ANÁLISIS DE SENTIMIENTOS EN TWITTER

JOSÉ CARLOS SOBRINO SANDE

MÁSTER UNIVERSITARIO EN INGENIERÍA INFORMÁTICA
INTELIGENCIA ARTIFICIAL

CONSULTOR: SAMIR KANAAN IZQUIERDO
PROFESOR RESPONSABLE DE LA ASIGNATURA: CARLES VENTURA ROYO
FECHA DE ENTREGA: JUNIO DE 2018

FICHA DEL TRABAJO FINAL DE MÁSTER

Título del trabajo:	Análisis de sentimientos en Twitter
Nombre del autor:	José Carlos Sobrino Sande
Nombre del consultor:	Samir Kanaan Izquierdo
Nombre del PRA:	Carles Ventura Royo
Fecha de presentación (mm/aaaa):	06/2018
Titulación o programa:	Máster Universitario en Ingeniería Informática
Área del trabajo final:	Inteligencia Artificial
Idioma del trabajo:	Español
Palabras clave:	Análisis de sentimientos, procesamiento del lenguaje natural, Twitter, aprendizaje supervisado, machine learning.

Resumen del trabajo (máximo 250 palabras): Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.

El Análisis de Sentimientos es un área de investigación enmarcada dentro del campo del Procesamiento del Lenguaje Natural y cuyo objetivo fundamental es el tratamiento computacional de opiniones, sentimientos y subjetividad en textos. En este contexto, una opinión es una valoración positiva o negativa acerca de un producto, servicio, organización, persona o cualquier otro tipo de ente sobre la que se expresa un texto determinado. La llegada de la Web 2.0 y la popularización de redes sociales de microblogging como Twitter han catapultado este campo de investigación de la Inteligencia Artificial hacia las más altas cotas de interés y notoriedad debido a la indiscutible importancia que supone el poder obtener el grado de valoración de miles de personas en cada instante para empresas, organizaciones, gobiernos y consumidores

El objetivo de este TFM es explicar los fundamentos teóricos sobre los que se asienta el análisis de sentimientos, su historia, aplicaciones y su relación con el procesamiento del lenguaje natural. Se ofrecerá una visión del estado del arte mediante un recorrido por los estudios publicados por decenas de autores y veremos los métodos más importantes que existen para desarrollar este tipo de soluciones. Implementaremos un clasificador de sentimientos para los mensajes de Twitter basado en algoritmos de aprendizaje supervisado y se llevará a cabo un estudio comparativo con las técnicas más populares para el análisis de sentimientos a nivel de documento. Finalmente, hablaremos de cómo se presenta el futuro para este tipo de sistemas.

Abstract (in English, 250 words or less):

Sentiment Analysis is a research area inside of Natural Language Processing field whose objective is the computational treatment of opinions, sentiments and subjectivity in texts. In this context, an opinion is a positive or negative evaluation about a product, service, organization, person or any other kind of entity about a specific text is expressed. The arrival of the Web 2.0 and the popularization of microblogging social

networks as Twitter have catapulted this investigation field of Artificial Intelligence to the highest levels of interesting and notoriety due to the indisputable importance that supposes to be able to obtain the degree of valuation of thousands of people in each moment for companies, organizations, governments and consumers.

The aim of this Master's Thesis is to explain the theoretical foundations over sentiment analysis is seated, its history, applications and its relationship with natural language processing. It will be offered a vision of the state of the art through a tour of the published studies by several authors and we will see the most important methods for developing this kind of solutions. It will be implemented a sentiment classifier for Twitter messages based on supervised learning algorithms and we will elaborate a comparative study with the most popular techniques for sentiment analysis at document level. Finally, we will talk about the future of this kind of systems.

Índice de contenidos

Lista de figuras y tablas

Resumen

El Análisis de Sentimientos es un área de investigación enmarcada dentro del campo del Procesamiento del Lenguaje Natural y cuyo objetivo fundamental es el tratamiento computacional de opiniones, sentimientos y subjetividad en textos. En este contexto, una opinión es una valoración positiva o negativa acerca de un producto, servicio, organización, persona o cualquier otro tipo de ente sobre la que se expresa un texto determinado. La llegada de la Web 2.0 y la popularización de redes sociales de microblogging como Twitter han catapultado este campo de investigación de la Inteligencia Artificial hacia las más altas cotas de interés y notoriedad debido a la indiscutible importancia que supone el poder obtener el grado de valoración de miles de personas en cada instante para empresas, organizaciones, gobiernos y consumidores. Esta ingente cantidad de información junto al aumento de la potencia de computación de los ordenadores han hecho posible la aplicación de técnicas de aprendizaje automático para la clasificación de los textos en base a su polaridad sentimental y han abierto una puerta a la que sin duda será una de las áreas de investigación y desarrollo más importantes de los próximos años.

El objetivo de este TFM es explicar los fundamentos teóricos sobre los que se asienta el análisis de sentimientos, su historia, aplicaciones y su relación con el procesamiento del lenguaje natural. Se ofrecerá una visión del estado del arte mediante un recorrido por los estudios publicados por decenas de autores y veremos los métodos más importantes que existen para desarrollar este tipo de soluciones. Implementaremos un clasificador de sentimientos para los mensajes de Twitter basado en algoritmos de aprendizaje supervisado y se llevará a cabo un estudio comparativo con las técnicas más populares para el análisis de sentimientos a nivel de documento. Finalmente, hablaremos de cómo se presenta el futuro para este tipo de sistemas.

Palabras clave

Análisis de sentimientos, procesamiento del lenguaje natural, Twitter, aprendizaje supervisado, machine learning.

1. Introducción

Este primer capítulo estará dedicado a la presentación del TFM. Su propósito es explicar el tema a tratar y situarlo en su correspondiente contexto, indicar cuáles son los objetivos generales y específicos que se persiguen con su realización, la planificación y la metodología seguidas y la estructura que tiene la presente memoria. Además, se incluye un resumen de los productos obtenidos durante su desarrollo y la motivación personal que me ha llevado a tratar el área de investigación del análisis de sentimientos.

1.1 Contexto y justificación del TFM

En todo proceso de toma de decisiones las personas hacemos uso de la opinión de otros individuos a la hora de decantarnos por una opción u otra. El pedir ayuda, consejo u opinión no es más que un recurso que permite al ser humano ampliar su conocimiento sobre un determinado tema con el objetivo de minimizar el riesgo que supone el tomar una mala decisión.

Hace no mucho tiempo, antes de la llegada de Internet y de su actual omnipresencia en todo aspecto de nuestras vidas, las fuentes de opinión principales eran el conocimiento y la experiencia de las personas más cercanas que formaban parte de nuestro círculo de relaciones y amistades. A la hora de comprar un televisor o de alquilar una película en un videoclub, nos decantábamos por una u otra opción en base a la opinión de nuestros amigos, familiares o compañeros de trabajo. El *boca-a-boca* era el sistema que usábamos para transmitir opiniones, juicios y expresar las virtudes o defectos de productos y servicios. En esa época, hace poco más de quince años, también hacíamos uso de otro tipo de recursos durante el proceso de toma de decisiones. Así, las publicaciones en papel especializadas en temas concretos nos asesoraban a la hora de adquirir un ordenador, un coche o irnos de vacaciones a algún país remoto. La sección de cultura y espectáculos de los periódicos tradicionales servía de altavoz para publicitar obras de teatro, exposiciones o cine y nos permitía organizar nuestro ocio de fin de semana.

A pesar de que las fuentes de opinión tradicionales no han desaparecido, es indudable que la implantación de Internet y especialmente la llegada de la llamada Web 2.0[1] han supuesto un profundo cambio en la manera en que las personas buscamos opiniones que nos ayuden durante el proceso de toma de decisiones. Internet se ha convertido en un océano inmenso en donde millones de personas expresan su opinión sobre cualquier tema y en cualquier momento. Es precisamente la bidireccionalidad a la que hace referencia el concepto de Web 2.0. la que ha hecho posible que cualquier persona pueda conocer al instante

[1] Concepto atribuido a Tim O'Reilly y nombrado en la conferencia sobre la Web 2.0 de O'Reilly Media en 2004. El término establece una primera época de Internet en donde la comunicación era unidireccional, ya que los usuarios eran objetos pasivos que recibían la información publicada por los administradores de los sitios web sin posibilidad de interactuar. En la Web 2.0 aparecen los blogs, foros de debate, wikis y las redes sociales dando lugar a una comunicación bidireccional en donde todos los actores de la red aportan contenido por igual.

la opinión de miles de usuarios sobre cualquier cuestión y además contribuir al debate con su propia opinión. Es esta manera de interactuar la que permite crear redes virtuales en donde varias personas relacionadas en base a un tema concreto y en un instante preciso pueden intercambiar su visión particular acerca de dicho tema. Las personas hemos pasado de pedir opinión a nuestro círculo de relaciones más cercano a buscar la opinión de absolutos desconocidos antes de tomar la decisión de comprar un televisor, alquilar una noche de hotel, ir al cine o incluso votar a un determinado partido político. El cambio ha sido tan contundente y efectivo que ya no existe sitio web que no cuente con una sección de opinión asociada a sus propias publicaciones y en donde los usuarios puedan expresar sus juicios acerca del tema a tratar, desde simples noticias de actualidad hasta videos amateur publicados por personas anónimas. La inmediatez de la información asociada al gran número de mensajes hace de Internet y de la Web 2.0 una absoluta revolución en cuestión de opiniones. Tanto es así que no solo los usuarios se han dado cuenta del gran valor que aporta esta información. También las organizaciones, empresas e incluso gobiernos saben de la importancia de estas opiniones y de su valor como herramienta para mejorar sus productos, servicios y su reputación general. Lo que hace años se tenía que obtener en base a largos y costosos procesos de análisis de encuestas ahora es posible conseguirlo sin tener que hacer inversiones en este tipo de estudios y con una mayor rapidez, midiendo en cada momento el sentir de las personas acerca de un determinado tema relacionado con su ámbito de negocio o actuación.

El éxito de este nuevo uso de Internet es indiscutible. Ya no compramos ningún producto en Amazon[2] sin mirar antes las opiniones de otras personas, no alquilamos un hotel en Booking[3] sin analizar las puntuaciones que han dado otros usuarios sobre la comida, la limpieza o la situación física del alojamiento, Menéame[4] nos permite modelar nuestra opinión después de leer las propias de decenas de usuarios sobre las noticias publicadas en el sitio web y nos decantamos por ver una u otra película en base a la puntuación publicada por miles de desconocidos en FilmAffinity[5]. Además, este uso no sólo es válido para la selección de productos y servicios. En el terreno laboral Glassdoor[6] es una fuente magnífica y muy útil para conocer la opinión de otros profesionales sobre una empresa y ayudar en el proceso de decisión antes de trabajar en ella. Otro ejemplo válido es LinkedIn[7], que permite opinar sobre nuestros colegas de profesión mediante recomendaciones de manera que contribuyan a que su reputación aumente y sean más atractivos de cara al mercado laboral. Prácticamente todo está sujeto a la posibilidad de opinar sobre él, mejorando o empeorando la percepción que tienen las personas acerca del elemento en cuestión y ayudando en el proceso de selección.

[2] https://www.amazon.es
[3] https://www.booking.com
[4] https://www.meneame.net
[5] https://www.filmaffinity.com
[6] https://www.glassdoor.com
[7] https://www.linkedin.com

Una fuente de opiniones fundamental dentro de este contexto son las redes sociales. Es innegable la ventaja que tiene para el proceso de toma de decisiones el poder contar en todo momento con los cientos de opiniones que sus usuarios expresan sobre cualquier tema en cada instante. Sólo hay que detenerse por un momento a pensar el valor que tienen los tweets de todas estas personas para empresas, partidos políticos, gobiernos u organizaciones de cualquier tipo. Poder conocer en todo momento qué es lo que piensan los usuarios de alguno de tus productos ofrece una ventaja competitiva impensable de obtener hace tan sólo diez años. La posibilidad de poder mejorar los aspectos menos populares de tus servicios o que un personaje público de cualquier índole pueda conocer si sus palabras o acciones son bien recibidas por sus seguidores es un instrumento de incuestionable utilidad.

Una vez conocida la importancia que tiene toda esta información de opinión y sentimiento vertida en cada instante en Internet, cabe preguntarse cómo poder trabajar con ella sin perderse en el vasto y extenso mar de opiniones y desfallecer en el intento. Procesar tal cantidad de datos requiere de nuevas tecnologías capaces de analizar y clasificar de manera automática el sentimiento y la polaridad descrita por los usuarios en sus opiniones. Es necesario encontrar un método que simplifique el proceso de entendimiento de todos estos textos escritos y haga viable el poder obtener una medida del sentir general que las personas expresan en la red para así poder explotar convenientemente esta valiosa información.

1.2 Objetivos del TFM

El objetivo principal de este TFM será el obtener un método automatizado que permita evaluar la opinión de un texto extraído de la red de microblogging Twitter[8] acerca de un determinado tema y clasificarlo en base a su polaridad de sentimiento. De esta forma, será posible extraer automáticamente la valoración que hacen los usuarios acerca de un tema, persona, producto o servicio concretos, haciendo viable además el análisis masivo de los mensajes publicados en la red social.

Aunque este campo de investigación es muy reciente, su gran popularidad ha propiciado que cada mes se publiquen decenas de estudios de diferentes universidades de todo el mundo; y a pesar de que no existe un método cuyos resultados destaquen con claridad por encima de los otros, una de las más importantes corrientes de investigación basa su trabajo en el uso de algoritmos de aprendizaje automático y en la habilidad que tienen este tipo de sistemas para la clasificación de textos a partir de las palabras y de las relaciones que entre ellas se establecen. Así, el TFM se centrará en este tipo de soluciones y ofrecerá un estudio comparativo en el que se tendrán en cuenta diversos algoritmos de aprendizaje automático supervisado y diferentes métodos de extracción de las características necesarias para su entrenamiento. Una cuestión a destacar es que el estudio hará uso de textos escritos en español en lugar de inglés, ya que la

[8] https://twitter.com

inmensa mayoría de artículos y recursos sobre este tema utilizan dicho idioma. En un apartado posterior, se explicará de qué manera es posible obtener beneficio de este tipo de clasificadores y las ventajas que ofrece el análisis de la polaridad de sentimiento en redes sociales como Twitter.

Por último, es importante mencionar que este TFM se acompañará de una parte teórica en la que se expondrán los conceptos más importantes y fundamentales sobre el análisis de sentimientos, sus problemas, limitaciones, así como un pequeño análisis del estado del arte y cómo se plantea el futuro para este tipo de sistemas.

En base a lo anterior, se pueden establecer los siguientes objetivos generales:

- Obtener un método automatizado que permita evaluar la opinión de un texto extraído de la red de microblogging Twitter acerca de un determinado tema. Este método deberá tener en cuenta las siguientes condiciones:
    - Los textos serán clasificados en base a su polaridad de sentimiento, tomando uno de entre cuatro valores posibles: positivo, negativo, neutro (entendido como textos que incluyen sentimientos positivos y negativos simultáneamente) o con ausencia de sentimiento.
    - Estos mensajes podrán versar sobre cualquier tema ya que el sistema no estará centrado en un dominio concreto.
    - El idioma de los textos será el español.
- Exponer los conceptos más significativos y relevantes acerca del análisis de sentimientos, mostrando sus problemas, limitaciones, la situación del estado del arte y el futuro de este tipo de sistemas.

Conseguir los objetivos generales anteriores dependerá de la obtención de los siguientes objetivos específicos:

- Ubicar el análisis de sentimientos dentro del área de investigación al que pertenece, el procesamiento del lenguaje natural, explicando en qué consiste dicha área, su historia, trabajos y aplicaciones más notables.

- Definir el concepto de análisis de sentimientos, comentando además su historia, autores, trabajos más relevantes y aplicaciones de este campo de investigación.

- Indicar cuáles son y en qué consisten los niveles y tareas necesarias para el proceso de análisis de sentimientos.

- Exponer los métodos existentes más importantes y populares para la clasificación de textos en base a su sentimiento, así como una visión general del estado del arte.

- Explicar cuáles son y en qué consisten los algoritmos de clasificación supervisada más usados para la categorización de textos en base al sentimiento.

- Obtener un corpus de mensajes extraídos de la red social Twitter etiquetados con su sentimiento y que será utilizado para el entrenamiento de los algoritmos de clasificación supervisada anteriores.

- Exponer las técnicas más usadas para preparar los datos de entrenamiento y extraer las características que mejor definan los mensajes con los que se entrenará los clasificadores.

- Ofrecer un estudio práctico comparativo en el que se tengan en cuenta los algoritmos de aprendizaje automático supervisado más importantes y los diferentes métodos de extracción de las características de los mensajes de entrenamiento.

- Proporcionar un conjunto de conclusiones acerca del estudio indicando cuáles son las mejores técnicas encontradas y cómo se ven afectados los algoritmos por la aplicación de los distintos métodos elegidos.

- Explicar con ejemplos reales de qué manera se pueden obtener beneficios de este tipo de sistemas mediante el análisis de las opiniones de los usuarios de Twitter sobre un tema, persona, producto o servicio concreto.

- Ofrecer una serie de conclusiones acerca de la realización del TFM y del análisis de sentimientos y su futuro teniendo en cuenta los conocimientos adquiridos durante su desarrollo.

1.3 Motivación personal

La selección de este tema para la realización del TFM responde a la atracción que siento por el área de Inteligencia Artificial y a la curiosidad y casi fascinación de los algoritmos de aprendizaje automático. El hecho de que existan sistemas que sean capaces de detectar una correlación y conexión oculta entre grupos de datos sin aparente relación abre la posibilidad de hallar respuesta a cuestiones que hasta ahora eran imposibles de abordar debido a que no contábamos con las herramientas adecuadas para detectar patrones en los datos y realizar predicciones en base a los mismos. La colosal cantidad de información que millones de internautas generamos en Internet en cada momento puede servir para detectar y predecir estados de ánimo, enfermedades, conflictos sociales, necesidades, carencias y un sinfín de temas relacionados con el ser humano y su realidad. Tal y como se están desarrollando los acontecimientos, en los próximos años los sistemas de aprendizaje automático formarán parte de nuestra vida y su implantación masiva hará posible el logro de objetivos que hoy en día parecen propios de la ciencia ficción.

1.4 Enfoque y método seguido

El análisis de sentimientos, y de manera especial el que centra su ámbito de actuación en las redes sociales, es un campo de investigación de reciente aparición por lo que, salvo casos muy puntuales, no existe una literatura extensa sobre el tema en los formatos más tradicionales. Así, la principal fuente de información se encuentra en los estudios de investigación publicados por universidades de todo el mundo y que pueden llegar a alcanzar los varios cientos cada año. Junto a los artículos de divulgación sobre el análisis de sentimientos que pueden ser consultados a través de Internet, esta será la principal fuente de conocimiento sobre la que se asentará el presente TFM. Por tanto, la estrategia consiste en recopilar este tipo de publicaciones mediante buscadores académicos especializados como Google Scholar[9] , Springer Link[10], Dialnet[11], BASE[12] o la biblioteca en línea de la UOC. Posteriormente, se deberán detectar aquellos estudios que tengan un mayor número de citas y extraer de ellos los conceptos más importantes y las técnicas más utilizadas para resolver el problema de la clasificación de textos en base al sentimiento.

El siguiente paso será buscar las herramientas que permitan construir sistemas de clasificación automatizados siguiendo las indicaciones de las publicaciones consultadas, así como los diversos ejemplos que ya existen en Internet. Además, se tendrán en cuenta de manera especial los conocimientos adquiridos en la asignatura de Inteligencia Artificial Avanzada y las prácticas realizadas con el lenguaje Python[13] y librerías de aprendizaje automático como Scikit-Learn[14].

Por último, a medida que avance la labor de investigación, se irá completando la memoria del TFM con toda la información obtenida durante su desarrollo y que representará la entrega final de este trabajo.

1.5 Distribución de tareas y planificación del proyecto

Para la elaboración del TFM se disponen de un total de 20 horas por cada semana de trabajo repartidas en 3 horas de lunes a viernes y 5 horas el sábado. La razón de esta distribución horaria responde a la necesidad de compaginar la realización de este TFM con mis obligaciones laborales y personales.

