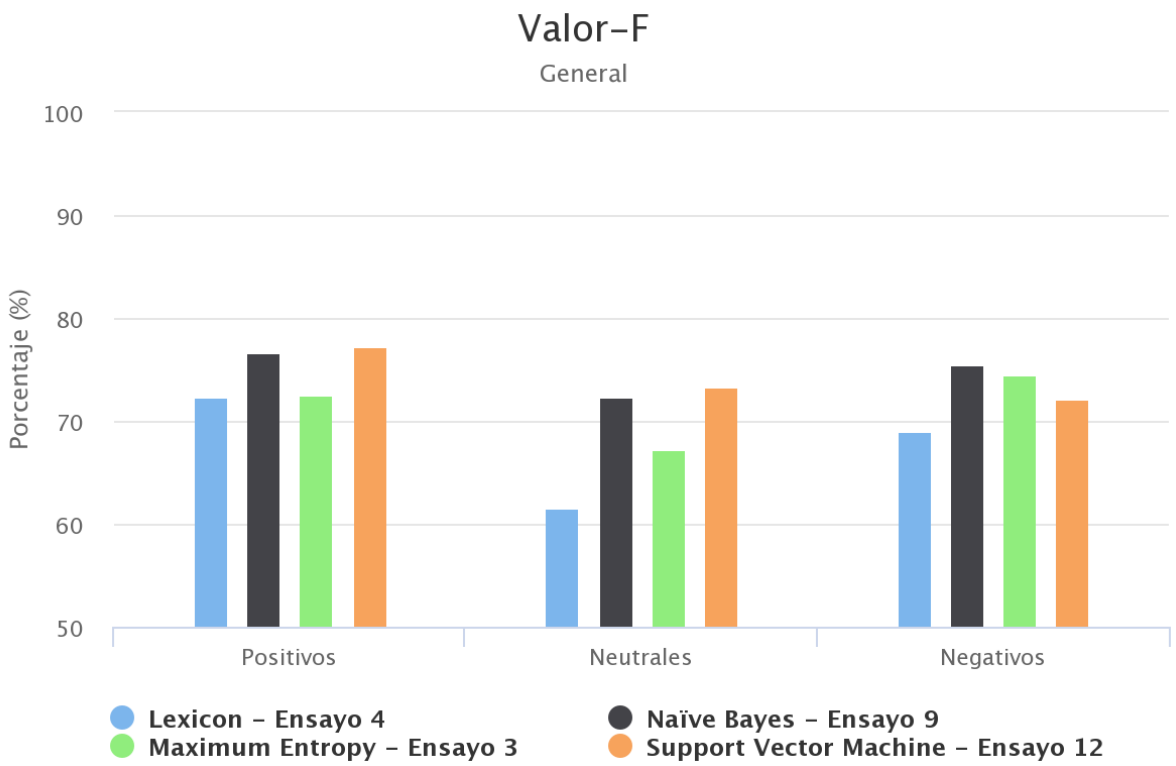


Gráfico 31. Ensayos con mejores valores de valor- F de cada técnica.

Siendo estos los mejores valores obtenidos entre todos los ensayos realizados para cada técnica, podemos ver que el de peor desempeño fue lexicon. Respecto de las técnicas de aprendizaje supervisado, como era de esperarse ME obtuvo los peores resultados, más que nada en la clase neutral. En cuanto a NB y SVM, en líneas generales obtuvieron valores muy similares destacándose por sobre el resto.

En el caso de NB, los mejores resultados fueron obtenidos en el ensayo 9, donde el entrenamiento se realizó con la totalidad de los comentarios, mayor tamaño del BoW y mayor número de folds.

SVM en cambio, obtuvo el mejor desempeño en el ensayo 12, donde también fue utilizado el total de los comentarios y mayor número de folds, pero con un menor tamaño del BoW.