[9] https://scholar.google.es/
[10] https://link.springer.com/
[11] https://dialnet.unirioja.es/
[12] https://www.base-search.net/
[13] https://www.python.org/
[14] http://scikit-learn.org

La siguiente imagen muestra la planificación propuesta para llevar a cabo todas las PACs de las que consta este trabajo y arroja una carga de trabajo de 322 horas. Como se puede observar, las fechas de finalización de cada PAC no coinciden con las indicadas en la planificación publicada en la UOC. Esto se debe a que las tareas de las que consta cada una de ellas imposibilitan que ambas fechas coincidan. No obstante, las fechas límite de cada entrega son respetadas en todo momento. Por último, el listado indica el reparto de las tareas de la memoria en las diferentes entregas a lo largo del semestre.

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	PAC0 - Definición de los contenidos del trabajo	16 horas	mié 21/02/18	lun 26/02/18	
2	PAC1 - Plan de trabajo	34 horas	lun 26/02/18	sáb 10/03/18	1
3	**PAC2 - Desarrollo del trabajo - Fase 1**	**77 horas**	**sáb 10/03/18**	**vie 06/04/18**	
4	Introducción	12 horas	sáb 10/03/18	mié 14/03/18	2
5	Procesamiento del lenguaje natural	15 horas	mié 14/03/18	lun 19/03/18	4
6	Análisis de sentimientos	20 horas	mar 20/03/18	lun 26/03/18	5
7	Análisis y clasificación de documentos	30 horas	mar 27/03/18	vie 06/04/18	6
8	**PAC3 - Desarrollo del trabajo - Fase 2**	**120 horas**	**vie 06/04/18**	**vie 18/05/18**	
9	**Análisis de sentimientos en Twitter**	**100 horas**	**vie 06/04/18**	**vie 11/05/18**	**3**
10	Introducción	6 horas	vie 06/04/18	sáb 07/04/18	3
11	Corpus de entrenamiento	12 horas	sáb 07/04/18	jue 12/04/18	10
12	Algoritmos de aprendizaje supervisado	6 horas	jue 12/04/18	sáb 14/04/18	11
13	Métricas y métodos de evaluación	6 horas	sáb 14/04/18	lun 16/04/18	12
14	Selección del clasificador linea de base	30 horas	mar 17/04/18	vie 27/04/18	13
15	Mejora del clasificador linea de base	20 horas	vie 27/04/18	vie 04/05/18	14
16	Conclusiones	20 horas	vie 04/05/18	vie 11/05/18	15
17	Aplicaciones del análisis de sentimientos en Twitter	20 horas	vie 11/05/18	vie 18/05/18	9
18	**PAC4 - Redacción de la memoria**	**50 horas**	**vie 18/05/18**	**lun 04/06/18**	
19	Conclusiones	30 horas	vie 18/05/18	lun 28/05/18	8
20	Anexos	8 horas	mar 29/05/18	jue 31/05/18	19
21	Bibliografía	3 horas	jue 31/05/18	vie 01/06/18	20
22	Glosario	1 hora	vie 01/06/18	vie 01/06/18	21
23	Redacción memoria	8 horas	sáb 02/06/18	lun 04/06/18	22
24	PAC5a - Elaboración de la presentación	25 horas	mar 05/06/18	mié 13/06/18	18

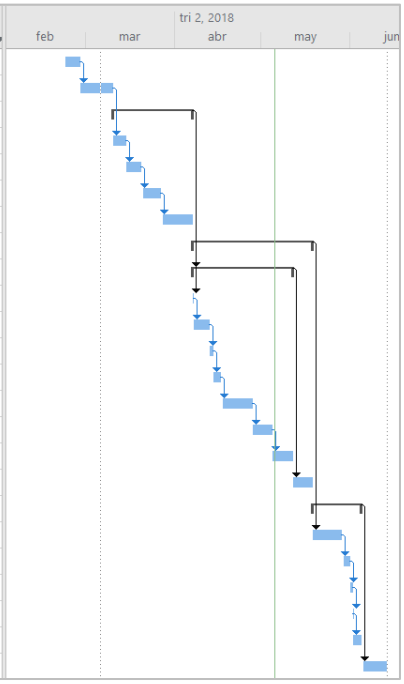

FIGURA 1.5.1 - DIAGRAMA DE GANTT PARA LA PLANIFICACIÓN DEL TFM

1.6 Breve resumen de los productos obtenidos

Los productos presentados en este TFM serán tres: memoria, código fuente y hoja de cálculo con los resultados del estudio comparativo. La memoria es el presente documento y en él se expone todo el conocimiento adquirido durante la realización del TFM. Por otra parte, se entregará un archivo en formato .ZIP con el código fuente desarrollado durante la parte práctica que incluye todos los recursos necesarios para llevar a cabo el estudio y la construcción del modelo de clasificación de sentimientos en Twitter. Por último, una hoja de cálculo en formato Excel mantendrá los resultados obtenidos de la práctica anterior. La estructura del código fuente y de la hoja de cálculo se explicará con mayor detalle en el apartado de anexos de esta memoria.

1.7 Estructura de la memoria

La memoria del TFM estará dividida en diez apartados, más un prefacio inicial a modo de resumen del trabajo. Cada una de estas secciones se expone a continuación:

1. **Introducción:** este será el capítulo de presentación del TFM e incluirá los apartados obligatorios indicados en la plantilla de memoria proporcionada por la UOC. Aquí se tratarán temas como el contexto en el que se encuadra el TFM, sus objetivos generales y específicos, la planificación del proyecto, el método seguido durante su desarrollo o el resumen de los productos obtenidos.

2. **Procesamiento del lenguaje natural:** en este apartado se hará una presentación del área del PLN mediante su definición y un breve recorrido por su historia, nombrando además algunos trabajos y autores relevantes. Así mismo, se expondrá el flujo de trabajo y los componentes que forman parte del desarrollo de cualquier aplicación relacionada con el PLN. Por último, se mostrarán algunas de sus posibles aplicaciones.

3. **Análisis de sentimientos:** como el caso anterior, se hará una presentación del concepto de análisis de sentimientos y un breve repaso a su historia, nombrando algunos autores y trabajos relevantes. Se explicarán los factores a los que se debe el auge actual de este campo de investigación, sus aplicaciones y las dificultades que presenta. En este apartado, también se explicarán los niveles de análisis de sentimientos existentes, las tareas necesarias para llevar a cabo dicho análisis y una definición formal sobre el concepto de Opinión.

4. **Análisis y clasificación de documentos:** se hará una explicación con mayor detalle del análisis de sentimientos a nivel de documento, ya que TFM girará en torno a este nivel de análisis. Además, se presentarán los dos enfoques principales para llevar a cabo la clasificación de textos: mediante aprendizaje supervisado y no supervisado. En ambos casos, se indicará en qué consisten, sus particularidades, ventajas e inconvenientes y algunos de los trabajos más destacados.

5. **Análisis de sentimientos en Twitter:** este apartado de la memoria será sobre todo práctico y en él se construirá un sistema automático que permita clasificar cualquier texto extraído de la red social Twitter en una de las siguientes categorías: positivo, negativo, neutro (entendido como un mensaje que contiene sentimientos positivos y negativos simultáneamente) o con ausencia de sentimiento. El método de clasificación será principalmente supervisado, pero hará uso de alguna técnica perteneciente al grupo de los métodos no supervisados como los diccionarios de palabras etiquetadas por su sentimiento. El idioma de los textos será el español.

6. **Aplicaciones del análisis de sentimientos en Twitter:** en esta sección se explicará de qué manera es posible utilizar el clasificador obtenido en el apartado anterior y en qué escenarios puede ser útil su uso. Se mostrarán diversos ejemplos prácticos de la vida real agrupados por ámbito de actuación y cómo empresas, organizaciones, partidos políticos y usuarios particulares se pueden beneficiar de los sistemas de análisis de sentimientos en Twitter.

7. **Conclusiones y cierre del TFM:** en este apartado se expondrán las conclusiones extraídas a partir de la elaboración del TFM y en relación al análisis de sentimientos, así como a los objetivos, metodología y a la planificación seguida. Cerraremos este trabajo con una serie de reflexiones acerca del futuro de los sistemas de análisis de sentimientos.

8. **Glosario:** listado con la terminología y acrónimos usados en la memoria.

9. **Bibliografía:** relación de referencias bibliográficas consultadas durante el desarrollo del TFM.

10. **Anexos:** aquí se incluirá el código fuente desarrollado durante el estudio comparativo de los algoritmos de clasificación supervisada y una referencia a los resultados obtenidos a partir de dicho estudio.

2. Procesamiento del lenguaje natural

En este apartado se tratará el campo del procesamiento del lenguaje natural. Esta área de estudio es en la que se enmarca el análisis de sentimientos y, como se verá a continuación, aplicaciones muy populares como buscadores web, traductores multidioma o asistentes virtuales dependen de sus métodos, técnicas y avances de investigación. En esta sección de la memoria se incluye un breve recorrido por su historia, algunos de los trabajos más destacados, los niveles comunes de análisis que existen dentro de todas las aplicaciones que procesan de alguna manera lenguajes naturales y los usos más importantes de esta área de investigación.

2.1 Introducción

El procesamiento del lenguaje natural (PLN o NPL por sus siglas del inglés *Natural Language Processing*), es un campo enmarcado dentro del área de la inteligencia artificial, la computación y la lingüística. Su objetivo fundamental es facilitar y hacer eficaz la comunicación entre las personas y los computadores mediante el uso de protocolos como los lenguajes naturales. Estos lenguajes son los usados por las personas para comunicarse entre sí tanto de forma oral como escrita. La comunicación es un elemento esencial para establecer relaciones entre individuos o entidades, sean éstas del mismo tipo o no. Es sencillo deducir que la comunicación entre elementos de la misma naturaleza, como entre personas, máquinas o animales de la misma especie, es más simple, directa y efectiva que cuando se produce entre entidades de diferente origen. Por esta razón y debido a la relación existente entre las personas y los computadores, se hace necesaria la búsqueda y el estudio de protocolos que faciliten la comunicación e interacción entre ambos objetos para así mejorar sus relaciones. Es el área del PLN quien se encarga de esta tarea.

La historia del PLN tiene su origen a mediados del siglo XX con la aparición de una nueva disciplina dentro de las ciencias de la computación. Su objetivo era el desarrollo de sistemas lo suficientemente inteligentes para que la comunicación entre personas y máquinas se hiciese mediante el uso de lenguaje natural. Por aquel entonces, justo después de la Segunda Guerra Mundial, era conocida la importancia de poseer algún sistema que permitiese traducir textos entre diferentes idiomas y de manera automática. Uno de los sistemas que se crearon aquella época fue el conocido como Experimento de Georgetown-IBM, en 1954. Desarrollado conjuntamente por la Universidad de Georgetown e IBM, el experimento consistió en una demostración de traducción automática entre los idiomas inglés y ruso. Contaba con un conjunto de reglas gramaticales y un par de cientos de elementos de vocabulario para llevar a cabo las traducciones. A través de una interfaz rudimentaria, un operador sin conocimientos de lengua rusa, introdujo una serie de frases sobre política, ciencia o matemáticas que fueron procesadas por un ordenador IBM 701, generando una impresión con las frases traducidas al inglés. Aunque es indiscutible el hito que supuso esta prueba, es

necesario decir que las oraciones a traducir fueron especialmente escogidas para la prueba. El sistema no llevaba a cabo ningún tipo de análisis sintáctico que detectase la estructura de las frases y el enfoque usado estaba basado en diccionarios en donde las palabras estaban asociadas a reglas muy específicas. Aun así, los resultados de la prueba generaron unas altas expectativas ya que sus autores afirmaban que el problema de la traducción automática estaría resuelto en pocos años, haciendo que la inversión en este tipo de sistemas se disparase. Pasados más de diez años desde aquella prueba, los investigadores en esta materia reconocían que sus avances eran mucho más lentos de lo esperado, de manera que los fondos invertidos para la investigación se redujeron radicalmente.

Pocos años después del experimento de Georgetown-IBM, en 1957, Noam Chomsky publicó su libro Syntactic Structures (Estructuras Sintácticas, en español) el cual supuso un gran acontecimiento y ejerció una enorme influencia en el campo de la lingüística. Chomsky creía que el cerebro humano contaba con una facilidad innata para usar y entender el lenguaje y esto se debía a que existían un conjunto de reglas y normas universales, comunes a todas las lenguas, que permitían operar con el lenguaje y con las que las personas ya contábamos a la hora de nacer. Estas reglas conforman la llamada “gramática universal” y fueron la base para que Chomsky introdujera la conocida como “gramática generativa transformacional”. La influencia de estas ideas creó una corriente de investigación del PLN en donde sus integrantes afirmaban que el éxito de este tipo de sistemas radicaba en el uso de reglas y patrones estructurales gramaticales formados entre las palabras de los textos. Estas reglas se debían combinar con diccionarios para resolver tareas como traducir textos, buscar información o interactuar con computadores usando un lenguaje similar al usado por humanos. Un ejemplo paradigmático de este enfoque es el programa ELIZA, desarrollado a mediados de la década de 1960 por Joseph Weizenbaum en el Instituto Tecnológico de Massachusetts (MIT). ELIZA era un programa que procesaba lenguaje natural y recreaba la sensación de una conversación coherente con un interlocutor humano. El funcionamiento consistía en extraer palabras clave de la frase introducida por el usuario y contestar con otra relacionada que mantenía en una base de datos interna. La conversación llegaba a ser tan convincente que producía la sensación de estar hablando con otro interlocutor humano. No obstante, tenía problemas cuando las palabras del usuario no figuraban en su base de datos. En estos casos se limitaba a reformular la frase del humano en forma de pregunta. Además, si la conversación se alargaba demasiado, ésta empezaba a ser incoherente. ELIZA puede ser considerado como un precursor primitivo de los asistentes virtuales más actuales como Siri o Cortana.

Las teorías de Chomsky permanecieron vigentes en el PLN durante las tres siguientes décadas, hasta bien entrada la década de 1980. En esta etapa, existía otro grupo de investigadores que confiaban en los modelos probabilísticos basados en datos para hallar soluciones a los problemas planteados dentro del PLN. Este enfoque trataba de buscar relaciones matemáticas entre los componentes de los textos, como letras,

palabras u oraciones y calculaba la probabilidad de que éstas apareciesen en determinados contextos. En base a estas probabilidades se puede llegar a deducir cuál será el siguiente componente lingüístico dentro de una secuencia sin necesidad de recurrir a reglas gramaticales. Uno de los proyectos de mayor éxito que forma parte de esta corriente de investigación es el programa CANDIDE, desarrollado en el año 1991 por investigadores del Thomas J. Watson Center de IBM en Nueva York. Este programa pretendía generar traducciones automáticas sin usar herramientas más allá de sistemas puramente estadísticos. Para ello, se hizo uso del conjunto de actas del Parlamento de Canadá y que constaba por aquel entonces de tres millones de oraciones escritas tanto en inglés como en francés. El proceso consistió en alinear palabras y oraciones de ambos idiomas y calcular la probabilidad de que una palabra de una oración en un idioma se correspondiese con otras palabras en el otro idioma. Los resultados fueron sorprendentemente positivos debido a que la mitad de las frases traducidas se correspondían de manera exacta o tenían un significado similar a las del texto original. Esta investigación es considerada un hito en el campo de la traducción automática y el uso de sistemas estadísticos dentro del PLN y puede ser considerada como la precursora de herramientas más modernas como el servicio de traducción multidioma Google Translate.

Con el paso de los años, el PLN estadístico se ha ido imponiendo con gran éxito dejando atrás las ideas chomskianas y el uso de reglas de transformación gramáticas escritas por humanos. Esto es debido principalmente al aumento de la potencia de cálculo de los ordenadores y a la gran cantidad de información que existe a disposición de los mismos y que es necesaria para construir los sistemas probabilísticos. Así, mediante el uso de los sistemas de aprendizaje automático, son los propios computadores los que aprenden el lenguaje natural, infiriendo las reglas y normas que gobiernan los lenguajes naturales.

2.2 Niveles de análisis para el PLN

Todo sistema de PLN debe llevar a cabo un conjunto de tareas de análisis del lenguaje que faciliten el entendimiento entre el usuario y el propio sistema. Estas tareas constituyen una arquitectura de niveles a través de los cuales y de manera secuencial las oraciones se analizan e interpretan hasta ser comprendidas y asimiladas por el sistema de PLN. A grandes rasgos, existen cuatro componentes principales o niveles de análisis, pero no todos deben ser implementados. Son las funciones a desempeñar por el sistema las que determinan qué niveles de análisis deben ser desarrollados. Estos componentes, ordenados de menor a mayor complejidad, son los siguientes:

- **Nivel de análisis morfológico**: en este componente se examinan las palabras para extraer raíces, rasgos flexivos, sufijos, prefijos y otros elementos. Su objetivo es entender cómo se construyen las palabras a partir de unidades de significado más pequeñas denominadas morfemas.

- **Nivel de análisis sintáctico**: analiza la estructura de las oraciones en base al modelo gramatical empleado con el objetivo de conocer cómo se unen las palabras para crear oraciones.

- **Nivel de análisis semántico**: proporciona sentido a las oraciones y les otorga un significado, resolviendo además las ambigüedades léxicas y estructurales que pudieran aparecer.

- **Nivel de análisis pragmático**: se encarga del análisis de los textos más allá del de una oración aislada, teniendo en consideración aquellas inmediatamente anteriores, la relación existente entre ellas y el contexto en el que se producen.

En el siguiente esquema se puede observar el flujo de trabajo típico de un sistema de PLN y el camino que recorre el texto a través de los diferentes niveles de su arquitectura. Habitualmente, el texto de entrada al sistema es procesado mediante una técnica conocida como *tokenización*. Ésta trata de identificar cuáles son las unidades mínimas de información, conocidas como *tokens,* dividiendo las oraciones en palabras individuales, signos de puntuación y otros elementos. Los *tokens* son tratados por cada uno de los componentes o niveles de la arquitectura hasta que finalmente el texto proporcionado como entrada es entendido por el sistema:

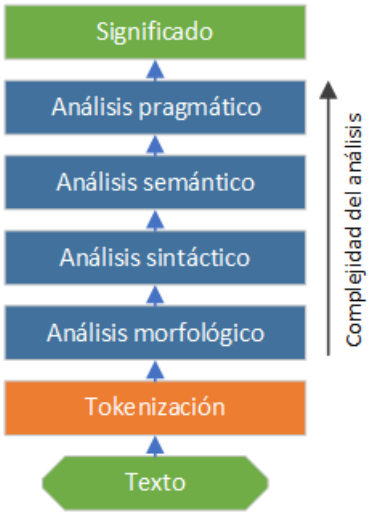

FIGURA 2.2.1 - NIVELES DE ANÁLISIS EN APLICACIONES DE PLN

Como se indicó anteriormente, dependiendo del sistema de PLN a desarrollar será necesaria la implementación de diferentes componentes de análisis. Por ejemplo, para un sistema de traducción automática, los niveles de análisis morfológico y sintáctico son suficientes. En cambio, un asistente virtual necesita además entender el significado de las órdenes del usuario, por lo que será necesario contar con componentes de análisis semántico y pragmático también.

2.3 Aplicaciones del PLN

Es fácil deducir que el PLN es una disciplina que cuenta con un alto potencial y múltiples posibilidades prácticas, tantas como los lenguajes naturales poseen. Su cometido es mejorar y hacer eficaz la comunicación entre personas y computadores. Por esta razón, cualquier área asociada al lenguaje y a las relaciones entre humanos y máquinas se puede ver afectada positivamente por el PLN. Aunque sus aplicaciones son innumerables y el único límite existente es la propia imaginación, algunas de las más populares e importantes son las siguientes:

- **Recuperación de la información**: el objetivo de este tipo de sistemas en la búsqueda y obtención de grupos de documentos electrónicos a partir de un conjunto de palabras clave proporcionadas por el usuario. Los documentos devueltos normalmente se ordenan en base a algún tipo de atributo que mide su relevancia dentro del resultado global. Estos sistemas son en los que basan su funcionamiento los buscadores de contenidos de Internet y representan la primera aplicación implantada masivamente dentro del mundo de las Tecnologías de la Información. Algunos ejemplos populares podrían ser el servicio Google Search o Microsoft Bing.

- **Traducción automática de textos**: una de las aplicaciones paradigmáticas de los sistemas de PLN es la traducción automática entre múltiples lenguajes naturales. Esta labor es tácitamente imposible para un humano debido a la dificultad que existe a la hora de encontrar personas que conozcan decenas de lenguas distintas o la sola combinación de lenguas sobre las que queremos hacer la traducción (por ejemplo, griego y camboyano simultáneamente). Los sistemas actuales de traducción automática utilizan un enfoque basado en mediciones estadísticas y relaciones entre textos a partir de un entrenamiento previo con cientos o miles de textos. Aunque las traducciones no siempre son perfectas y no pueden sustituir a humanos en textos complejos en los que se requiera alta fiabilidad, sí son aceptables para tareas como traducir un mensaje de una red social, una página web o una reseña en una página de alquiler de coches. Uno de los ejemplos más significativos de este tipo de sistemas es el traductor Google Translate.

- **Reconocimiento del habla**: este tipo de sistemas permiten a las personas interactuar con los ordenadores u otros dispositivos electrónicos como teléfonos inteligentes o automóviles mediante el uso de un lenguaje natural y por medio de la voz. En los últimos tiempos se han hecho muy populares los llamados “asistentes virtuales” como Cortana, Siri o Google Assistant. Éstos pueden realizar acciones como enviar un correo electrónico, gestionar el calendario de citas o incluso realizar una compra usando para ello sólo comandos de voz. Otras aplicaciones que hacen uso de estos sistemas de PLN son los servicios telefónicos de atención al cliente. Estos servicios permiten realizar

diferentes gestiones sin necesidad de un interlocutor humano, siendo muy habituales en actividades de banca y telecomunicaciones. Virtualmente todo dispositivo electrónico podría en el futuro poseer un sistema de reconocimiento del habla de manera que sus funciones pudiesen ser controladas mediante comandos de voz. Por tanto, el campo del reconocimiento de la voz humana se postula como uno de los más importantes debido a la popularización de este tipo de sistemas.

- **Extracción de la información**: este tipo de tareas consiste en analizar textos o mensajes con el objetivo de capturar y extraer automáticamente aquella información considerada de interés. Una aplicación habitual es el escaneo de documentos escritos en algún lenguaje natural y para después volcar la información extraída a una base de datos de manera automática. Estos documentos pueden ser anuncios por palabras, artículos de prensa, informes de carácter científico, etc., y los datos a extraer, nombres de personas, organizaciones, teléfonos, fechas, valores monetarios u otros. El proceso de extracción de la información es básico para poder clasificar documentos, resumirlos o relacionarlos entre sí.

- **Análisis de sentimientos**: como se verá a lo largo de este TFM, el análisis de sentimientos ofrece la posibilidad de conocer automáticamente cuál es la opinión que una persona tiene sobre un determinado tema a partir de las ideas expresadas en un texto. Este sentimiento u opinión es una valoración cualitativa o cuantitativa acerca de un producto, servicio, persona o cualquier otro tipo de entidad. El poder extraer de manera automática esta información permite la creación de poderosas herramientas que facilitarán conocer cuál es el sentir de las personas en cada momento en relación a diferentes objetos de estudio.

3. Análisis de sentimientos

En este apartado se expondrá el concepto de análisis de sentimientos mediante su definición y un breve repaso por su historia, nombrando también algunos de los autores y trabajos más relevantes. Como se verá a continuación, el análisis de sentimientos es un área de investigación en pleno auge cuyo esplendor se debe a una serie de factores muy determinados. En este capítulo también se indicarán las aplicaciones más importantes que tiene este campo en la vida real, los diferentes niveles de análisis de sentimientos que se pueden ejecutar sobre textos escritos, las tareas necesarias para su realización y una definición formal sobre el concepto de Opinión. Para finalizar esta sección se explicarán cuáles son las dificultades más comunes a las que se enfrenta el análisis de sentimientos.

3.1 Introducción

El Análisis de Sentimientos (AS o SA por sus siglas del inglés *Sentiment Analysis*) es un campo de investigación dentro del PLN que trata de extraer de manera automática y mediante técnicas computacionales información subjetiva expresada en el texto de un documento dado y acerca de un determinado tema. De esta forma, mediante el análisis de sentimientos podremos saber si un texto presenta connotaciones positivas o negativas. Una definición ampliamente extendida de este concepto es la ofrecida por los investigadores Pang y Lee en (Pang & Lee, 2008) y que define el análisis de sentimientos como:

"Tratamiento computacional de opiniones, sentimientos y subjetividad en textos."

Esta definición es la más aceptada por la comunidad de investigadores, pero debido a su generalidad otros autores como Cambria y Hussain (Cambria & Hussain, 2012) han definido el análisis de sentimientos de la siguiente manera:

"Conjunto de técnicas computacionales para la extracción, clasificación, comprensión y evaluación de opiniones expresadas en fuentes publicadas en Internet, comentarios en portales web y en otros contenidos generados por usuarios."

Se puede observar que la segunda definición es mucho más concreta que la primera y sólo hace referencia a las opiniones, dejando fuera del alcance de estudio a los sentimientos y a la subjetividad. Es posible que dejar fuera a los sentimientos sea un error puesto que muchas veces las opiniones están fundamentadas y emanan de los sentimientos de quien las expresa, pero como indica E. Martínez en (Martínez Cámara, 2016), sí es un acierto no hacer referencia a la subjetividad ya que las opiniones se pueden encontrar en oraciones subjetivas y también objetivas. En cualquier caso, ambas definiciones son útiles y válidas para comprender en qué consiste el análisis de sentimientos.

Aunque la historia del análisis de sentimientos pertenece sin ningún género duda al siglo XXI, existen algunos trabajos desarrollados mucho antes considerados como precursores de este campo de investigación. Uno de ellos es (Carbonell, 1979) en donde se propone un modelo computacional que permite representar el pensamiento subjetivo de las personas, tratando de entender su ideología y su personalidad a través de la subjetividad que contienen sus textos escritos. Pocos años después, en (Wilks & Bien, 1984), se presenta un estudio sobre las creencias que tiene un sujeto sobre otro en base al conocimiento que tienen de ambos por separado. Estos dos estudios, aunque relacionados, no avanzaron hacia lo que hoy se conoce como análisis de sentimientos, sino hacia otros campos de investigación como la interpretación de metáforas, los puntos de vista, el afecto y otras áreas relacionadas.

La verdadera explosión de trabajos de investigación del análisis de sentimientos se produce a partir de 2001 y su número ha ido incrementándose de manera exponencial con el paso de los años. En (Pang & Lee, 2008) se atribuye este progresivo interés a tres factores:

- La popularización de los métodos de aprendizaje automático y su uso dentro de las diferentes áreas del PLN.

- La disponibilidad de datos con los que entrenar a los sistemas de aprendizaje automáticos provenientes principalmente de Internet y de su capacidad para generar ingentes cantidades de información, en especial a partir de la aparición de la llamada Web 2.0.

- El creciente interés por explotar esta información por parte de organizaciones y empresas debido a las posibilidades que ofrece el poder obtener automáticamente una valoración por parte de las personas acerca de productos, servicios o personas concretas.

Aunque la lista de trabajos de investigación es interminable, es importante destacar algunos considerados como los verdaderos creadores de los métodos que más se utilizan a la hora de capturar el sentimiento global de textos y documentos. (Pang et al., 2002) es el primer trabajo conocido que hace uso de algoritmos de aprendizaje automático para la clasificación de textos; en este caso, críticas de películas extraídas de un sitio Web especializado en esta temática. Otro precursor es P.D. Turney que en su estudio (Turney, 2002) muestra un sistema que es capaz de clasificar opiniones de usuarios sobre diversos productos y servicios como automóviles, viajes o películas, mediante un análisis gramatical de las oraciones y una serie de consultas en el motor de búsquedas AltaVista. Cabe también destacar (Taboada et al., 2011) que hace uso de diccionarios de palabras etiquetadas con su polaridad para la clasificación de textos y (Pak, A., & Paroubek, P., 2010) que usa algoritmos de aprendizaje automático para clasificar mensajes de Twitter en base a su sentimiento y además presenta un sistema para crear automáticamente un corpus con el que

entrenar dichos algoritmos. El número de trabajos es tan desmesuradamente grande que existen estudios cuyo objetivo principal es analizar el estado del arte mostrando las publicaciones más importantes sobre este tema. Dos recursos notables sobre este tipo de estudios podrían ser (Medhat, W. et al., 2014) y (Kumar, A., & Teeja, M. S, 2012).

Además de estos trabajos de investigación, también es necesario destacar el estudio de divulgación de Pang y Lee (Pang & Lee, 2008) en donde se muestran diversas técnicas y procedimientos para la construcción de sistemas de análisis de sentimientos, así como el libro (Liu, 2012) en donde se desarrollan en detalle todos los conceptos relacionados con el análisis de sentimientos y cuyos planteamientos siguen vigentes a fecha de hoy.

La popularidad de este tipo de estudios y su vertiginosa proliferación ha dado lugar al uso de múltiples y variados nombres para hacer referencia a este campo de investigación: *opinion mining, sentiment analysis, review mining, subjectivity analysis, affective computing* y algunos otros. A pesar de esta explosión terminológica, existe un consenso internacional a la hora de hacer referencia a esta área en relación a los términos a utilizar y son *sentiment analysis* y *opinion mining*. Así, en español, las traducciones análisis de sentimientos y minería de opinión son también ampliamente aceptadas y serán las utilizadas a lo largo de este TFM.

3.2 Aplicaciones del análisis de sentimientos

Los beneficios del análisis de sentimientos son múltiples y sustanciales. Tanto es así, que empresas, organizaciones y gobiernos de todo el mundo son los principales interesados en el avance de este campo de investigación. Poder saber qué es lo que piensa la gente sobre sus productos, medidas y políticas en cada momento es una herramienta muy valiosa y bien utilizada puede ofrecer ventajas competitivas impensables de conseguir hasta hace pocos años. De la misma forma, la monitorización de las redes sociales y la extracción del sentir global sobre determinados temas, puede ayudar a detectar la gestación de determinados acontecimientos sociales como huelgas, sediciones, revueltas, etc.

En concreto, algunas de las aplicaciones del análisis de sentimientos podrían ser las siguientes:

- **Valoración de opinión de productos y servicios**: probablemente esta sea la aplicación más práctica y directa del análisis de sentimientos. Mediante esta técnica es posible que las empresas puedan conocer la opinión de los usuarios acerca de sus productos sin necesidad de llevar a cabo estudios tradicionales como encuestas de satisfacción. Así, mediante las opiniones vertidas en foros, blogs y especialmente redes sociales, será posible conocer a los usuarios les gusta o no un determinado producto. De esta forma, las empresas pueden conocer en cualquier momento si sus productos son

del agrado de los usuarios y, en caso negativo, poder replantear estrategias en el menor tiempo posible otorgando así ventajas competitivas.

- **Corrección de opinión**: es habitual que los usuarios expresen su opinión en sitios de compras online indicando, además de una reseña, una puntuación en una escala de, por ejemplo, entre 1 y 5 puntos. Puede ocurrir que, por error, el usuario no indique correctamente dicha puntuación. Así, un sistema de análisis de sentimientos podría analizar las palabras del usuario y corregir automáticamente dicha puntuación.

- **Mejora de los sistemas de recomendación de productos**: en base a las opiniones de los usuarios, una tienda online podrá priorizar los productos que ofrece en base a dichas opiniones o no recomendar aquellos cuya opinión general sea negativa.

- **Posicionamiento de publicidad on-line**: los anunciantes de determinados productos podrían requerir que sus anuncios fuesen publicados sólo en sitios web en donde se expresen conceptos positivos, huyendo de aquellas páginas en donde los textos expresen sentimientos negativos.

- **Reputación política**: el análisis de sentimientos demuestra un fuerte potencial para conocer la opinión de la gente sobre un determinado partido político, un candidato o la acogida de determinadas políticas implantadas por el Gobierno, así como sus variaciones a lo largo del tiempo.

- **Análisis del mercado financiero**: a partir de la información contenida en páginas web, foros y redes sociales sobre una empresa concreta, es posible prever cuál será su evolución en el mercado financiero a partir del valor agregado de la polaridad de todas las opiniones encontradas.

3.3 Definición formal de Opinión

Con el objetivo de exponer los elementos de estudio del análisis de sentimientos, analizaremos el siguiente ejemplo ficticio que podría haber sido extraído de cualquier web de compra de dispositivos electrónicos. Esta definición de Opinión aparece descrita en (Liu, 2012):

Susana Martínez escribió:
(17 de marzo de 2018)

Hace tiempo que estaba pensando en cambiar de televisor [1]. Me decidí por este televisor Samsung debido a que es muy elegante [2]. La calidad de la imagen es increíblemente buena [3]. Además, el sonido es magnífico [4]. Un problema es que el sistema de navegación es un poco lento [5] pero viene traducido a diferentes idiomas, entre ellos el español [6]. Mis hijos están encantados con él [7], pero mi marido piensa que es demasiado grande para nuestro salón [8].

En el anterior ejemplo, lo primero que se observa es que no todas las sentencias que forman parte del texto expresan sentimientos. Así, la oración marcada con [1] simplemente indica que la persona quería cambiar de televisor, pero no expresa ningún tipo de opinión. Otra cuestión relevante es que algunas frases valoran el televisor Samsung en sí mismo [2][7] pero otras, [3][4][5][6][8], lo hacen de determinados componentes como la calidad de la imagen, el sonido, el sistema de navegación y el tamaño. Es importante observar que estas opiniones pueden ser tanto positivas como negativas. También es llamativo lo que ocurre con la frase [6] en la que una afirmación objetiva, como es el hecho de que el televisor posee un sistema de navegación traducido a varios idiomas, esconde una valoración claramente positiva, demostrando así que no sólo las frases subjetivas transmiten sentimientos. Para finalizar, se puede observar que la opinión no sólo procede de la persona que escribe el comentario si no de sus hijos [7] y de su marido [8].

A partir del ejemplo anterior es posible deducir cuáles serían los componentes que formarían parte de la definición formal de una opinión. De esta forma, una opinión se define como una cuádrupla *(o, s, h, t)* en donde *o* es el objeto de opinión, *s* el sentimiento, positivo o negativo, *h* es la persona que expresa dicha opinión y *t* el momento en el que lo hace. Esta definición, aunque concisa y válida, no permite representar determinados elementos que aparecen en el ejemplo, tales como el sonido, la imagen o el sistema de navegación.

El objeto global de valoración, el televisor, se divide a su vez en otra serie de componentes sobre los que también existen opiniones y éstas no tienen por qué coincidir con la persona principal que las emite ni con el sentimiento global acerca de dicho objeto de opinión. De esta forma, se deduce que un objeto de opinión puede estar formado por componentes más pequeños por lo que es necesario redefinir la expresión formal de opinión inicial, sustituyendo el objeto de la cuádrupla por otro elemento denominado entidad. Así, una entidad *e* es un producto, servicio, tema, organización, persona o asunto descrita por el par *(T, W)* en donde *T* es una jerarquía de partes y subpartes en las que se compone la entidad y *W* es un conjunto de atributos de la propia entidad. Además, cada parte o subparte tiene sus propios atributos. Así, en nuestro caso, el televisor estará formado por una jerarquía de componentes como la pantalla, los altavoces o el mando a distancia. Todos estos componentes forman una jerarquía dependiente del nodo raíz o entidad, que en este caso es el televisor. Cada uno de los elementos de esta estructura tiene asociado un conjunto de atributos, como el brillo, el contraste o la resolución en el caso de la pantalla, y pueden tener una valoración distinta y no emitida por la misma persona en todos los casos.

Teniendo en cuenta lo anterior, podemos redefinir la expresión formal de opinión como una quíntupla $(e_i, a_{ij}, p_{ijkl}, h_K, t_l)$ donde e_i se refiere a la entidad, a_{ij} es el aspecto de la entidad e_i sobre el que se emite opinión, p_{ijkl} es la opinión en sí misma, h_K la persona que la realiza y t_l el momento en el que la hace. Los valores que puede tomar p_{ijkl} son diversos pudiendo tratarse de una valoración numérica dentro de un rango

(por ejemplo, entre 1 y 5) o bien un valor concreto a elegir entre positivo, negativo y neutro. Además, como se verá más adelante, existirá un caso especial en donde la opinión puede estar referida a la propia entidad, es decir, en donde e_i y a_{ij} sean iguales. En estos casos a_{ij} se representa por la palabra GENERAL.

Aunque la definición anterior es suficiente para la mayoría de las aplicaciones, no está exenta de algunas limitaciones que impiden tratar determinados tipos de opinión mediante esta representación formal. Por ejemplo, las opiniones que relacionan dos o más entidades resultan en una pérdida de contexto. En el caso anterior, el marido de Susana Martínez dice que el tamaño del televisor es demasiado grande, pero en relación al salón. Por tanto, con el método presentado, el aspecto "tamaño" sería valorado como negativo, aunque en realidad éste es negativo con respecto al salón. Aquí, el salón es el contexto y no puede ser representado mediante la quíntupla anterior.

Otro posible problema es que la jerarquía de entidades y aspectos puede llegar a ser muy compleja y difícil de manejar. Esto ocurre cuando existen opiniones sobre aspectos pertenecientes a su vez a otros aspectos de la entidad raíz. Por ejemplo, en la opinión del televisor se valora de manera negativa la velocidad del sistema de navegación. La velocidad es un aspecto perteneciente al sistema de navegación, y éste último es su vez un aspecto del televisor. Para poder representar esta jerarquía, tendrán que existir dos entidades distintas, el televisor y el sistema de navegación, cada una de ellas con sus aspectos y valoraciones propios lo que puede acabar creando jerarquías demasiado complejas y difíciles de administrar.

Por último, la definición formal sólo puede ser usada en las opiniones conocidas como regulares, que son las que valoran un aspecto concreto de una única entidad. En cambio, aquellas opiniones que establecen una relación entre dos o más entidades y en donde su emisor expresa su preferencia en base a algún aspecto (por ejemplo, la calidad de imagen del televisor Samsung es mejor que la del televisor Sony) no pueden ser representadas mediante la quíntupla anterior. Este tipo de opiniones, conocidas como comparativas, necesitan otro tipo de enfoque a la hora de ser analizadas.

3.4 Tareas del análisis de sentimientos

A partir de la definición formal de opinión presentada en el apartado anterior, se pueden enumerar las tareas que son necesarias para poblar la quíntupla de opinión y así poder llevar a cabo el trabajo de análisis de sentimientos. Estas tareas son las siguientes:

- **Tarea 1 - Extraer y categorizar entidades**: dado un documento, en primer lugar, se deben encontrar e identificar todas las entidades que contiene y agruparlas en base a su significado común. Cada uno de estos grupos representará a única entidad e_i.

- **Tarea 2 - Extraer y categorizar aspectos**: en esta tarea se buscarán y capturarán los aspectos del documento teniendo en cuenta que pueden existir distintas formas de expresarlos. Una vez localizados, se deben extraer y agrupar para, a continuación, asociar cada grupo con su entidad correspondiente. Cada uno de estos grupos de la entidad e_i representa a un único aspecto $a_{ij.}$

- **Tarea 3 - Extraer y categorizar a los autores de la opinión**: en este caso, la extracción se hará para cada autor o autores de las opiniones vertidas en el documento, tiendo en cuenta de nuevo que un mismo autor h_K puede ser representado en el texto de diferentes maneras.

- **Tarea 4 - Extraer el momento temporal**: se trata de detectar el momento t_l en el que la opinión fue emitida.

- **Tarea 5 - Clasificar la polaridad a nivel de aspecto**: para cada par de entidad e_i y aspecto a_{ij} se debe determinar la valoración p_{ijkl} emitida por el autor de la opinión.

- **Tarea 6 - Generar la quíntupla de opinión**: con todos los elementos identificados en los pasos anteriores, se crearán las quíntuplas que representen las distintas opiniones expresadas por sus autores.