Como se pudo observar, en todos los análisis realizados, ME tuvo los peores resultados, y si a esto le sumamos el tiempo que tarda en entrenarse por las iteraciones que debe ejecutar (30 minutos aproximadamente contra 30 segundos de SVM y 10 segundos de NB), creemos que son motivos suficientes para descartarla a la hora de culminar la aplicación que desarrollamos.

Capítulo 7

Conclusiones y trabajos futuros

7.1 Repaso

Debido a lo investigado y estudiado a lo largo de este trabajo, consideramos que tanto el Análisis de Sentimientos como el Procesamiento de Lenguaje Natural, son temas muy interesantes e instalados: Google o Microsoft son ejemplos de grandes compañías que ofrecen servicios de estas herramientas. A su vez, son muy recientes y todavía se encuentran en crecimiento. Como existen varios aportes ricos en contenido y desarrollo, también existen temas por abordar y pulir, como es el entendimiento del contexto, por parte del análisis pragmático, para poder reconocer ambigüedades y sarcasmo, entre otras limitaciones. Además de poder extender los aportes en Español, contemplando las diferentes terminologías según el dialecto, y seguramente en otros idiomas que no son el Inglés.

Lo que nos llamó la atención y nos interesó de estos temas fue el gran salto y aporte que brindan a raíz del gran crecimiento de información que actualmente existe. No solo en las redes sociales, como hemos estudiado, sino también en reseñas de productos y encuestas de todo tipo, desde políticas hasta de satisfacción. Además resultan de gran interés tanto para la comunidad educativa como para las empresas y tienen diversas aplicaciones como ya hemos visto. Algunas de ellas, en el caso de NLP, comprenden desde descubrimiento investigativo hasta traducción. En SA, por otra parte, van desde clasificación de la opinión de un usuario en Internet hasta reconocer la temática de un documento.

Como hemos mencionado, las redes sociales son cada vez más utilizadas por muchos y distintos usuarios, generando mucha actividad que puede manifestarse de diversas formas, ya sea compartiendo su estado de ánimo, opinando respecto a diversos temas y reaccionando al contenido publicado, entre algunos ejemplos. Toda esta participación e interacción no garantiza la calidad de la información. La exposición selectiva, de hecho, juega un papel fundamental en el consumo de

contenido y la difusión de información. Los usuarios tienden a seleccionar información a la que adhiere, reforzando su visión del mundo e ignora la información disidente. Este patrón provoca la formación de grupos polarizados donde la interacción con personas de ideas afines podría incluso intensificar la polarización. Se puede mostrar que los patrones de consumo presentan comunidades en distintos medios informativos, utilizando una técnica que combina la extracción automática de temas y el Análisis de Sentimientos, para caracterizar mejor la dinámica interna del grupo. Así, se podría comparar cómo se presentan los mismos temas en las publicaciones y la respuesta emocional relacionada en los comentarios y que la polarización influye en la percepción de los temas.

7.2 Conclusiones y trabajos futuros

A lo largo del capítulo anterior, analizamos los resultados de las distintas métricas para todos los ensayos realizados. A continuación exponemos las conclusiones a las que llegamos.

Luego del análisis realizado sobre la técnica basada en diccionario, pudimos observar el gran impacto que tienen los valores seleccionados como límites entre las clases. Llegando a disminuir la diferencia entre precisiones de cada clase hasta en un 20%. Esta técnica también se vio beneficiada al filtrar las stopwords, reduciendo el ruido que éstas podrían llegar a producir en la valoración de la polaridad. Si bien los resultados finales obtenidos en el mejor de los ensayos (ensayo 4) no son óptimos, a lo largo del presente documento se nombraron diversas formas de mejorar el rendimiento de esta técnica. Un ejemplo simple es la implementación propuesta en [50], donde se le da mayor importancia a las palabras mientras más cercanas al final de la frase se encuentren. Otros ejemplos que requieren un análisis, investigación y desarrollo más exhaustivo consisten en la detección de negaciones y sarcasmo dentro del texto, siendo estos últimos parte del estado del arte en materia de NLP y SA.