Retomando el ejemplo del televisor, se pueden generar las quíntuplas de opinión siguiendo los pasos antes descritos. De esta forma:

- **Tarea 1 - Extraer y categorizar entidades**: estas son el televisor Samsung y el sistema de navegación del televisor.

- **Tarea 2 - Extraer y categorizar aspectos**: la calidad de imagen, el sonido y su tamaño son aspectos del televisor. Por otra parte, la velocidad y los idiomas serán aspectos del sistema de navegación.

- **Tarea 3 - Extraer y categorizar a los autores de la opinión**: Susana Martínez, sus hijos y su marido.

- **Tarea 4 - Extraer el momento temporal**: como no se especifica otra cosa, todas las opiniones fueron emitidas el 17 de marzo de 2018.

- **Tarea 5 - Clasificar la polaridad a nivel de aspecto**: la calidad de imagen, el sonido y los idiomas del sistema de navegación tienen valoraciones positivas. En cambio, el tamaño del televisor y la velocidad del sistema de navegación obtienen valoraciones negativas. Por último, existe una valoración positiva del televisor en general por parte de los hijos.

- **Tarea 6 - Generar la quíntupla de opinión: en nuestro caso, serán seis quíntuplas distintas**:
  - (Televisor Samsung, calidad imagen, positivo, Susana Martínez, 17 de marzo de 2018)
  - (Televisor Samsung, sonido, positivo, Susana Martínez, 17 de marzo de 2018)
  - (Sistema de navegación, velocidad, negativo, Susana Martínez, 17 de marzo de 2018)
  - (Sistema de navegación, idiomas, positivo, Susana Martínez, 17 de marzo de 2018)
  - (Televisor Samsung, GENERAL, positivo, Hijos de Susana Martínez, 17 de marzo de 2018)
  - (Televisor Samsung, tamaño, negativo, Marido de Susana Martínez, 17 de marzo de 2018)

3.5 Niveles de análisis de sentimientos

El análisis de sentimientos de un documento se puede llevar a cabo a tres niveles distintos en base a la granularidad, profundidad y detalle requeridos. Estos niveles son:

- **Análisis a nivel de documento**: en este nivel se analiza el sentimiento global de un documento como un todo indivisible, clasificándolo como positivo, negativo o neutro o usando otro sistema de calificación. En estos casos, se asume que dicho documento expresa una valoración sobre una única entidad (por ejemplo, un servicio o producto) por lo que no es aplicable en aquellos que hablen sobre varias entidades simultáneamente.

- **Análisis a nivel de oración**: en este caso, se divide el documento en oraciones individuales para extraer posteriormente la opinión que contiene cada una de ellas. La opinión de cada oración puede ser, de nuevo, positiva, negativa o neutra o bien tomar un valor en base a cualquier otro tipo de medida.

- **Análisis a nivel de aspecto y entidad**: este es el nivel de análisis con mayor detalle posible, en donde una entidad está formada por distintos elementos o aspectos y sobre cada uno de ellos se expresa una opinión cuya polaridad puede ser distinta en cada caso. Este nivel es el que se corresponde con la quíntupla presentada en el apartado anterior y el que mayor desafío presenta en la actualidad para los investigadores de la materia.

3.6 Dificultades para el análisis de sentimientos

Además de la dificultad que implica ejecutar correcta y eficazmente las tareas necesarias para el análisis de sentimientos, esta área de investigación presenta una serie de problemas y obstáculos que los investigadores de la materia deben solucionar para obtener los mejores resultados. Algunos de estos problemas son herencia del PLN, pero otros pertenecen de manera exclusiva al campo del análisis de sentimientos.

Uno de estas complicaciones es que el sentimiento de las palabras a menudo depende del contexto en el que están ubicadas. Por ejemplo, la palabra “cabeza” tiene una connotación positiva dentro de la frase “tener la cabeza bien amueblada”. En cambio, en la frase “perder la cabeza” su connotación es sin duda la contraria. Otro ejemplo podría ser el verbo “morir”, que suele denotar sentimientos negativos. No obstante, no es así en la frase “morir de la risa”.

Una cuestión similar al problema del contexto es que el dominio afecta directamente a la polaridad de las palabras. Así, una misma palabra puede expresar sentimientos positivos en un contexto y negativos en otro. Por ejemplo, si hablamos de teléfonos móviles, decir que la batería es “interminable” se considera algo positivo. En cambio, si es sobre películas, una película “interminable” suele denotar sentimientos negativos.

Otra dificultad que necesita de algún sistema para ser resuelta es la detención y el tratamiento de la negación en las oraciones. Palabras como “no” suelen invertir la polaridad de aquellas a quienes acompañan. Por ejemplo, en “la película no es buena”, el adjetivo “buena” es positivo pero su polaridad se invierte debido a la aparición de la partícula “no”. El manejo de las negaciones es una tarea compleja ya que no siempre una negación invierte el sentimiento de las palabras a las que acompañan. Un ejemplo lo tenemos en la frase “No hay duda de que es el mejor” en donde “no” no afecta a la polaridad del adverbio “mejor”.

Detectar las sutilezas de figuras como el sarcasmo y la ironía a veces no es ni siquiera sencillo para las personas y resolverlas depende en muchos casos del contexto en el que se encuadran. Por ejemplo, en el mensaje “¡Este televisor es genial! ¡Sólo me ha durado dos meses!” no se están alabando las bondades del televisor sino emitiendo una opinión muy negativa sobre su calidad. Las expresiones coloquiales también suponen una dificultad para extraer el sentimiento debido a que no poseen ninguna palabra con algún tipo de polaridad sentimental. Este es el caso de la expresión “costar un ojo de la cara”.

Tratar correctamente los intensificadores es fundamental para realizar un buen análisis del sentimiento de un texto ya que estos elementos varían el grado o la intensidad de la opinión. Por ejemplo, en la oración “El coche es muy veloz”, el adverbio “muy” amplifica la connotación de la palabra “veloz”. En cambio, en “el alquiler es escasamente asequible”, la palabra “escasamente” hace lo contrario con el adjetivo “asequible”.

Por último, hay que destacar que la ambigüedad es uno de los elementos lingüísticos más complicados de resolver dentro del PLN. Mientras las personas somos capaces de deducir los significados de oraciones ambiguas por medio del contexto y en base a nuestras experiencias, los computadores no cuentan con recursos para ello. Un ejemplo de ambigüedad podría ser “Juan vio un niño con un telescopio en la ventana” en donde no se sabe si era el niño quien tenía el telescopio o si fue Juan quien lo usó para ver al niño. Este tipo de escenarios complican mucho el análisis de sentimientos, especialmente a niveles en donde se requiere una gran granularidad.

4. Análisis y clasificación de documentos

En este apartado se explicará con mayor detalle en qué consiste y cuáles son las particularidades del análisis de sentimientos a nivel de documento, ya que el estudio práctico de este TFM estará centrado en dicho nivel de análisis. Además, se presentarán los dos grandes grupos en los que se encuadran la mayor parte de los métodos conocidos para clasificar los textos en base a su polaridad: mediante aprendizaje supervisado y no supervisado. En ambos casos se indicará en qué consisten, sus características más relevantes, sus ventajas e inconvenientes y algunos de los trabajos más destacados.

4.1 Introducción

En la sección anterior se mostró que existen tres posibles niveles de análisis de sentimientos a los que un texto puede ser sometido: análisis a nivel de documento, a nivel de oración o a nivel de aspecto y entidad. El análisis a nivel de documento es probablemente al que se dedica un mayor porcentaje de los estudios que cada año se publican sobre esta área de investigación y su objetivo principal consiste en clasificar un documento en base al sentimiento que en él se expresa. Esta tarea también se conoce como clasificación de sentimientos en documentos. Los documentos son considerados las unidades básicas de información y éstos pueden ser opiniones en blogs, en tiendas online, sitios web especializados o mensajes en redes sociales. La opinión generalmente toma un valor de entre tres posibles: sentimiento positivo, negativo o neutro, aunque, como se verá a continuación, también existen otras escalas y éstas además pueden ser numéricas, continuas o discretas.

Tomando como punto de partida la definición formal de opinión descrita en anteriores apartados, la clasificación de sentimiento a nivel de documento se puede representar con la siguiente quíntupla:

(__, GENERAL, s, __, __)

Así, dado un documento *d,* este nivel de análisis trata de determinar el sentimiento *s* del aspecto GENERAL de la entidad *e*. La entidad *e*, el autor de la opinión *h* y el momento en el que fue emitida *t* son conocidos o irrelevantes. El valor *s* puede ser una categoría de entre varias disponibles (por ejemplo, positivo, negativo o neutro) o bien un valor numérico (por ejemplo, un valor entre 1 y 5). Al primer caso se le conoce como clasificación mientras que el segundo se denomina regresión.

Para poder asegurar que este proceso de clasificación pueda ser llevado a la práctica, es necesario asumir que el documento que se quiere clasificar expresa una opinión sobre una única entidad *e* y dicha opinión pertenece a una única persona *h*. Por tanto, si en un texto se emiten opiniones sobre distintas entidades sus valoraciones o sentimientos podrían ser distintos entre sí lo que impediría clasificar el

documento global en una única categoría. Lo mismo ocurre si en ese texto son varias las personas que manifiestan su opinión. En este caso, es posible que sus opiniones sean diferentes por lo que el proceso de clasificación fallaría por la misma razón que el anterior ejemplo. En cualquier caso, este tipo de análisis de sentimiento es apropiado para reseñas de productos y servicios y puede ser también aplicado a mensajes de redes sociales. En todos los casos y en general, el texto está escrito por una única persona y, habitualmente, trata sobre un solo tema o entidad.

4.2 Métodos para la clasificación de documentos

Para llevar a cabo la clasificación de un documento en base a su sentimiento existen diversos métodos y técnicas que se van refinando y mejorando a medida que avanzan las investigaciones sobre esta materia y aparecen en escena nuevos estudios y trabajos. A pesar de la multitud de artículos y publicaciones presentados cada año, cuestión que pone de manifiesto una corriente de investigación en pleno apogeo y crecimiento, no parece existir un consenso claro sobre qué técnicas se deben usar para obtener los mejores resultados en el proceso de clasificación de textos. Y es debido a este gran número de publicaciones y a un campo de investigación sumido en un proceso de fuerte expansión por lo que no es sencillo fijar una división clara de los métodos existentes en la actualidad. Aun así, varios autores como (Liu, 2012) o (Biagioni, 2016) establecen dos grandes grupos, métodos supervisados y no supervisados y éstos últimos a su vez basados en diccionarios o en relaciones lingüísticas.

4.2.1 Clasificación mediante aprendizaje supervisado

La clasificación mediante técnicas de aprendizaje supervisado está basada en el uso de algoritmos de aprendizaje automático, conocidos también como *machine learning.* Su tipificación de "supervisados" se debe a que estos métodos necesitan de un grupo de documentos de ejemplo previamente etiquetados para generar un modelo que será usado posteriormente para clasificar nuevos textos y que en este contexto es conocido como "corpus". Su funcionamiento se basa en la relación matemática creada entre los elementos de ejemplo durante un proceso conocido como entrenamiento y en donde se genera un modelo estadístico que agrupa dichos elementos en tantos conjuntos como diferentes etiquetas o clases existan en el grupo de documentos de entrenamiento. Posteriormente, el modelo generado se utiliza con un ejemplo no etiquetado para determinar de cuál de los grupos existentes formaría parte, realizando así una predicción en base a los ejemplos aportados durante la fase de entrenamiento.

El éxito y efectividad de los sistemas de aprendizaje automático a la hora de clasificar nuevos elementos depende principalmente de dos factores: del algoritmo de clasificación seleccionado y de las características o *features* elegidas para representar los elementos de ejemplo y con los que entrenar dicho algoritmo. Existen

decenas de algoritmos de aprendizaje automático distintos cuyos resultados además pueden ser mejorados mediante la configuración de sus diferentes parámetros; pero son las características elegidas para el entrenamiento las verdaderamente importantes y de ellas depende en gran medida el éxito de este tipo de métodos de clasificación.

El primer trabajo del que se tiene constancia y que aplica este enfoque es (Pang et al., 2002). Su objetivo era la clasificación de críticas de películas en dos posibles categorías: positivo o negativo. Para ello, los autores elaboraron un corpus basado en las críticas escritas por los usuarios del sitio web Internet Movie Database [15] y con él entrenaron varios algoritmos de aprendizaje automático como Naive Bayes, Máxima Entropía y Máquinas de Vectores de Soporte (SVM). Para representar los textos y llevar a cabo el proceso de entrenamiento hicieron uso de varias características como *unigramas*, *bigramas*, categorías gramaticales a nivel de palabra e información sobre la posición que cada término ocupaba dentro del texto al que pertenecía, así como varias combinaciones de todas ellas. Además, a cada una de estas características se le aplicaba un valor de ponderación entre dos posibles: presencia y frecuencia. Las conclusiones de este estudio es que el algoritmo SVM entrenado con *unigramas* y con ponderación por presencia es la configuración que mejores resultados ofrece.

Sólo un año más tarde, en (Dave et al., 2003) se presenta otro estudio basado en métodos supervisados cuyo objetivo es la clasificación de reseñas de productos tecnológicos y que parte de un corpus compuesto por comentarios extraídos de las webs de Amazon y Cnet [16]. Este trabajo contradice las conclusiones de (Pang et al., 2002) ya que el mejor resultado se obtiene mediante el uso de n-gramas ponderados por su frecuencia relativa y con una implementación específica del algoritmo de Naive Bayes. El contraste de resultados que existe entre estos dos trabajos precursores de los métodos de clasificación supervisada es una muestra de lo que ha ocurrido en los siguientes años de investigación y cuya conclusión es que a día de hoy no parece posible determinar con rotundidad qué combinación de características y algoritmos ofrece los mejores resultados para resolver el problema de la clasificación de sentimientos mediante métodos de aprendizaje supervisado. Aun así, (Martínez Cámara, 2016, pag. 109 y sig.) asegura que en general se obtienen buenos resultados con algoritmos SVM, *unigramas* como características de los textos de entrenamiento y ponderados por su frecuencia relativa en el caso de textos cortos (típicos de redes sociales) o con la fórmula TF-IDF [17] en el caso de textos largos.

Los métodos de clasificación mediante aprendizaje automático son sistemas que ofrecen buenos resultados en el trabajo de clasificación por sentimiento, pero cuentan con dos importantes desventajas. Por

[15] http://www.imdb.com/

[16] https://www.cnet.com

[17] Frecuencia de término – frecuencia inversa de documento o *Term frequency – Inverse document frequency* en inglés

una parte, necesitan un corpus o juego de datos inicial con sus ejemplos previamente clasificados y no siempre es posible contar con él debido al coste que supone tener que categorizar, muchas veces a mano, dichos ejemplos. Además, el tamaño del corpus es fundamental para poder obtener resultados aceptables. Por otra parte, los modelos resultantes son muy dependientes del dominio. Esto significa que un algoritmo entrenado con un corpus sobre comentarios de películas puede no ofrecer el mismo rendimiento a la hora de clasificar reseñas sobre automóviles, ya que una misma palabra no siempre posee el mismo sentimiento en contextos distintos. Por ejemplo, si hablamos de restaurantes, en la frase "La ración era grande", en este contexto la palabra "grande" tiene una connotación positiva. En cambio, si el tema son ordenadores portátiles, el comentario "El portátil es grande" puede indicar que el tamaño del ordenador no es atractivo y, por tanto, en este contexto "grande" tiene una connotación negativa. Por tanto, se hace necesario un juego de pruebas diferentes para cada uno de los dominios con el coste de tiempo y esfuerzo que esto conlleva.

4.2.2 Clasificación mediante aprendizaje no supervisado

Para resolver el hecho de tener que contar con ejemplos etiquetados y salvar el problema de la dependencia de dominio, aparecen en escena los sistemas de clasificación mediante aprendizaje no supervisado. Estos sistemas tratan de inferir la polaridad del sentimiento global de un documento a partir de la orientación semántica de las palabras o frases que lo conforman. Existen dos enfoques para la resolución del problema de clasificación de textos mediante aprendizaje no supervisado: métodos basados en diccionarios y métodos basados en relaciones lingüísticas.

4.2.2.1 Métodos basados en diccionarios

Los métodos basados en diccionarios (o lexicones, del inglés *lexicon*) hacen uso de listados de palabras y frases previamente etiquetadas con la polaridad de sentimiento que expresan y, en ocasiones, además con su intensidad o la fuerza de dicho sentimiento. Los textos a clasificar se dividen en unidades más pequeñas, como palabras o frases, y se buscan en los diccionarios de sentimiento. Así, el sentimiento global del texto vendrá dado por algún tipo de función matemática que tenga en cuenta el sentimiento individual de las unidades de trabajo y en base a lo indicado para ellas en el diccionario.

Un ejemplo de este tipo de aproximaciones es el mostrado en (Taboada et al., 2011). En este estudio se hace uso de un diccionario de palabras y frases previamente etiquetadas teniendo en cuenta su polaridad e intensidad. El sistema extrae las palabras que contienen información de sentimiento mediante un análisis gramatical y que incluye adjetivos, sustantivos, verbos y adverbios. A continuación, busca su puntuación en el diccionario y calcula el sentimiento global del texto por medio de un conjunto de reglas matemáticas. Las conclusiones de este estudio determinan que el uso de métodos basados en diccionarios permite la creación

de sistemas robustos, con buenos resultados independientemente del dominio y que pueden ser mejorados mediante la agregación de múltiples fuentes de conocimiento.

El uso de técnicas basadas en diccionarios facilita la obtención de sistemas de clasificación independientes del dominio, pero esto presenta también algunos inconvenientes. Uno de estos problemas es que en ocasiones se pierde precisión ya que las palabras pueden poseer diferente polaridad dependiendo del contexto en el que se usen. Por ejemplo, el adjetivo "silencioso/a" tiene una connotación positiva si se aplica al ruido que hace una lavadora durante su funcionamiento, pero es negativa si se utiliza en referencia al sistema de sonido de un televisor. Este tipo de problemas se pueden resolver mediante diccionarios construidos a partir de las palabras de un corpus centrado en el dominio que se desea estudiar.

También dentro del mismo dominio pueden existir palabras que no siempre tienen la misma polaridad de sentimiento. Por ejemplo, si hablamos de teléfonos móviles, el adjetivo "largo/a" es positivo en la frase "la batería tiene una autonomía larga". En cambio, en "las aplicaciones necesitan un tiempo largo para arrancar" esa misma palabra es claramente negativa.

Otro inconveniente evidente de este tipo de técnicas es la necesidad de contar con un diccionario de palabras etiquetadas con su sentimiento. Estos diccionarios pueden ser construidos a mano, mediante técnicas automatizadas partiendo de diccionarios ya existentes o extrayendo las palabras que forman parte de un corpus concreto. No obstante, ya existen varios recursos disponibles para ser usados, pero la gran mayoría de éstos son en lengua inglesa. Algunos ejemplos notables son SentiWordNet[18] cuya primera versión fue presentada en (Esuli & Sebastiani, 2007), BLOL[19] desarrollado en (Hu & Liu, 2004) o iSOL[20] , este último recurso en español y que fue presentado en (Molina-González et al., 2013). Aun contando con estos elementos, uno de los problemas de este tipo de sistemas es la dificultad para encontrar diccionarios en cualquier idioma siendo necesario muchas veces hacer traducciones a partir de los ya disponibles.

4.2.2.2 Métodos basados en relaciones lingüísticas

Además de los métodos basados en diccionarios, en los sistemas de clasificación no supervisada existe otro tipo de modelos basados en relaciones lingüísticas. Estos métodos buscan ciertos patrones en los textos que puedan expresar opiniones y sentimientos con mayor probabilidad, extrayendo las palabras que lo forman para luego ser usadas en la categorización del texto global. Para ello, se obtiene la categoría gramatical de las palabras, llamada también *parts-of-speech* o POS en inglés, y se determina si dichos

[18] http://sentiwordnet.isti.cnr.it/
[19] http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar
[20] http://timm.ujaen.es/recursos/isol/

patrones expresan una opinión positiva o negativa. Finalmente, el sentimiento global texto se calcula mediante algún tipo de función matemática.