Con respecto a las técnicas de aprendizaje supervisado, en general y como era de esperarse, tuvieron una mejoría cuando mayor fue el número de folds de la validación cruzada, ya que contaban con un conjunto de entrenamiento más amplio.

Particularmente en el caso de la técnica de Naïve Bayes, llegamos a la conclusión que aumentar la cantidad de características proporcionadas, es decir el tamaño del BoW, beneficia su capacidad de discernir entre las clases neutral y negativa.

Respecto de la técnica de Máxima Entropía, luego de analizar los resultados arrojados por los ensayos, concluimos que ésta enfrenta grandes dificultades a la hora de diferenciar comentarios neutrales, dado que o bien, los ignora (como sucede en los ensayos 1 al 6), o asume que la mayoría son neutrales (ensayos 7 al 12).

Por parte de la técnica de Máquinas de Vectores de Soporte, existe una leve mejora cuando se filtran los comentarios, utilizando sólo aquellos que tienen como mínimo dos tokens, y cuando el tamaño del BoW es menor.

En líneas generales, los mejores resultados obtenidos a lo largo de los ensayos fueron los de NB y SVM, siendo ambos muy similares.

Con el objetivo de mejorar los resultados, se podría extender el conjunto de comentarios etiquetados, y así ampliar el volumen de datos utilizado para el entrenamiento de las técnicas. También es posible aplicar distintas estructuras para el BoW (como TF-IDF), o métodos de entrenamiento y validación alternativos (como por ejemplo Leave-One-Out [54]), para ver si mejoran las técnicas.

Gracias a la aplicación desarrollada, pudimos concluir con nuestro objetivo principal de estudiar y comparar diferentes técnicas de Análisis de Sentimiento y Procesamiento de Lenguaje Natural. Se combinaron estos dos grandes temas en una herramienta capaz de seleccionar o filtrar los comentarios de los usuarios ante las publicaciones, pudiendo focalizar su uso tanto a una empresa que desee hacer un estudio de marca o servicio, como así también a una persona pública que esté interesada en conocer su imagen en la sociedad, cómo es su connotación. Conforme fuimos avanzando en la realización del desarrollo propuesto en este trabajo, y considerando el cambio de rumbo que tuvimos que afrontar, nos planteamos los siguientes posibles trabajos a futuros:

- Ver la influencia que pueden tener los usuarios entre sí dentro de un hilo de discusión, y analizar si esto genera cambios en la opinión de los mismos.
- Analizar la posibilidad de agrupar usuarios en base a la similitud de su valoración sobre los noticias/publicaciones.
- Analizar la posibilidad de extraer información sobre el usuario (por ejemplo la edad, el género o el lugar de residencia), y así poder realizar un ranking de

noticias/publicaciones más aceptadas o rechazadas para grupos con ciertas características similares.

- Estudiar la posibilidad de analizar el perfil de los usuarios (publicaciones, comentarios, etcétera), dentro de su red de amigos y así poder formar para cada uno un “perfil personal” que represente sus ideologías/creencias. Esto sería realizado con el fin de poder detectar cuando es sarcástico, irónico, etcétera, y así poder detectar falsos positivo/negativo y mejorar la predicción de las técnicas.
- Comparar el ranking de publicaciones de dos portales de noticias distintos, evaluando la aceptación o rechazo por parte de los usuarios.
- Permitir que el usuario que utilice la aplicación pueda reaccionar, según las que aporta Facebook, sobre los comentarios en tiempo real, persistiendo los mismos en dicho momento y luego realizar el entrenamiento de las técnicas en segundo plano sin afectar la usabilidad del sistema.
- Elegir las publicaciones a recolectar por un tópico en particular (política, economía, deportes, espectáculos, etcétera) con todos sus comentarios y reacciones.