El trabajo precursor de este tipo de métodos es (Turney, 2002). El método de Turney consistía en extraer frases de los textos a clasificar formadas por adjetivos y adverbios y que respondiesen a una serie de patrones preestablecidos que suelen expresar opiniones y sentimientos. Posteriormente, estimaba el sentimiento de dichos patrones con ayuda del buscador web ya desaparecido AltaVista y dos palabras a modo de semillas: 'excellent' y 'poor'. Para ello, obtenía el número de resultados de cada frase en combinación con cada una de las semillas y mediante el punto de información mutua (*pointwise mutual information* o PMI, en inglés) podía saber si la frase era positiva o negativa. Para finalizar, se usaba el sentimiento parcial de cada frase para calcular el sentimiento global del texto.

Otro ejemplo que hace uso de este tipo de enfoque para clasificar textos es (Hatzivassiloglou & McKeown, 1997). Este trabajo afirma que, dependiendo del conector usado para unir las palabras, éstas tendrán una orientación semántica igual u opuesta. Así, si dos adjetivos están relacionados mediante la conjunción "y" ambas tendrán la misma polaridad de sentimiento. Ocurre así lo contrario con la conjunción "pero". Haciendo uso de un grupo de conectores y de varias palabras semilla es posible obtener una relación de palabras etiquetadas por su sentimiento y en base a ellas deducir el sentimiento global del texto en el que se encuentran.

Este tipo de sistemas resuelven el problema de la clasificación de una manera vistosa y elegante, pero de alguna manera necesitan apoyarse en algún tipo de recurso que verdaderamente aporte orientación semántica como por ejemplo un grupo de palabras semilla o alguna base de conocimiento. Si éste no existe o no cuenta con la calidad suficiente, el sistema completo puede no ofrecer buenos resultados.

5. Análisis de sentimientos en Twitter

En este apartado se presentará un método para resolver el problema de la clasificación de textos por su sentimiento a nivel de documento. Estos documentos serán mensajes que han sido publicados en la red social Twitter y el método elegido estará basado en algoritmos de aprendizaje supervisado. Para ello, será necesario contar con un corpus de entrenamiento cuyos ejemplos deberán haber sido etiquetados previamente con la categoría del sentimiento al que pertenecen. En esta sección se entrenarán varios algoritmos usando diferentes técnicas y se detallarán los pasos necesarios para crear los modelos. De todas las posibles combinaciones se escogerá la mejor en base a una serie de medidas muy utilizadas para evaluar este tipo de sistemas.

NOTA: el código fuente con el que se han realizado las pruebas de este apartado y sus resultados se pueden consultar en detalle en la sección de Anexos de este mismo documento.

5.1 Introducción

Como se explicó en secciones anteriores, existen dos grandes grupos de métodos para resolver el problema de la clasificación de documentos: métodos supervisados y métodos no supervisados. En este TFM se mostrará una solución supervisada, basada en algoritmos de aprendizaje automático y entrenados mediante un corpus formado por miles de tweets clasificados a mano por un grupo de personas.

Esta prueba se divide en dos partes. En la primera se probará la efectividad de varios algoritmos entrenados con los mensajes del corpus y sobre los que se habrán aplicado distintas técnicas de normalización, extracción de las características y métodos de ponderación. La mejor combinación, el clasificador línea de base [21], pasará a una segunda fase para, mediante nuevas técnicas de extracción de características, tratar de mejorar los resultados del modelo.

En las siguientes secciones se presentará el corpus de mensajes y los cuatros algoritmos de aprendizaje automático seleccionados para esta práctica, así como las métricas a usar para determinar de manera objetiva cuáles son los mejores clasificadores configurados. Cerraremos este apartado con las conclusiones extraídas a partir de estos experimentos.

Ya a nivel técnico, destacar que los modelos serán escritos en lenguaje Python[22] y se hará uso de librerías específicas para este tipo de desarrollos como NLTK[23], especializada en el procesamiento de lenguajes

[21] Del inglés, *baseline,* producto que se establece como punto de partida para la elaboración de nuevos productos o estudios de investigación.
[22] https://www.python.org/
[23] http://www.nltk.org/

naturales, y Scikit-Learn[24], que ofrece recursos para la implementación de sistemas de aprendizaje automático.

5.2 Corpus de entrenamiento

Uno de los problemas de los métodos supervisados es la necesidad de contar con un juego de pruebas representativo y previamente etiquetado para entrenar los algoritmos de aprendizaje automático, es decir, un corpus. En el caso concreto de la clasificación de mensajes de Twitter, aunque en la actualidad existen varios recursos en idioma inglés, no es tan sencillo encontrarlos en español. La creación de este tipo de elementos a menudo resulta complicada debido al enorme coste en términos de tiempo y esfuerzo necesarios para completarla. Este fue el problema con el que se encontraron los autores del primer trabajo de análisis de sentimientos en Twitter (Go et al., 2009), ya que en ese momento Twitter no era una red popular y, por tanto, todavía no existían corpus disponibles. Para resolver este obstáculo diseñaron un sistema automatizado que permitía crear un corpus siguiendo las ideas presentadas en (Read, 2005). A grandes rasgos, (Read, 2005) afirmaba que cuando un usuario añadía un emoticono a un determinado texto, este elemento era un indicador del sentimiento de sus palabras escritas. Dicho de otro modo: si a un texto se le añadía un símbolo de *carita feliz* :-) el texto tendría una connotación positiva. Si en cambio el símbolo era una *carita triste* :-(, esas palabras serían negativas. En base a este planteamiento, (Go et al., 2009) generaron su corpus de mensajes mediante búsquedas en Twitter de emoticonos positivos y negativos. La clase de cada mensaje sería la del emoticono usado durante cada búsqueda.

Esta técnica de creación de un corpus es la que utilizan (Pak, A. & Paroubek, P., 2010), pero además introducen una tercera clase para representar mensajes que no transmiten sentimientos positivos ni negativos, una clase neutra. Para ello, (Pak, A. & Paroubek, P., 2010) señalan que dichos mensajes pueden provenir de cuentas de Twitter pertenecientes a medios de comunicación como The New York Times o Washington Post, ya que parten de la idea de que la objetividad carece de opinión y sentimiento, cuestión que contradice lo indicado en anteriores capítulos. En cualquier caso, los mensajes etiquetados con la clase neutra se extraerían de tales cuentas de usuario, completando así el corpus mediante un procedimiento automatizado.

Para las pruebas de este TFM se hará uso de un corpus en español ya existente cuya autoría pertenece al Taller de Análisis de Sentimientos (TASS[25]) de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN[26]). Esta sociedad organiza anualmente una competición en el que se presentan distintos métodos para la clasificación de tweets en cuatro y seis categorías. Comenzó en el año 2012 y ya va por su

[24] http://scikit-learn.org/stable/
[25] http://www.sepln.org/workshops/tass/
[26] http://www.sepln.org/

séptima convocatoria. Aunque no publica cada año un corpus nuevo, existen tres que serán usados en conjunto para el entrenamiento y la evaluación de los clasificadores:

- **General Corpus**, que incluye 68000 tweets escritos en español por 150 personajes y celebridades conocidas dentro del mundo de la política, economía, comunicación y cultura y que fueron obtenidos entre noviembre de 2011 y marzo de 2012.

- **Politics Corpus**, que ofrece 2500 tweets extraídos durante la campaña de Elecciones a las Cortes Generales de España de 2011. Estos mensajes mencionan a los cuatro partidos políticos más relevantes de aquel momento: PP, PSOE, IU y UPyD.

- **International TASS Corpus (InterTASS):** contiene 3400 mensajes de Twitter escritos en español y sobre cualquier tipo de tema.

Los mensajes del corpus se encuentran clasificados en cuatro y seis categorías de sentimiento: Muy positivo (P+), Positivo (P), Neutro (NEU), Negativo (N), Muy negativo (N+) y Sin sentimiento (NONE). A partir de estas seis categorías y mediante la unificación de los mensajes divididos por su intensidad en grupos únicos, se obtiene el sistema de clasificación basado en cuatro clases: Positivo (P), Neutro (NEU), Negativo (N) y Sin sentimiento (NONE). Esta será la clasificación en la que se basarán las pruebas del TFM.

Antes de seguir, es necesario explicar la diferencia entre mensajes sin sentimiento (NONE) y mensajes neutros (NEU). Los primeros son precisamente eso, tweets en los que no se expresa ninguna idea positiva ni negativa. Por ejemplo:

> *"Los retos urgentes de las entidades locales y el papel de la FEMP http://t.co/4EOOg9c4"*
>
> (tweetId: 148406016810299392. Polarity: NONE)

Por otra parte, los mensajes neutros (NEU) poseen un sentimiento a medio camino entre lo positivo y lo negativo y éste puede ser debido a dos razones: que las palabras usadas sean realmente neutras (AGREEMENT) o bien que contengan palabras tanto positivas como negativas en el mismo mensaje (DISAGREEMENT):

> *Rajoy: ''Intentaremos repartir de manera equitativa los costos de esta crisis económica. La primera obligación de un gobernante es ser justo"*
>
> (tweetId: 146471403909169152. Polarity: NEU - AGREEMENT)

> *"Soy y seré del grupo parlamentario durante la legislatura pero discrepo de la exclusión y el reparto. Por eso seré diputado leal pero raso."*
>
> (tweetId: 147378792556539906. Polarity: NEU - DISAGREEMENT)

Los mensajes de estos tres corpus han sido clasificados a mano. Debido a que tienen un formato diferente, se creará un nuevo corpus global a partir de la unión de todos ellos, pero con la información estrictamente necesaria para entrenar los algoritmos de aprendizaje supervisado.

La siguiente tabla muestra el número de tweets de cada clase y para cada una de las colecciones a las que pertenecen:

	Positivo (P)	Negativo (N)	Neutro (NEU)	Sin sentimiento (NONE)
General Corpus	25117	18026	1975	22899
Politics Corpus	613	681	933	221
International TASS Corpus	1116	1404	418	475
TOTAL (%)	26846 (36%)	20111 (27%)	3326 (5%)	23595 (32%)

TABLA 5.2.1 - DISTRIBUCIÓN DE NÚMERO DE CLASES POR CORPUS

No es difícil advertir la gran diferencia que existe entre el número de mensajes clasificados como neutros y el resto de mensajes. Aunque lo ideal en este tipo de pruebas es que exista equilibrio entre el número de muestras de cada clase, en algunas ocasiones esto no es posible. Esta situación se deberá tener en cuenta a la hora de medir la eficacia de los clasificadores.

5.3 Algoritmos de clasificación

Los algoritmos de aprendizaje supervisado se pueden dividir principalmente en dos grandes grupos: de regresión y de clasificación. Los primeros permiten inferir un valor numérico a partir de una serie de datos de entrada, por ejemplo, las ventas que tendrá una determinada empresa. En cambio, los de clasificación se utilizan para deducir a qué grupo pertenece un ejemplo dado de entre los grupos disponibles. Aunque ambos tipos de algoritmos pueden ser usados en el análisis de sentimientos, nos centraremos en cuatro algoritmos de clasificación muy populares y que ya han sido utilizados en múltiples ocasiones para esta tarea: Naive Bayes, máquinas de vectores de soporte, K vecinos más cercanos y árboles de decisión.

5.3.1 Naive Bayes

La familia de algoritmos Naive Bayes están basados en el célebre Teorema de Bayes, el cual dice lo siguiente:

Sea {A_1, A_2, … , A_n} un conjunto de sucesos mutuamente excluyentes y exhaustivos, y tales que la probabilidad de cada uno de ellos es distinta de cero. Sea B un suceso cualquiera del que se conocen las probabilidades condicionales $P(B|A_i)$. Entonces, la probabilidad de $P(A_i|B)$ viene da por la expresión:

$$P(A_i|B) = \frac{P(B|A_i) * P(A_i)}{P(B)}$$

En el caso concreto de la clasificación de textos, los sucesos excluyentes y exhaustivos son las diferentes clases que se pueden asignar a un mensaje, de manera que no es posible asignar más de una simultáneamente (excluyentes) y esas clases son todos los tipos que existen (exhaustivos). Los algoritmos Naive Bayes suelen recibir el apelativo de "ingenuos" debido a que en sus cálculos las características seleccionadas para representar a los ejemplos de entrenamiento son estadísticamente independientes y contribuyen por igual en el proceso de clasificación. Dicho de otro modo y en el caso concreto de la clasificación de textos, se considera que las palabras de un mismo mensaje no mantienen ningún tipo de relación entre sí y es indiferente la posición que tienen dentro del texto al que pertenecen.

5.3.2 Máquinas de vectores de soporte

Las máquinas de vectores de soporte (del inglés, Support Vector Machine o SVM) son un grupo de algoritmos de aprendizaje supervisado desarrollados por (Vapnik, 1982) en los laboratorios AT&T. De manera visual, podemos pensar en este tipo de algoritmos como la representación gráfica de un espacio multidimensional en donde se sitúan los puntos que simbolizan los ejemplos de entrenamiento. Un *hiperplano*, denominado vector de soporte, los separa la mayor distancia posible en base a su clase. De esta forma, el vector determina la frontera que sirve para clasificar un nuevo elemento, por lo que dependiendo a qué parte del espacio pertenezca, se le asignará una clase u otra.

Este tipo de algoritmos cuenta con una serie de parámetros que permiten ajustar su configuración interna y así optimizar los resultados durante el proceso de clasificación. Uno de estos parámetros es el *kernel* y se utiliza cuando no es posible separar las muestras mediante una línea recta, plano o hiperplano de N dimensiones, permitiendo tal separación mediante otro tipo de funciones matemáticas como polinomios, funciones de base radial Gaussiana, Sigmoid u otras. Otro de estos parámetros es *regularization* (también conocido como "C") que permite crear un margen blando de manera que se consientan ciertos errores en la clasificación y se evite el sobreentrenamiento. Y, para terminar, el parámetro *gamma* determina la distancia

máxima a partir de la cual una muestra pierde su influencia en la configuración del vector de soporte, y *margin*, que es la separación entre el vector y las muestras de cada clase más cercanas al mismo. [Fuente imagen[27]]

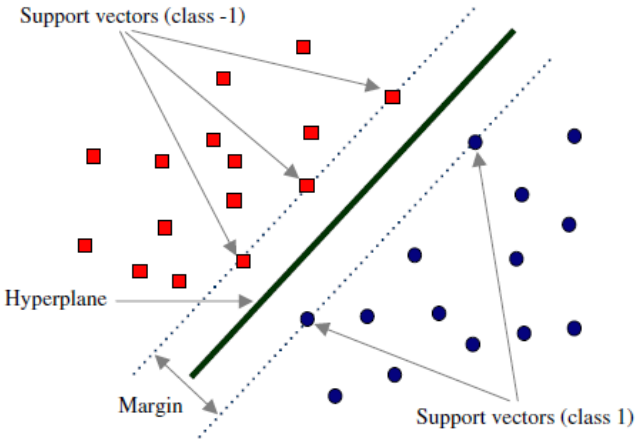

FIGURA 5.3.2.1 - MÁQUINA DE VECTORES DE SOPORTE

5.3.3 K vecinos más cercanos

El algoritmo k-vecinos más cercanos (del inglés, k-nearest neighbors o k-nn) hace uso de una función de similitud para la clasificación de los elementos que se le proporcionan una vez el algoritmo haya sido entrenado. De esta forma, cuando se quiere predecir la clase de un nuevo ejemplo, se buscan los k ejemplos con los que la función de similitud sea máxima y se le asigna la clase mayoritaria de entre los seleccionados. [Fuente imagen[28]]

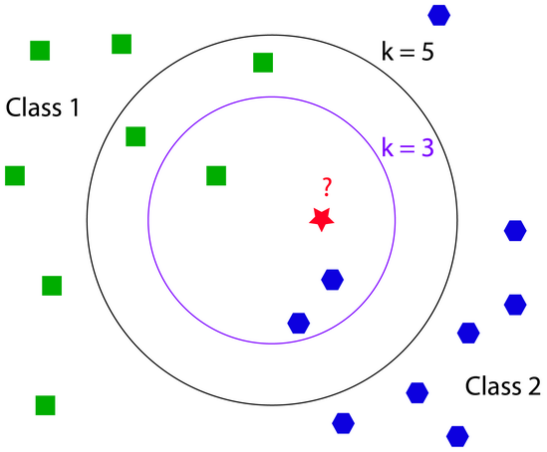

FIGURA 5.3.3.1 - K VECINOS MÁS CERCANOS

[27] http://penseeartificielle.fr/comprendre-langage-poules-grace-algorithme-svm/
[28] https://mertricks.com/category/machine-learning/

Este clasificador se dice “retardado” o “vago” debido a que no crea ningún modelo estadístico a partir de los datos de entrenamiento, sino que los memoriza y los utiliza cada vez que tiene que predecir a qué grupo pertenece un nuevo elemento. Su éxito depende en gran medida del parámetro k, es decir, de la cantidad de ejemplos vecinos que el algoritmo utiliza para determinar a qué clase pertenece un ejemplo dado.

5.3.4 Árboles de decisión

Los árboles de decisión forman uno de los grupos de algoritmos más reconocidos y utilizados dentro del campo de la Inteligencia Artificial y del aprendizaje automático. Su estructura es la de un grafo dirigido en forma de árbol compuesto por un conjunto de reglas extraídas a partir de las características de los datos de entrenamiento y que se aplican de manera sucesiva a la hora de predecir a qué clase pertenece un nuevo ejemplo. En general, un árbol de decisión está formado por nodos y líneas que unen dichos nodos, comenzando en uno raíz y terminando en varios con las posibles clasificaciones que se pueden establecer a una muestra dada. Partiendo de la raíz, el paso entre los distintos nodos del árbol se lleva a cabo mediante la evaluación de algún tipo de condición y que determina el recorrido que seguirá la muestra hasta encontrar la clase que le corresponde en un nodo terminal. [Fuente imagen[29]]

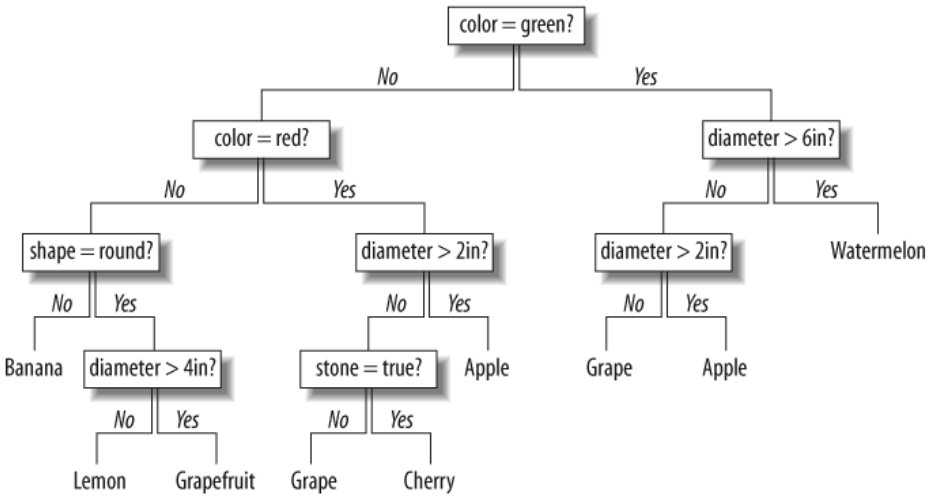

FIGURA 5.3.4.1 - ÁRBOLES DE DECISIÓN

5.4 Métricas y métodos de evaluación de resultados

Para determinar el rendimiento de los algoritmos y de su configuración, es necesario contar con una serie de medidas que permitan evaluar de manera objetiva su eficacia a la hora de clasificar los ejemplos que se le proporcionen. Para ello, es importante no tener sólo en cuenta las muestras clasificadas correcta e incorrectamente, sino también las que habiéndose clasificado de manera errónea podrían haberse

[29] https://www.safaribooksonline.com/library/view/programming-collective-intelligence/9780596529321/ch12s02.html

etiquetado bien. Para entender los cuatro posibles estados de un ejemplo a clasificar, pensemos en una clase A y en un algoritmo que determina si dicho ejemplo pertenece o no a esa clase:

- **True Positives (Verdaderos Positivos o TP):** son los ejemplos que han sido marcados de manera correcta como pertenecientes a la clase A.