Bibliografía

[1] Meire, M., et al. (2016). The added value of auxiliary data in sentiment analysis on Facebook posts. *Decision Support Systems, 89*, 98-112. doi: http://dx.doi.org/10.1016/j.dss.2016.06.013

[2] Dubiau, L., Ale, J. (2013). *Análisis de sentimientos sobre un corpus en español: Experimentación con un caso de estudio.* In: Proceedings of the 14th Argentine Symposium on Artificial Intelligence, ASAI, pp. 36–47 (2013). http://42jaiio.sadio.org.ar/proceedings/simposios/Trabajos/ASAI/04.pdf

[3] Ortigosa A., e. a. (2014). Sentiment analysis in facebook and its application to e-learning. *Computers in Human Behavior, 31*(10), 527-541. doi: http://dx.doi.org/10.1016/j.chb.2013.05.024

[4] Mejía Llano, J. C. (2017, Mayo 2). Estadísticas de redes sociales: Usuarios de Facebook, Instagram, LinkedIn, Twitter, Whatsapp y otros + Infografía. http://www.juancmejia.com/marketing-digital/estadisticas-de-redes-sociales-usuarios-de-facebook-instagram-linkedin-twitter-whatsapp-y-otros-infografia

[5] Caturini, L. (2017, Junio 21). En la Argentina Facebook tiene 20 millones de usuarios activos por mes. http://wp.enciclomedios.com/?p=18437

[6] Montejo-Ráez, A., e. a. (2012). Detección de la polaridad en citas periodísticas: una solución no supervisada. *Procesamiento del Lenguaje Natural, 49*, 149-156. http://rua.ua.es/dspace/handle/10045/23940

[7] Aisopos, F., e. a. (2016). *Using n-gram graphs for sentiment analysis: an extended study on Twitter.* Paper presented at 2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService), Oxford, UK. doi: http://dx.doi.org/10.1109/BigDataService.2016.13

[8] Barnaghi, P., et al. (2016). *Opinion Mining and Sentiment Polarity on Twitter and Correlation Between Events and Sentiment.* Paper presented at 2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService), Oxford, UK. doi: http://dx.doi.org/10.1109/BigDataService.2016.36

[9] Sidorov G. et al. (2012). *Empirical study of machine learning based approach for opinion mining in tweets.* Paper presented at MICAI 2012: Advances in Artificial Intelligence, San Luis Potosí, Mexico. doi: http://dx.doi.org/10.1007/978-3-642-37807-2_1

[10] Castro, C. L., Braga, A. P. (2013). Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems, 24*(6), 888-899. doi: http://dx.doi.org/10.1109/TNNLS.2013.2246188

[11] Dashtipour, K., Poria, S., Hussain, A., et al. (2016). Multilingual sentiment analysis: State of the art and independent comparison of techniques. *Congnitive Computation, 8*(4), 757-771. doi: http://dx.doi.org/10.1007/s12559-016-9415-7

[12] Serrano-Guerrero, j., et al. (2015). Sentiment analysis: A review and comparative analysis of web services. *Information Sciences, 311*, 18-38. doi: http://dx.doi.org/10.1016/j.ins.2015.03.040

[13] Yen, S.-J., Lee, Y.-S. (2009). Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications, 36*(3), 5718-5727. doi: http://dx.doi.org/10.1016/j.eswa.2008.06.108

[14] Han, J., Kamber, M. (2006). *Data Mining: Concepts and Techniques Third Edition.* Morgan Kaufmann Publisher: CA. ISBN: 13: 978-1-55860-901-3. http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf

[15] **Facebook for developers**. *API Graph*. https://developers.facebook.com/docs/graph-api/

[16] **Ana Nieto**. (2017). *Las 30 Redes Sociales más Utilizadas*. Webempresa20 - Internet orientado a resultados. http://www.webempresa20.com/blog/las-30-redes-sociales-mas-utilizadas.html

[17] **Gabriela González**. (2017, Agosto 23). *El estado de las redes sociales en 2017*. https://www.genbeta.com/a-fondo/el-estado-de-las-redes-sociales-en-2017

[18] **Ministerio de Educación del Gobierno de España**. *El uso de las redes sociales*.http://www.ite.educacion.es/formacion/materiales/112/cd/m7/el_uso_de_las_redes_sociales.html