- **False Positives (Falsos Positivos o FP):** serán los ejemplos marcados como de clase A, pero en realidad no pertenecen a ella, es decir, han sido clasificados de manera incorrecta.

- **True Negatives (Verdaderos Negativos o TN):** en este caso, los ejemplos no son de la clase A y han sido clasificados correctamente.

- **False Negatives (Falsos Negativos o FN):** en este grupo estarán los ejemplos marcados como no pertenecientes a la clase A, pero en realidad sí lo son y, por tanto, no se han clasificado correctamente.

Teniendo en cuenta los estados anteriores, podemos definir las siguientes medidas que serán usadas para evaluar nuestros modelos:

- **Exactitud (del inglés Accuracy):** esta es la medida de rendimiento más simple e intuitiva y representa la razón entre las predicciones correctas sobre el total de predicciones realizadas. Dicho de otra manera, es el número de elementos clasificados correctamente entre el número total de clasificaciones llevadas a cabo.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

  Es habitual pensar que el modelo que ofrezca una mayor exactitud es el mejor modelo. En realidad, esta medida es adecuada en el caso de que el número de elementos de cada clase sea aproximadamente el mismo y el corpus esté balanceado. En caso contrario, es necesario hacer uso de otro tipo de medidas como la precisión, la exhaustividad y el valor-F. Al contrario que la exactitud, estas medidas no valoran el rendimiento del modelo teniendo en cuenta todas las clases del sistema, sino que lo hacen sobre clases individuales. Dicho de otro modo, la precisión, la exhaustividad y el valor-F ofrecerán valores distintos para la clase A y para la B.

- **Precisión (del inglés Precision):** es la razón entre el número de documentos clasificados correctamente como pertenecientes a la clase A y el número total de documentos de que han sido clasificados por el modelo como de clase A.

$$Precision = \frac{TP}{TP + FP}$$

La precisión mide la proporción de identificaciones positivas que son realmente correctas. Nótese que su valor aumenta a medida que el número de falsos positivos disminuye.

- **Exhaustividad (del inglés Recall):** es la relación entre los documentos clasificados correctamente como pertenecientes a la clase A y la suma de todos los documentos de la clase A.

$$Recall = \frac{TP}{TP + FN}$$

La cobertura es la proporción de elementos positivos reales identificados acertadamente. También se puede ver como la capacidad que tiene el modelo de construir de manera correcta las clases. Cuanto más cercano a 1, mejor estarán definidas las distintas clases existentes ya que su valor aumenta a medida que disminuye el número de falsos negativos.

- **Valor-F (del inglés F-score):** es habitual que para medir la eficiencia de un modelo de clasificación se haga uso de los valores de cobertura y exhaustividad. Para ello, el valor-F se presenta como la media armónica entre ambas medidas y suele utilizarse como referencia para comparar el rendimiento entre varios modelos. La fórmula del valor-F combina las dos medidas anteriores de manera ponderada a través de un parámetro β lo que permite otorgar una mayor importancia a una que a otra:

$$F_\beta = (1 + \beta^2) * \frac{Precision * Recall}{(\beta^2 * Precision) + Recall}$$

Es frecuente que la precisión y la exhaustividad tengan el mismo peso en la fórmula, es decir, con un valor β igual a 1. A esta configuración se le conoce como Valor-F_1 o F_1-score.

En el caso de un sistema que cuente con más de dos clases, como el nuestro, se debe calcular cada una de las métricas anteriores para cada clase y combinarlas entre ellas para obtener una medida global. Para ello, existen tres posibilidades:

- **Macro-averaging:** en este caso se calculan las medidas de cada clase para, a continuación, calcular la media aritmética:

$$Macro - Precision = \frac{\sum_{i=1}^{n} Precision_i}{n}$$

$$Macro - Recall = \frac{\sum_{i=1}^{n} Recall_i}{n}$$

El problema es que estas medidas no tienen en cuenta la posible desigualdad entre el número de ejemplos de cada clase por lo que sus resultados pueden no ser fiables en ese tipo de escenarios.

- **Micro-averaging:** en este caso se tienen en cuenta el número de elementos de cada clase ya que hacen uso de todos los resultados para el cálculo de los indicadores:

$$Micro - Precision = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i}$$

$$Micro - Recall = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i}$$

La medidas *micro-averaging* sí tienen en cuenta la desigualdad entre el número de ejemplos de cada tipo, pero en el caso de que el sistema cuente con más de dos clases distintas, tenemos que $\sum_{i=1}^{n} FP_i = \sum_{i=1}^{n} FN_i$ por lo que la precisión, la exhaustividad y el valor-F_1 tomarán siempre los mismos valores.

- **Weighted-averaging:** para resolver el problema de tener un corpus desequilibrado y con más de dos clases distintas, la librería Scikit-Learn cuenta con un grupo de medidas adicionales basadas en las fórmulas *macro-averaging*. En ellas se pondera cada componente de la fórmula en base al peso que cada clase tiene dentro del sistema global:

$$Weighted - Precision = \frac{\sum_{i=1}^{n} Precision_i * P_i}{n}$$

$$Weighted - Recall = \frac{\sum_{i=1}^{n} Recall_i * P_i}{n}$$

Estas medidas ponderadas serán las que determinarán cuál de nuestros modelos tiene el mejor rendimiento.

Por otra parte, es importante explicar que los modelos de este apartado serán construidos mediante un sistema de validación cruzada (del inglés cross-validation). Este método reduce la dependencia entre la partición de datos usada para la fase de entrenamiento y para la de pruebas. La validación cruzada divide el número de muestras del corpus en k conjuntos de igual tamaño de manera que k-1 grupos son usados para entrenar el sistema y el grupo sobrante para su evaluación. Este proceso se repetirá k veces y en cada uno de ellos se escogerá un grupo diferente para validar la eficacia del modelo. En cada iteración se calculará el

valor-F_1 ponderado y se hallará la media entre todos, medida que servirá como referencia de su efectividad. La validación cruzada suele tomar los valores 3, 5 o 10 para el parámetro k. En nuestro caso concreto será 10.

5.5 Proceso de entrenamiento de los algoritmos

El proceso habitual para la construcción de un clasificador de textos basado en un sistema de aprendizaje automático consta de varias etapas secuenciales. En primer lugar, es necesario preparar los datos del corpus para entrenar los algoritmos. Para ello, se debe limpiar y normalizar su información con el objetivo de reducir o eliminar aquellos datos que puedan influir de manera negativa en el resultado final. A continuación, cada uno de los textos de ejemplo se somete a un proceso denominado *tokenización,* el cual los divide en unidades más pequeñas o *tokens* y que habitualmente son las palabras de los mensajes. A partir de los *tokens* se extraen las características que representen a los mensajes originales y, de manera opcional, se puede aplicar de un método para reducir su número. Para finalizar, estas características se ponderan en función de la importancia que se les quiera dar y con ellas se entrenan los clasificadores.

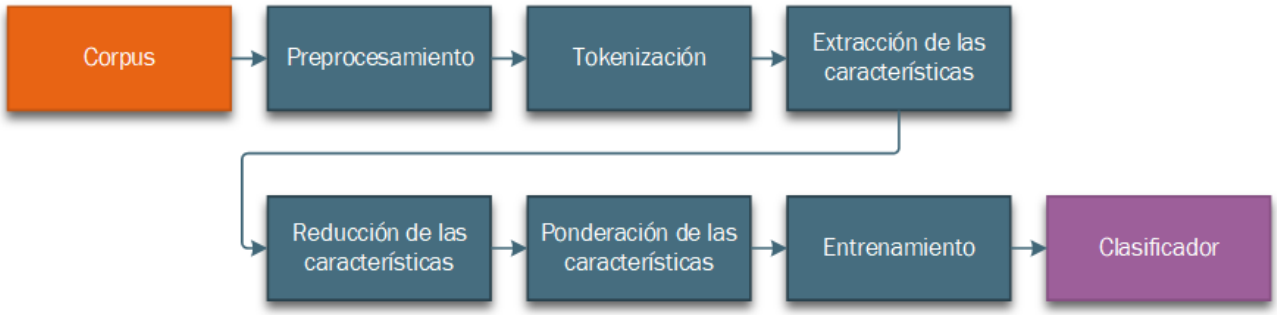

FIGURA 5.5.1 - FASES PARA EL ENTRENAMIENTO DE ALGORITMOS DE APRENDIZAJE SUPERVISADO

5.5.1 Preprocesamiento

En todo método que se haga uso de algoritmos de aprendizaje automático es necesario tratar previamente los datos con los que serán entrenados. El objetivo de esta fase es limpiar y normalizar la información para evitar que determinados datos puedan influir de manera negativa en el resultado final. Esta cuestión es crucial cuando hablamos de mensajes extraídos de redes sociales ya que es muy habitual encontrar mensajes con faltas de ortografía, repeticiones de caracteres, mezcla de letras mayúsculas y minúsculas o, en el caso concreto de redes de microblogging como Twitter, uso de jerga y abreviaturas para escribir mayor contenido en un número tan reducido de caracteres.

La normalización de mensajes de Twitter es un trabajo complejo y laborioso que llega a ser un arte en sí mismo. Su importancia es tal que se han organizado diversos talleres sobre este tema en Internet en los

últimos años, como TweetNorm[30], organizado por la Sociedad Española para el Procesamiento del Lenguaje Natural, la misma entidad organizadora de TASS.

Para este TFM se han seleccionado un conjunto de reglas sencillas de aplicar y que suelen ser comunes en la construcción de este tipo de clasificadores. El objetivo que persiguen todos ellos es la normalización de los mensajes, pero evitando en todo momento que los cambios aplicados provoquen la pérdida de la polaridad de sentimiento. Estas son las reglas a utilizar:

- **Normalización de mayúsculas y minúsculas:** aunque para las personas es sencillo saber que las palabras “coche” y “COCHE” tienen exactamente el mismo significado, para los algoritmos de aprendizaje automático esto no así. De hecho, son tratadas como palabras totalmente distintas, sin ningún tipo de relación entre ellas. Para evitar que esto suceda y mantener el significado de las palabras sin tener en cuenta la forma de sus caracteres, todos los mensajes serán convertidos a su equivalente en letras minúsculas.

- **Tratamiento de la duplicidad de caracteres:** en las redes sociales es habitual repetir las mismas letras en las palabras para dar intensidad a lo que se intenta expresar. Por ejemplo, no es lo mismo escribir “Qué calor hace” o “Qué calooooooor haceeeeeee”. Aunque ambas frases conceptualmente son equivalentes, la segunda enfatiza la sensación de calor mediante la repetición de los caracteres de la frase. En nuestro caso, no necesitamos conocer en qué grado se emiten las opiniones, sólo saber qué polaridad tienen por lo que no es necesario mantener estas repeticiones. Por tanto, toda secuencia de tres o más caracteres iguales se reducirá a sólo dos. Por ejemplo, el caso anterior quedaría como “Qué caloor hacee”. La razón por la que no se reducen a una única letra es porque en español es válido escribir determinados caracteres hasta dos veces consecutivas para letras como la erre, ele, ene, eme y varias vocales. El proceso de reducción de letras repetidas permitirá establecer relaciones entre palabras que realmente son iguales. Por ejemplo, “Vaaaamoooos”, “Vaaamooooooos” y “Vaaaamooos” serán la misma palabra: “Vaamoos”.

- **Eliminación de tildes:** en las redes sociales los usuarios no acostumbran a hacer un buen uso de las tildes. Por esta razón, las palabras “alegría” y “alegria" (sin tilde en la letra i) serían consideradas por los algoritmos como distintas. Para evitar esta pérdida de relación semántica, serán eliminadas todas las tildes de las vocales de los mensajes de entrenamiento.

[30] http://komunitatea.elhuyar.eus/tweet-norm/

- **Eliminación de números:** por norma general, las cifras numéricas no suelen contener información que ayude al proceso de clasificación de polaridad de sentimiento por lo que serán removidas de los textos y así ayudar a reducir la cantidad de características del corpus.

- **Eliminación de *retweets:*** aunque en la actualidad esto ya no es así, hace unos años, justo cuando se extrajeron gran parte de los tweets que forman el corpus de entrenamiento, las menciones a las publicaciones de otros usuarios se hacían mediante el uso de la palabra reservada "RT" seguida del nombre de usuario y del texto a referenciar. Esta palabra clave no aporta información relevante para el proceso de clasificación por lo que será eliminada de todos los mensajes.

- **Eliminación de retornos de carro:** algunos mensajes de Twitter contienen saltos de línea y retornos de carro, por lo que el texto aparece escrito en diferentes líneas. Estos elementos serán también eliminados para que los mensajes aparezcan escritos en una sola línea.

- **Normalización de risas:** en muchas ocasiones y de manera especial antes de la popularización de los *emojis*[31] , es habitual que los mensajes se acompañen de la onomatopeya de la risa para otorgar un mayor énfasis a aquello que se está narrando. Este símbolo puede ser escrito de múltiples y distintas maneras como "jajaja", "jeje", "jijijiji", etc. Teniendo en cuenta lo anterior, no es difícil encontrarse con otras combinaciones como "ajaajajjaj", "jojjojoj" o "jaaaajjj" o incluso "juass" y el acrónimo LOL (del inglés, *Laughing out loud*) que significa "reirse en voz alta". La representación escrita de la risa suele ser un indicador importante de polaridad de sentimiento en textos por lo que es fundamental normalizar estas secuencias de caracteres y representarlas exactamente igual.

- **Menciones, enlaces y hashtags:** estos tres elementos son muy comunes en los mensajes de Twitter. Las menciones sirven para hacer referencia a otros usuarios de la red social mediante su nombre precedido del símbolo de la arroba (por ejemplo, @UOCuniversidad). Los llamados hashtags son cadenas de texto con algún significado precedidas por el símbolo de la almohadilla. Actúan como etiquetas que se incluyen en los mensajes y permiten darles un contexto (por ejemplo, #MachineLearning). Por último, es posible añadir a los tweets enlaces a páginas web para enriquecer los mensajes (por ejemplo: http://t.co/4diTkV2a). Debido a la variabilidad de estos tres elementos, por sí mismos no parece que puedan ayudar a determinar la polaridad de los mensajes en los que están incluidos. No obstante, quizás sí lo hagan si se unifican y normalizan. Para comprobarlo, se les aplicarán dos tratamientos distintos: eliminarlos de los mensajes o normalizarlos, es decir, sustituirlos por palabras clave que los representen.

[31] Imágenes o pictogramas usados para expresar ideas, sentimientos o emociones en medios de comunicación digital y considerados como los sucesores de los emoticos.

- **Normalización de jerga:** en las redes sociales se utiliza un lenguaje coloquial y muy informal que hace uso intensivo de abreviaturas y secuencias de caracteres sin aparente sentido. Esta jerga aumenta en Twitter debido a la limitación del número de caracteres permitidos por mensajes. Así, es habitual escribir "q" en lugar de "que", "tb" en lugar de "también" o el clásico "tqm" para referirse a "te quiero mucho". El proceso de normalización tendrá en cuenta esta realidad y aplicará un conjunto de reglas para sustituir estas abreviaturas por sus palabras reales.

5.5.2 Tokenización

Una vez completado el proceso de normalización de los mensajes del corpus, la siguiente etapa es la denominada *tokenización*. En esta fase los textos se dividen en unidades más pequeñas llamadas *tokens* y que normalmente se corresponden con las palabras de cada texto. Este proceso puede ser tan sencillo como separar los términos de las frases por los espacios en blanco y los caracteres de puntuación o bien considerar además que la agrupación de determinados símbolos puede contener algún tipo de información que sea útil al proceso de clasificación. Este podría ser el caso de los emoticonos, secuencias de caracteres de puntuación que suelen ser un indicador de la polaridad del sentimiento de las palabras a las que acompañan. En nuestro caso, se hará uso de un *tokenizador* específico para mensajes de Twitter que mantienen los emoticones, menciones, *hashtags* y URLs como *tokens.*

5.5.3 Extracción de las características

A partir de los *tokens* obtenidos en el paso anterior, se definirá la manera de representar con ellos los mensajes de los que proceden, creando así las llamadas características. Lo habitual en la tarea de clasificación de textos es hacer uso del modelo de bolsa de palabras (del inglés, bag of words o BoW) en donde cada mensaje se representa mediante sus *tokens* sin tener en cuenta ningún orden concreto entre ellos. Esta bolsa de palabras puede contener *unigramas*, es decir, *tokens* independientes, *bigramas*, formados por la concatenación de dos *tokens* preservando el orden original que éstos tenían dentro del mensaje del que proceden, *trigramas*, etc. En nuestro caso concreto, las características serán *unigramas* o *tokens* individuales.

5.5.4 Reducción de las características

Esta fase es opcional y su objetivo es disminuir el número de características del corpus mediante la eliminación de determinados *tokens* o de su conversión buscando una misma manera de representarlos. Existe tres técnicas habituales para llevar a cabo esta tarea: eliminación de *stopwords*, lematización y *stemming*.

- **Eliminación de stopwords:** existe un conjunto de palabras que, aunque son necesarias para construir oraciones con sentido, carecen de información que ayude a determinar la polaridad de los textos en los que se encuentran. En español, estas palabras son las preposiciones, los pronombres, las conjunciones y las distintas formas del verbo haber, entre otras. Mediante esta técnica, todos los términos pertenecientes a la lista de *stopwords* serán eliminadas del modelo antes del entrenamiento de los algoritmos.

- **Lematización:** este es un proceso de normalización morfológica que transforma cada palabra en su lema mediante el uso de diccionarios y de un proceso de análisis morfológico. A modo de ejemplo, la lematización convertiría la palabra “guapas” a su lema “guapo”. Por tanto, muchas características tomarían la misma forma, reduciendo así su variabilidad.

- **Stemming:** se trata de otro método de normalización morfológica pero más agresivo que la lematización. En este caso, una palabra se transforma a su raíz por medio de la supresión de sus sufijos e inflexiones. Siguiendo el ejemplo anterior, la palabra “guapas” se convertiría a su raíz, “guap”.

Como se verá más adelante, en nuestras pruebas se medirá la eficacia de la eliminación de la lista de *stopwords* y de la normalización mediante la aplicación de la técnica de *stemming*.

5.5.5 Ponderación de las características

Las características extraídas de las fases anteriores pueden ser consideradas todas de igual importancia u otorgarles distintos pesos en función de algún tipo de criterio. Aunque existen múltiples métodos de ponderación, hay cuatro modelos muy populares en la clasificación de textos y que tienen su origen en el campo de la Recuperación de la Información. Dichos pesos determinan la relevancia de cada característica dentro del mensaje al que pertenecen y, por tanto, influyen a la hora de clasificar los textos por parte de los algoritmos de aprendizaje supervisado:

- **Ponderación binaria (del inglés Binary Term Ocurrences o BTO):** dada una lista con todas las características de todos los mensajes del corpus de entrenamiento, para cada mensaje se indicará con un valor 1 aquellas características que formen parte del mismo, y con un 0 en caso contrario.

- **Frecuencia absoluta (del inglés Term Ocurrences o TO):** en este caso, cada característica tendrá un peso igual al número de veces que aparece en un mensaje dado.

- **Frecuencia relativa (del inglés Term Frequency o TF):** este modelo de ponderación es igual al anterior, pero al valor de cada característica se le aplica un proceso de normalización Euclídea[32] que tiene en cuenta el número de características del mensaje al que pertenecen y sus frecuencias absolutas.

- **Esquema TF-IDF (del inglés Term Frequency – Inverse Document Frequency):** este método otorga una mayor importancia a aquellas características que aparecen un mayor número de veces en el corpus, pero en pocos mensajes del mismo. Estos términos son los que suelen ayudar a identificar con mayor facilidad las distintas clases existentes. De esta forma, se evitan los problemas que implica el uso la frecuencia absoluta o relativa en donde las características más repetidas son las que tienen mayor importancia independientemente del tipo de mensajes en el que aparezca. TF-IDF[33] es una medida muy utilizada en la tarea de clasificación de textos y en el campo de la Recuperación de Información.

5.6 Clasificador línea de base

En este primer apartado práctico se buscará el clasificador línea de base mediante la combinación de varias técnicas y que será el punto de partida de una siguiente fase en la que se intentará mejorar su rendimiento. Estas combinaciones estarán formadas los siguientes elementos:

- **Algoritmos de aprendizaje supervisado:** serán los cuatro algoritmos presentados en anteriores apartados, es decir, Naive Bayes, máquinas de vectores de soporte, k vecinos más cercanos y árboles de decisión. Se debe destacar que el clasificador de máquinas de vectores tendrá un *kernel* lineal, k vecinos más cercanos un valor k igual a 200 y el clasificador de árboles de decisión usa una implementación del algoritmo CART. La elección de los dos primeros casos se debe a que los resultados más prometedores se obtienen con esas configuraciones según las pruebas preliminares realizadas al principio de este TFM. En el caso de los árboles de decisión, CART es la implementación incluida en las librerías usadas en esta práctica.

- **Preprocesamiento del corpus:** se emplearán todas las reglas descritas anteriormente, pero se medirá por separado el rendimiento de los modelos con la normalización de los elementos de Twitter y con su borrado.

[32] https://es.wikipedia.org/wiki/Norma_vectorial#Definición_de_norma_euclídea
[33] https://es.wikipedia.org/wiki/Tf-idf

- **Reducción de las características:** se probarán cuatro combinaciones distintas de reducción del espacio vectorial: sin ninguna técnica, sólo aplicando la eliminación de las *stopwords*, sólo mediante *stemming* y con ambas a la vez.

- **Ponderación de las características:** existirán modelos con cada una de las cuatro técnicas presentadas: ponderación binaria, frecuencia absoluta, frecuencia relativa y TF-IDF.

Debido a que el corpus completo es demasiado extenso y el número de pruebas a realizar también, se ha extraído un subconjunto de mensajes a partir de dicho corpus formado por el 30% de las muestras y respetando la proporción de cada clase. Para obtener una mayor fiabilidad en los resultados y evitar efectos no deseados como el sobreentrenamiento (del inglés, overfitting), se probarán todos los modelos mediante un esquema de validación cruzada[34] de k iteraciones en donde k es igual a 10. La medida de referencia será el valor-F_1 ponderado.

La siguiente tabla muestra los resultados obtenidos para cada una de las configuraciones de los modelos. Como se puede observar, se ha aplicado un mapa de color en donde los tonos verdes representan los mejores valores, y los rojos, los peores. Además, para cada algoritmo se ha resaltado en blanco su mejor resultado:

[34] https://es.wikipedia.org/wiki/Validación_cruzada

					MÁQUINA DE VECTORES			NAIVE BAYES			ÁRBOLES DE DECISIÓN			K VECINOS MÁS CERCANOS		
#	POND.	TWITTER	STOPWORDS	STEMMING	PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE
1	BTO	REMOVE	FALSE	FALSE	64,91%	65,73%	65,20%	61,54%	63,20%	60,82%	51,92%	52,41%	52,04%	43,19%	31,80%	15,49%
2	BTO	REMOVE	TRUE	FALSE	64,52%	65,24%	64,69%	60,69%	62,74%	60,47%	56,18%	56,57%	56,03%	46,90%	32,10%	16,11%
3	BTO	REMOVE	FALSE	TRUE	65,29%	66,03%	65,58%	61,04%	63,25%	61,04%	53,53%	54,05%	53,67%	46,88%	31,83%	15,55%
4	BTO	REMOVE	TRUE	TRUE	64,47%	65,23%	64,76%	60,50%	62,89%	60,69%	**56,61%**	**57,28%**	**56,77%**	46,91%	32,13%	16,18%
5	BTO	NORMAL.	FALSE	FALSE	65,26%	66,06%	65,53%	**62,67%**	**64,52%**	**62,39%**	52,03%	52,47%	52,13%	46,94%	32,51%	16,94%
6	BTO	NORMAL.	TRUE	FALSE	64,80%	65,50%	64,95%	62,31%	63,94%	61,90%	55,36%	55,63%	55,05%	46,71%	33,34%	18,62%
7	BTO	NORMAL.	FALSE	TRUE	65,27%	65,99%	65,55%	**62,13%**	**64,37%**	**62,39%**	53,43%	53,89%	53,54%	46,59%	32,66%	17,25%
8	BTO	NORMAL.	TRUE	TRUE	64,48%	65,23%	64,76%	61,63%	64,09%	62,10%	56,23%	56,79%	56,29%	46,69%	33,30%	18,54%
9	TO	REMOVE	FALSE	FALSE	64,99%	65,76%	65,26%	61,58%	63,06%	60,73%	51,88%	52,50%	52,09%	53,25%	34,87%	22,39%
10	TO	REMOVE	TRUE	FALSE	64,28%	64,98%	64,45%	60,26%	62,29%	60,08%	55,32%	55,74%	55,18%	47,55%	34,85%	22,41%
11	TO	REMOVE	FALSE	TRUE	65,17%	65,88%	65,46%	60,78%	62,77%	60,63%	53,04%	53,70%	53,28%	53,98%	35,03%	22,70%
12	TO	REMOVE	TRUE	TRUE	64,45%	65,17%	64,72%	60,15%	62,42%	60,29%	56,09%	56,75%	56,21%	48,98%	35,07%	22,70%
13	TO	NORMAL.	FALSE	FALSE	65,36%	66,15%	65,63%	62,19%	63,88%	61,79%	52,34%	52,83%	52,49%	52,85%	36,30%	25,28%
14	TO	NORMAL.	TRUE	FALSE	64,51%	65,17%	64,65%	61,18%	63,28%	61,25%	54,67%	55,18%	54,62%	47,73%	36,23%	25,17%
15	TO	NORMAL.	FALSE	TRUE	65,31%	66,05%	65,60%	61,88%	63,77%	61,84%	53,12%	53,59%	53,28%	53,90%	36,58%	25,71%
16	TO	NORMAL.	TRUE	TRUE	64,57%	65,28%	64,83%	60,86%	63,25%	61,33%	55,71%	56,38%	55,86%	49,90%	36,26%	25,09%
17	TF	REMOVE	FALSE	FALSE	67,12%	68,67%	67,28%	59,10%	59,07%	55,48%	50,50%	50,98%	50,67%	51,14%	38,81%	31,45%
18	TF	REMOVE	TRUE	FALSE	65,78%	67,33%	66,13%	59,24%	60,82%	57,91%	54,06%	54,54%	53,99%	47,00%	39,52%	31,97%
19	TF	REMOVE	FALSE	TRUE	67,73%	69,34%	67,91%	59,66%	60,79%	57,58%	51,62%	52,16%	51,83%	51,09%	38,97%	31,63%
20	TF	REMOVE	TRUE	TRUE	66,55%	68,31%	66,94%	59,64%	61,45%	58,56%	54,87%	55,46%	55,07%	48,97%	40,13%	32,45%
21	TF	NORMAL.	FALSE	FALSE	67,10%	68,80%	67,42%	59,53%	59,23%	55,99%	50,38%	50,83%	50,56%	50,16%	52,28%	51,09%
22	TF	NORMAL.	TRUE	FALSE	66,12%	67,77%	66,52%	59,16%	60,29%	57,52%	52,51%	52,85%	52,45%	46,33%	47,41%	44,18%
23	TF	NORMAL.	FALSE	TRUE	**68,08%**	**69,57%**	**68,15%**	59,94%	60,94%	57,99%	52,34%	52,82%	52,53%	50,88%	53,10%	51,87%
24	TF	NORMAL.	TRUE	TRUE	66,71%	68,38%	67,04%	59,89%	61,45%	58,78%	54,34%	54,86%	54,44%	47,51%	48,88%	45,95%
25	TF-IDF	REMOVE	FALSE	FALSE	65,66%	67,22%	66,09%	60,03%	61,68%	58,71%	49,67%	50,19%	49,85%	65,13%	42,09%	34,22%
26	TF-IDF	REMOVE	TRUE	FALSE	64,49%	66,06%	65,00%	59,67%	61,92%	59,28%	53,51%	54,04%	53,54%	68,15%	40,47%	31,18%
27	TF-IDF	REMOVE	FALSE	TRUE	66,20%	67,96%	66,78%	59,73%	61,94%	59,07%	51,53%	51,95%	51,68%	65,27%	42,01%	34,06%
28	TF-IDF	REMOVE	TRUE	TRUE	65,39%	67,01%	65,90%	59,45%	61,91%	59,26%	54,60%	55,12%	54,75%	67,43%	40,80%	31,99%
29	TF-IDF	NORMAL.	FALSE	FALSE	66,03%	67,65%	66,51%	60,57%	62,12%	59,33%	49,79%	50,26%	49,96%	56,32%	58,62%	57,08%
30	TF-IDF	NORMAL.	TRUE	FALSE	65,15%	66,60%	65,56%	60,13%	62,26%	59,75%	51,31%	51,75%	51,30%	55,22%	55,46%	53,42%
31	TF-IDF	NORMAL.	FALSE	TRUE	66,55%	68,18%	67,04%	60,69%	62,74%	60,09%	51,54%	52,02%	51,74%	**56,97%**	**59,66%**	**58,06%**
32	TF-IDF	NORMAL.	TRUE	TRUE	65,68%	67,27%	66,17%	60,17%	62,54%	60,06%	53,60%	54,21%	53,75%	55,93%	57,09%	55,15%

TABLA 5.6.1 - RESULTADOS DE LA BÚSQUEDA DEL CLASIFICADOR LÍNEA DE BASE

No hay ninguna duda de que el algoritmo ganador es la máquina de vectores de soporte y, el que peores resultados ha obtenido en términos generales, k vecinos más cercanos. No obstante, en uno de los experimentos llega a obtener un mayor rendimiento que los árboles de decisión, quedando por tanto en la tercera posición. Este gráfico de barras muestra los mejores valores para los cuatro algoritmos probados:

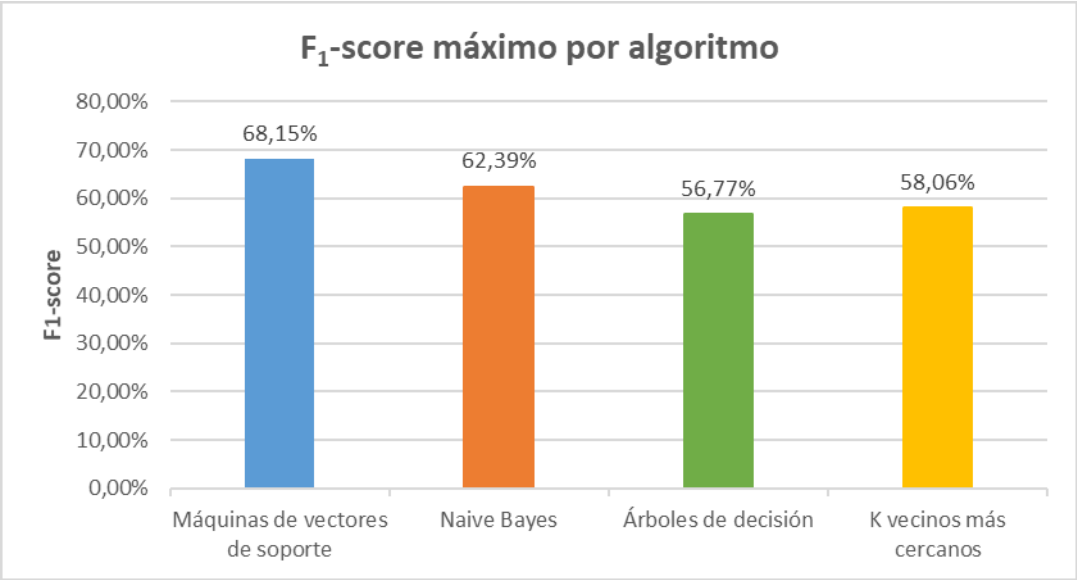

FIGURA 5.6.1 - F1-SCORE MÁXIMO POR ALGORITMO

Estos resultados vienen a confirmar lo indicado en múltiples estudios y en donde se afirma que las máquinas de vectores son los clasificadores con un mejor rendimiento para el análisis de sentimientos en documentos cortos, seguido de cerca por la familia de algoritmos Naive Bayes (Pang et al., 2002), (Go et al. 2009), (Martínez-Cámara, E. et al., 2015), (Martínez Cámara, 2016) y otros. Es también importante destacar el resultado aceptable del algoritmo de árboles de decisión, mucho mejor de lo esperado, ya que no se utiliza de manera habitual en la tarea de categorización de textos debido a que no suele ofrece buenos resultados. Un estudio sobre el rendimiento del algoritmo C4.5[35] en la clasificación de textos se puede consultar en (Gabrilovich & Markovitch, 2004). Finalmente, aunque los primeros test de k vecinos más cercanos presentan unos valores muy bajos, éstos mejoran de manera notable durante las ponderaciones TF y TF-IDF. Existen trabajos como (Tan, S., & Zhang, J., 2008) en donde se puede ver que k vecinos más cercanos rinde casi igual que Naive Bayes y máquinas de soporte de vectores en determinadas condiciones, aunque siempre por detrás de ambos.

[35] Algoritmo para la generación de árboles de decisión usado en tareas de clasificación.

En la siguiente gráfica se puede observar con mayor claridad la superioridad de unos algoritmos sobre los otros:

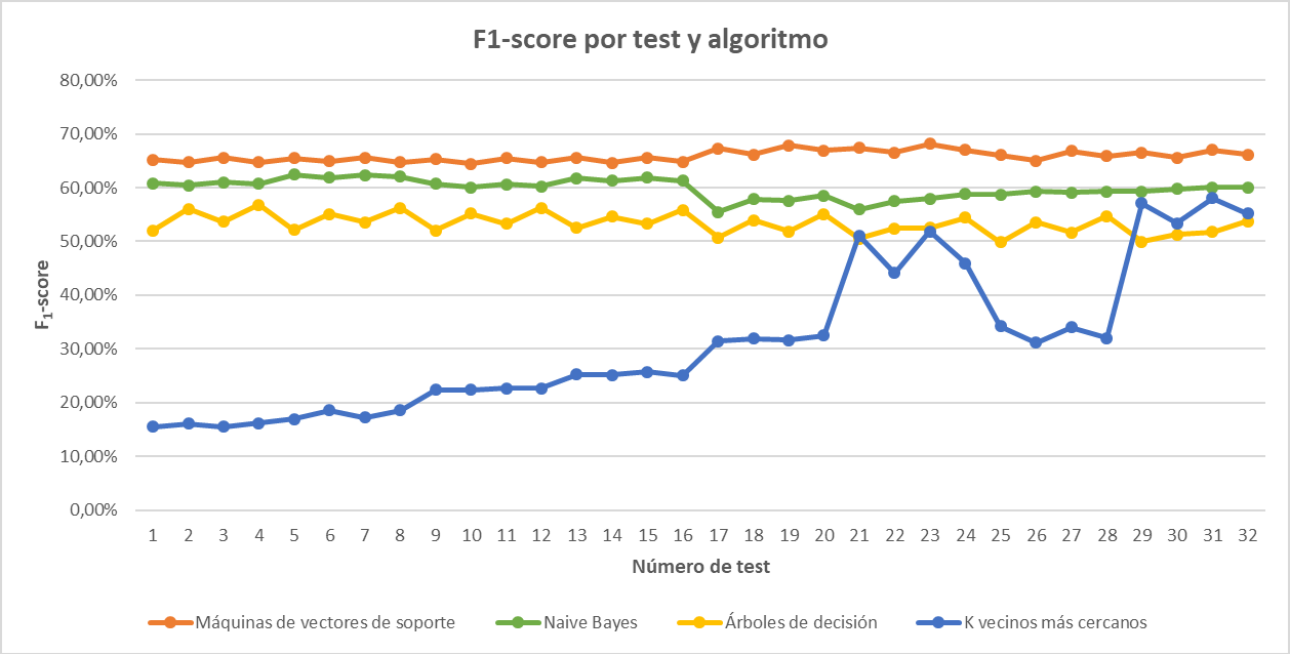

FIGURA 5.6.2 - F1-SCORE POR TEST Y ALGORITMO

La progresión de cada línea permite apreciar en qué grado aumenta o disminuye el rendimiento de cada algoritmo la regla de ponderación aplicada. Los test 1-8 usan una ponderación BTO, en el intervalo 9-16 la ponderación es TO, las pruebas 17-24 tienen ponderación TF y entre los números 25-32 la ponderación es TF-IDF.

Si nos fijamos con detalle, en el caso de las máquinas de vectores de soporte y k vecinos más cercanos, el rendimiento aumenta en las ponderaciones normalizadas TF y TF-IDF, justo al contrario que en el caso de Naive Bayes y los árboles de decisión, en donde el rendimiento disminuye, aunque, en Naive Bayes se observa una mejoría a medida que avanza el número de test. Estas variaciones son más fáciles de percibir en el siguiente gráfico de barras, en donde se muestra el valor medio de la medida F_1-score ponderada para los diferentes test agrupados por algoritmo y método de ponderación:

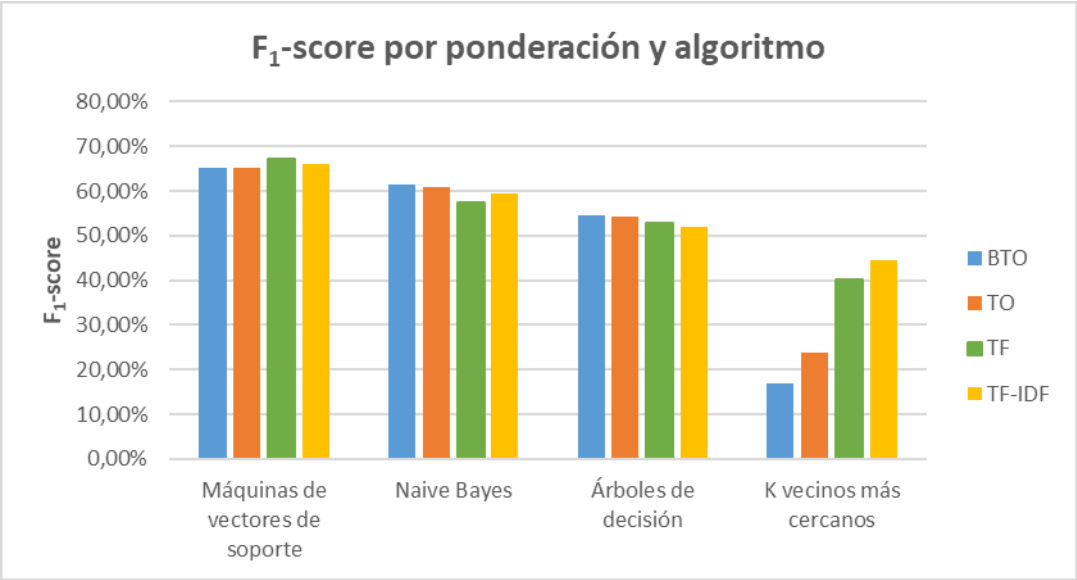

FIGURA 5.6.3 - F1-SCORE MEDIO POR PONDERACIÓN Y ALGORITMO

Aquí se percibe que Naive Bayes y los árboles de decisión obtienen mejores resultados con ponderaciones BTO y TO. El caso es el inverso para las máquinas de vectores de soporte, con el mejor resultado en la ponderación TF, y k vecinos más cercanos, con una mejoría espectacular para los métodos TF y TF-IDF, aunque todavía insuficiente para alcanzar a sus rivales.

Volviendo al gráfico principal, cada línea muestra una forma de dientes de sierra, con subidas y bajadas que son especialmente notables para el caso de los árboles de decisión. Estos picos se deben a las técnicas de reducción de las características. En concreto, los test con un número par son exactamente iguales los test justamente anteriores, pero habiendo aplicado el método de reducción por *stopwords*. El mayor beneficiado de esta técnica es sin duda el algoritmo de árboles de decisión. Es muy llamativa la forma montañosa que acompañan a la gráfica en todo momento y en donde los picos se sitúan justo en los test pares y los valles, en los impares. El caso contrario, aunque de forma mucho más sensible, lo tenemos en las máquinas de vectores de soporte. Aquí, la reducción de características por medio de la eliminación de las *stopwords* merma el rendimiento del clasificador en todas las pruebas. Por último, los dos algoritmos restantes no presentan el mismo comportamiento en todos los test y la eliminación de las *stopwords* afecta de manera positiva en algunos métodos de ponderación y negativa en otros. En concreto, Naive Bayes disminuye su rendimiento de manera sensible en las ponderaciones BTO y TO, al revés que para TF y TF-IDF. El caso de k vecinos más cercanos es justamente el contrario, pero hay que destacar que en TF y TF-IDF su uso implica una pérdida notable de rendimiento.