[19] W. Medhat, A. Hassan & H. Korashy. (2014, Diciembre). *Sentiment analysis algorithms and applications: A survey*. Ain Shams Engineering Journal, Volume 5, Issue 4, 1093-1113. doi: https://doi.org/10.1016/j.asej.2014.04.011

[20] Obar, J.A. and Wildman, S. (2015). *Social media definition and the governance challenge: An introduction to the special issue*. Telecommunications Policy, 39(9),

745-750.; Quello Center Working Paper No. 2647377. Available at SSRN: https://ssrn.com/abstract=2647377 or http://dx.doi.org/10.2139/ssrn.2647377

[21] A. Richter, M. Koch (2008). *Functions of Social Networking Services*. In: Proc. 8TH International Conference on the Design of Cooperative Systems, Carry-le-rouet, France, Institut d'Etudes Politiques d'Aix-en-Provence, pp. 87-98. http://www.kooperationssysteme.de/docs/pubs/RichterKoch2008-coop-sns.pdf

[22] **Computer Hope**. (2017, Octubre 30). *Social network*. https://www.computerhope.com/jargon/s/socinetw.htm

[23] **Elise Moreau**. (2018, Julio 9). *The Top Social Networking Sites People Are Using*. https://www.lifewire.com/top-social-networking-sites-people-are-using-3486554

[24] **Priit Kallas**. (2018, Agosto 2). *Top 15 Most Popular Social Networking Sites and Apps [August 2018]* https://www.dreamgrow.com/top-15-most-popular-social-networking-sites/

[25] E. Mitchelstein, P. J. Boczkowski. (2018, Enero-Junio). *Juventud, estatus y conexiones: Explicación del consumo incidental de noticias en redes sociales*. Revista Mexicana de Opinión Pública, 24, pp. 131-145, ISSN 1870-7300. doi: 10.22201/fcpys.24484911e.2018.24.61647

[26] N. Indurkhya, F. J. Damerau. (2010). *Handbook of Natural Language Processing Second Edition*. International Standard Book Number-13: 978-1-4200-8593-8. https://karczmarczuk.users.greyc.fr/TEACH/TAL/Doc/Handbook%20Of%20Natural%20Language%20Processing,%20Second%20Edition%20Chapman%20&%20Hall%20Crc%20Machine%20Learning%20&%20Pattern%20Recognition%202010.pdf

[27] Luciana Dubiau. (2013, Octubre). *Procesamiento de Lenguaje Natural en Sistemas de Análisis de Sentimientos*. http://materias.fi.uba.ar/7500/Dubiau.pdf

[28] **Facebook for developers**. *Cambios radicales*. https://developers.facebook.com/docs/graph-api/changelog/breaking-changes/#7-de-febrero-de-2018

[29] C. Cherpas. (1992, Abril). *Natural language processing, pragmatics, and verbal behavior*. The Analysis of Verbal Behavior 1992, 10, 135-147. doi: 10.1007/BF03392880

[30] Eduardo Sosa. (1997, Enero). *Procesamiento del lenguaje natural: revisión del estado actual, bases teóricas y aplicaciones, Parte I*. Revista internacional científica y profesional. ISSN 1386-6710.

http://www.elprofesionaldelainformacion.com/contenidos/1997/enero/procesamiento_del_lenguaje_natural_revisin_del_estado_actual_bases_tericas_y_aplicaciones_parte_i.html

[31] R. Collobert, J. Weston, et al. (2011). *Natural Language Processing (Almost) from Scratch*. Journal of Machine Learning Research 12 (2011) 2493-2537. http://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf

[32] B. Liu. (2012, Abril 22). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 167, ISBN 978-1-60845-884-4. https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf

[33] **Jack Rae**. (2015, Febrero 14). *What is a generative model?.* https://www.quora.com/What-is-a-generative-model

[34] **O. Veksler**. *Pattern Recognition*. http://www.csd.uwo.ca/~olga/Courses/CS434a_541a/Lecture8.pdf

[35] **Sunil Ray**. (2017, Septiembre 11). *6 Easy Steps to Learn Naive Bayes Algorithm (with codes in Python and R)*. https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[36] **Jason Brownlee**. (2016, Abril 11). *Naive Bayes for Machine Learning*. In Machine Learning Algorithms. https://machinelearningmastery.com/naive-bayes-for-machine-learning/

[37] **CHristianCH**. (2013, Mayo 2). *Clasificador Naïve Bayes. ¿Cómo funciona?*. http://naivebayes.blogspot.com/

[38] **Christopher Potts**. (2011). *Sentiment Symposium Tutorial: Classifiers*. http://sentiment.christopherpotts.net/classifiers.html#maxent

[39] K. Nigam, J. Lafferty, A. McCallum. (1999). *Using Maximum Entropy for Text Classification*. IJCAI-99, Workshop on Machine Learning for Information Filtering, pages 61–67. http://www.kamalnigam.com/papers/maxent-ijcaiws99.pdf

[40] Jana Álvarez. (2016, Diciembre 22). *Machine Learning y Support Vector Machines: porque el tiempo es dinero*. https://www.analiticaweb.es/machine-learning-y-support-vector-machines-porque-el-tiempo-es-dinero-2/

[41] **Support Vector Machines (SVM) Introductory Overview**. http://www.statsoft.com/Textbook/Support-Vector-Machines

[42] F. L. Cruz, J. A. Troyano, B. Pontes, F. J. Ortega. (2014, Septiembre). *ML-SentiCon: Un lexicón multilingüe de polaridades semánticas a nivel de lemas*. Procesamiento del Lenguaje Natural, Revista nº 53, pp 113-120. https://rua.ua.es/dspace/bitstream/10045/40031/1/PLN_53_12.pdf

[43] **Support Vector Machines**. http://scikit-learn.org/stable/modules/svm.html

[44] **Stark Overflow**. *Developer survey Results 2017*. https://insights.stackoverflow.com/survey/2017#technology-databases

[45] **Facebook for developers**. *Tokens de acceso*. https://developers.facebook.com/docs/facebook-login/access-tokens/

[46] **Sean Gallagher**. (2014, Septiembre 30). *How to get a never expiring Facebook Page Access Token*. https://www.rocketmarketinginc.com/blog/get-never-expiring-facebook-page-access-token/

[47] **Juan Merodio**. (2018, Marzo 14). *Facebook Open Graph*. https://www.juanmerodio.com/que-es-el-open-graph-de-facebook-y-como-podemos-utilizarlo-en-la-empresa/

[48] **Mohammed Sunasra**. (2017, Noviembre 11). *Performance Metrics for Classification problems in Machine Learning*. doi: https://medium.com/greyatom/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b

[49] *Welcome to Flask*. http://flask.pocoo.org/docs/1.0/

[50] Pang, B., Lee, L., & Vaithyanathan, S. (2002). *Thumbs up? Sentiment classification using machine learning techniques.* In Proceedings of Empirical Methods in Natural Language Processing (EMNLP) (pp. 79–86). http://www.cs.cornell.edu/home/llee/papers/sentiment.pdf

[51] *OAuth 2.0*. https://oauth.net/2/

[52] *Highcharts.* https://www.highcharts.com/docs

[53] *Understanding Highcharts.* https://www.highcharts.com/docs/chart-concepts/understanding-highcharts

[54] S. Arlot & A. Celisse. (2010). *A survey of cross-validation procedures for model selection*. Statistics Surveys. Vol. 4. (2010) 40–79. ISSN: 1935-7516. doi: 10.1214/09-SS054

[55] Han, J., Kamber, M., Pei, J. (2011) *Data mining: concepts and techniques*. Third Edition. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, Elsevier Inc. 2012. (pp 346-389) ISBN 978-0-12-381479-1 http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf

[56] *What is Text Analytics API Version 2.0?*. https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/overview

[57] *Google Cloud*. https://cloud.google.com/docs/overview/

[58] *IBM Watson*. https://console.bluemix.net/developer/watson/documentation