El impacto de la reducción de características mediante *stemming* es un poco más complicado de visualizar en la gráfica de líneas. En cada grupo de cuatro test, el tercero en la secuencia tiene la misma

configuración que el primero, pero con la técnica de *stemming*. El cuarto test tiene reducción de características por *stopwords* y *stemming* simultáneamente. Todos los algoritmos mejoran su rendimiento con *stemming*, pero de nuevo son los árboles de decisión quienes aprovechan las técnicas de reducción de características de manera especial.

El siguiente gráfico de barras muestra los valores alcanzados para cada algoritmo y las cuatro combinaciones de reducción de las características:

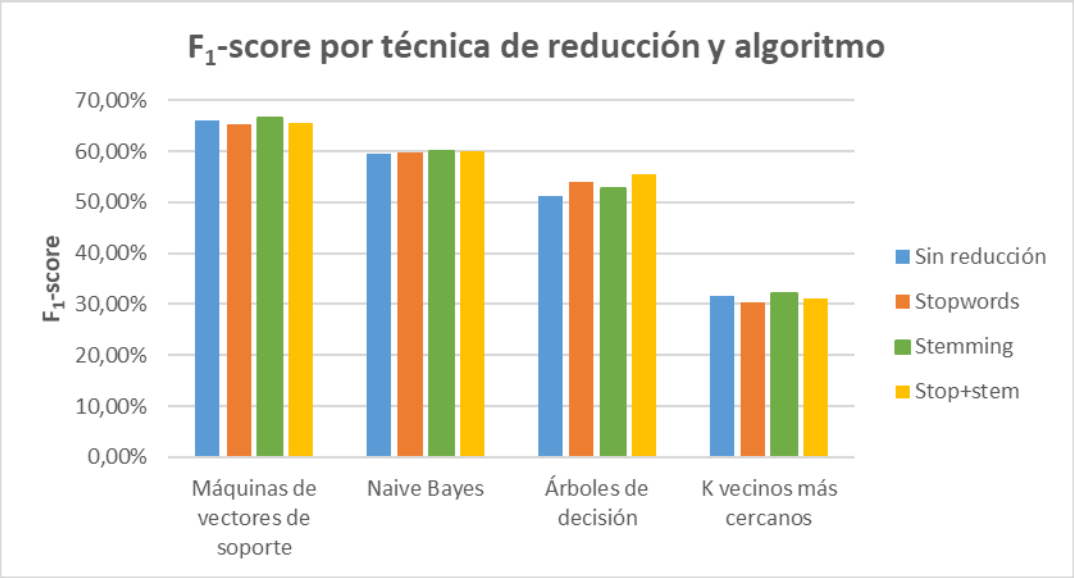

FIGURA 5.6.4 - F1-SCORE MEDIO POR TÉCNICA DE REDUCCIÓN Y ALGORITMO

Es sencillo apreciar que la reducción por *stopwords* no beneficia al rendimiento de los algoritmos de máquinas de vectores de soporte ni a k vecinos más cercanos, justo lo contrario que el *stemming*, que sí ayuda, aunque sólo sensiblemente. El algoritmo que mayor beneficio obtiene de la reducción de características es sin duda los árboles de decisión, con una mejora de más de cuatro puntos porcentuales.

La última técnica a analizar es la influencia del borrado o normalización de los elementos de Twitter, menciones, hashtags y direcciones URL insertadas en los mensajes. En el gráfico de líneas se borran estos elementos en grupos de cuatro test consecutivos y se normalizan en los cuatro siguientes. A simple vista, no parece que una técnica u otra afecte de manera sustancia al rendimiento, con la excepción de k vecinos más cercanos. En este último caso y de manera muy evidente, en las ponderaciones TF (test 17-24) y TF-IDF (test 25-32) se observa una ganancia muy marcada para la normalización con respecto al borrado (los cuatro test finales de cada uno de los intervalos anteriores). En el siguiente gráfico de barras se puede ver con mayor claridad la influencia de estas técnicas en los resultados:

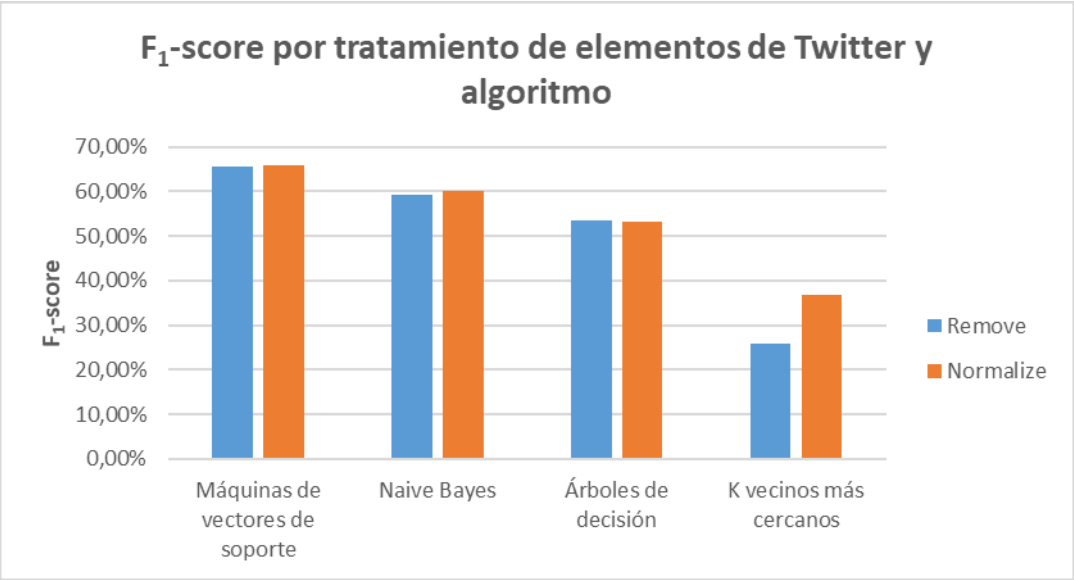

FIGURA 5.6.5 - F1-SCORE MEDIO POR TRATAMIENTO DE ELEMENTO DE TWITTER Y ALGORITMO

En las máquinas de vectores de soporte y Naive Bayes existe una leve mejoría en la normalización de los elementos de Twitter, justo al revés que en los árboles de decisión. Pero tal y como se indicó en anteriores párrafos, quién se ve más afectado por la normalización es k vecinos más cercanos, mejorando su rendimiento medio en más de once puntos porcentuales.

De todo lo dicho anteriormente, podemos extraer las siguientes conclusiones:

- El algoritmo máquinas de vectores de soporte obtiene los mejores resultados en todos los modelos construidos. Le siguen en rendimiento medio Naive Bayes, los árboles de decisión y, ya a mucha distancia, k vecinos más cercanos.

- A pesar de que el resultado medio de k vecinos más cercanos es muy bajo en relación al de los otros algoritmos, en uno de los experimentos obtiene la tercera posición, por lo que su rendimiento es aceptable bajo determinadas condiciones.

- Las ponderaciones basadas en el número de ocurrencias de las características, como BT y TO, benefician a Naive Bayes y a los árboles de decisión. En cambio, las máquinas de vectores de soporte y k vecinos más cercanos mejoran sus resultados con ponderaciones normalizadas como TF y TF-IDF.

- La reducción de características mediante la técnica de *stemming* ayuda a aumentar los valores de todos los algoritmos sensiblemente, aunque de manera especial para los árboles de decisión, en más de un punto porcentual.

- La reducción de características mediante la eliminación de las *stopwords* penaliza a las máquinas de soporte de vectores y a k vecinos más cercanos, pero mejora ligeramente a Naive Bayes y, sobre todo, a los árboles de decisión con casi tres puntos porcentuales.

- Los árboles de decisión es el algoritmo en donde la reducción de características funciona mejor, permitiendo aumentar sus resultados en más de cuatro puntos de media.

- No hay mucha diferencia entre eliminar los elementos de Twitter y normalizarlos para las máquinas de soporte de vectores y Naive Bayes, aunque la normalización mejora algo sus rendimientos. Al contrario, y en relación al punto anterior, los árboles de decisión se benefician cuando se eliminan los elementos, es decir, cuando el número de características se reduce. En el caso de k vecinos más cercanos, la normalización aumenta de manera espectacular el rendimiento del clasificador.

De entre todos los sistemas construidos, el mejor es la máquina de vectores de soporte con los elementos de Twitter normalizados, una reducción de las características basado en *stemming* y ponderación TF. Esta configuración es la misma que la elegida como la más eficiente[36] en (Martínez Cámara, 2016) y será el clasificador línea de base seleccionado para el paso siguiente.

5.7 Mejora del clasificador línea de base

Partiendo del modelo del apartado anterior, trataremos de mejorar sus resultados mediante un conjunto de características adicionales más allá de los *unigramas* de los mensajes del corpus. Estas características serán confeccionadas con varios métodos extraídos de distintos trabajos de investigación consultados para este TFM:

- **Símbolos de sentimiento**: como ya fue señalado en diversas ocasiones, los usuarios de Twitter acostumbran a añadir emoticonos o emojis a sus mensajes y éstos son un indicador inequívoco de la polaridad de sentimiento de sus palabras (Wang, H., & Castanon, J. A., 2015). Para comprobar si estos símbolos ayudan al proceso de clasificación de los textos, se incluirá como característica el número de símbolos positivos, negativos y neutros que contiene cada mensaje. Para ello, se han creado tres listas distintas de símbolos, una para cada tipo de sentimiento, a partir de un listado de emoticonos [37] y otro de emojis [38] extraídos de Internet.

- **Onomatopeya de la risa:** al igual que el caso anterior, la risa escrita en todas sus variantes suele ser indicador de polaridad de sentimiento. Para incluir esa técnica en el clasificador, en cada mensaje se

[36] En el estudio de (Martínez Cámara, 2016) sólo se contempla la eliminación de los elementos de Twitter, nunca su normalización
[37] https://en.wikipedia.org/wiki/List_of_emoticons
[38] https://unicode.org/emoji/charts/emoji-list.html

normalizará la onomatopeya de la risa siguiendo el mismo método que en el preprocesamiento del corpus y se asociará el número de ocurrencias a cada tweet.

- **Lexicón de palabras con sentimiento:** un recurso clásico y muy usado en la clasificación de textos por sentimiento son los diccionarios de palabras etiquetadas con su polaridad, normalmente positiva o negativa. Así, para cada texto se buscará el número de palabras positivas y negativas que contiene y esta información se añadirá como característica asociada a cada mensaje. Para nuestro modelo, se ha confeccionado un lexicón mediante la unión de otros dos ya existentes: iSOL[39] (Molina-González et al., 2013) y Spanish Sentimient Lexicon[40] (Perez-Rosas et al., 2012). A este listado de palabras se le ha aplicado un proceso de *stemming* resultando un conjunto de 1181 términos positivos y 2555 negativos.

  La connotación de una palabra no sólo depende de ella misma sino también de aquellas otras con las que tiene algún tipo de relación sintáctica dentro de la frase u oración a la que pertenecen. Esta cuestión se aprecia de manera clara en el fenómeno de la negación: aunque el adjetivo "agradable" es una palabra positiva, en la frase "el día no es agradable" ese mismo adjetivo adquiere una connotación negativa debido a la partícula "no". Por tanto, el sistema de recuento de palabras por sentimiento tendrá un mecanismo de detención de la negación. Así, si una palabra de un texto se encuentra en el diccionario y está precedida por una partícula negativa (por ejemplo, "no" o "ni") dentro de una ventana de tres términos, la polaridad de la palabra se invertirá (Anta et al., 2013).

- **Categoría gramatical de palabras (Parts of speech):** este método asocia a cada término su categoría gramatical por medio de un conjunto de etiquetas denominado Dependencias Universales[41] (del inglés, Universal Dependencies o UD). Algunas de estas etiquetas son ADJ para adjetivos, VERB para verbos, NOUN para sustantivos, INTJ para interjecciones y así hasta 17 categorías gramaticales diferentes. En varios estudios se afirma que esta información es útil a la hora de clasificar los mensajes por su sentimiento (Pak, A., & Paroubek, P., 2010), (Xia et al., 2011). Para comprobarlo, en cada texto se calculará el número de etiquetas de cada tipo y se añadirá como característica de cada mensaje. La clasificación gramatical se hará mediante la utilidad Stanford CoreNLP[42].

- **Patrones de categorías gramaticales de palabras (Parts of speech patterns):** algunos estudios como (Gamallo, P., & Garcia, M., 2014) sugieren que determinados patrones gramaticales son más comunes en los mensajes de un tipo de sentimiento que en otros y, por tanto, pueden ser útiles para

[39] http://timm.ujaen.es/recursos/isol/
[40] http://web.eecs.umich.edu/~mihalcea/downloads.html#SPANISH_SENT_LEXICONS
[41] http://universaldependencies.org/u/pos/index.html
[42] https://stanfordnlp.github.io/CoreNLP/

ayudar al proceso de clasificación. Estos patrones son secuencias de etiquetas como ADJ-NOUN, VERB-NOUN, VERB-PRON-NOUN y algunas otras. Teniendo en cuenta lo anterior, esta técnica trata de capturar dichos patrones y añadir las palabras que los forman como una característica más al modelo.

- **Elementos de Twitter:** con estas características se pretende saber si la aparición de determinados elementos como menciones, hashtags o URLs, son indicativos del sentimiento de los mensajes a los que pertenecen. De esta forma, para cada tweet se contará el número de cada tipo y el resultado se incluirá como característica.

- **Patrones de caracteres:** por último, se desea conocer si ciertos patrones alfanuméricos sugieren polaridad de sentimiento. Para cada texto se obtendrá el número de símbolos de admiración que poseen, el número de letras mayúsculas y el número máximo de letras repetidas consecutivas que contienen. La teoría sobre la que se sustenta este método es que cuanto mayor sea cualquiera de estos números, mayor será la intensidad de mensaje a transmitir y, por tanto, más alejado se encontrará un mensaje de las categorías NEU y NONE. Para evitar que las menciones o las URLs influyan en el resultado, estos elementos serán eliminados de los mensajes previamente.

Para probar la efectividad de las técnicas presentadas, se entrenará el clasificador línea de base con el subconjunto de mensajes del corpus global utilizado en el apartado anterior y se testeará sobre el otro subconjunto restante de mensajes.

En la siguiente tabla se pueden ver los resultados obtenidos mediante la aplicación de las técnicas anteriores y la mejora con respecto al modelo línea de base:

	Precision	Recall	F_1-Score	Accuracy	Improved F_1-Score	Improved Accuracy
Línea de base	68,57%	69,87%	68,47%	69,87%		
Símbolos de sentimiento	68,57%	69,91%	68,51%	69,91%	0,04%	0,04%
Parts of speech	68,93%	70,23%	68,85%	70,23%	0,38%	0,36%
Parts of speech patterns	67,15%	68,46%	67,37%	68,46%	-1,10%	-1,41%
Parts of speech patterns (sólo patrones)	69,34%	70,31%	69,23%	70,31%	0,76%	0,44%
Lexicón de sentimiento	68,95%	70,44%	68,93%	70,44%	0,46%	0,56%
Onomatopeya de la risa	68,58%	69,90%	68,50%	69,90%	0,03%	0,02%
Patrones de caracteres	68,63%	69,93%	68,54%	69,93%	0,07%	0,06%
Elementos de Twitter	68,50%	69,82%	68,41%	69,82%	-0,06%	-0,05%
Elementos de Twitter (twitter_elements=NONE)	69,18%	70,22%	69,14%	70,22%	0,67%	0,35%

Unión sólo positivos	69,68%	70,87%	69,69%	70,87%	1,22%	1,00%
Unión sólo positivos, C=0.5, chi2=7000	70,09%	71,19%	69,81%	71,19%	1,34%	1,32%

TABLA 5.7.1 - RESULTADOS DE TÉCNICAS DE MEJORA DEL CLASIFICADOR LÍNEA DE BASE

Como se puede observar, ninguna de las técnicas implementadas aumenta de manera espectacular los resultados del clasificador línea de base. En su lugar, los incrementos son menores destacando por encima de todos los métodos basados en etiquetas gramaticales, el diccionario de palabras de sentimiento y los elementos de Twitter. La razón por la que no se obtienen mejoras más elevadas puede residir en el hecho de que el corpus de entrenamiento contiene mucha información, más de la que el resto de técnicas pueden aportar. De hecho, si el corpus global lo dividimos en dos subconjuntos, uno de entrenamiento con sólo el 5% de mensajes y otro de test con el 95% restante, se constata que la contribución de las características adicionales aumenta, obteniendo en este caso una mejora del 2,5% tanto para el valor-F_1 como para la exactitud.

Más allá de esta cuestión, es evidente que las etiquetas gramaticales y sus relaciones contienen información importante que ayuda de manera especial en la clasificación de los textos. El recuento del número de etiquetas de cada tipo que tiene cada mensaje aumenta el valor-F_1 en un 0,38%, lo que se traduce en una subida del 0,36% en términos de exactitud. Esta mejora es incluso más llamativa si lo que tomamos como característica son los patrones gramaticales. En este punto es necesario aclarar que lo que al principio se iba a usar como característica eran las palabras de los textos, pero este enfoque disminuía el valor-F_1 un 1,10%, un resultado realmente malo. Después de ejecutar varios experimentos se pudo observar que, si en lugar de las palabras se añadían los propios patrones como características, el resultado aumentaba el 0,76% (método “Parts of speech patterns (sólo patrones)”). En cualquier caso, la clase gramatical es un elemento de gran riqueza que puede ser determinante a la hora de construir clasificadores robustos y eficaces.

Es aceptado que los diccionarios de palabras catalogadas por sentimiento ayudan de manera importante en la tarea de clasificación de textos. En nuestro caso, el rendimiento del clasificador línea de base aumentó en un 0,46% en términos de valor-F_1 y el 0,56% respecto a la exactitud, la mayor de las subidas de este indicador. Contar con un buen lexicón puede ayudar de manera sustancial a la hora de clasificar los textos, así como implementar un sistema más eficiente para el tratamiento de la negación, cuestión de vital importancia en el análisis de sentimientos.

Respecto a la técnica que cuenta el número de menciones, hashtags y URLs de cada mensaje, el resultado inicial no era muy prometedor ya que el rendimiento disminuía en un 0,06% para el valor-F_1. De

nuevo, mediante una serie de experimentos, se descubrió que, si estos elementos no se normalizaban durante la fase de preprocesamiento, la técnica conseguía aumentar en un 0,67% el valor-F_1. La explicación a este comportamiento es que estas características tienen un mayor peso si se tratan por separado que si se añaden como *unigramas* junto con el resto de las palabras del corpus. Este método es el que tiene como nombre "Elementos de Twitter (twitter_elements=NONE)" en la tabla anterior.

El resto de técnicas contribuyen a mejorar el rendimiento del clasificador línea de base, pero de manera muy tenue. Quizás sea posible obtener mejores resultados mediante un estudio más profundo sobre qué características aportan más información al modelo y, de esta forma, ajustar los pesos de cada técnica para ampliar el margen de mejora.

Para finalizar, se ha construido un modelo con todas las técnicas que ofrecen valores de mejora positivos y, después de diversos experimentos, se ha concluido que con un valor C=0.5 para el algoritmo de clasificación y una reducción de características mediante la técnica *k highest scores*[43] con k=7000 y la función de selección X^2 (chi cuadrado), se obtiene el mejor de los resultados: 69,81% para el valor-F_1 y 71,19% para la exactitud. Nótese el incremento de los resultados para cada clase en comparación con el clasificador línea de base, en especial en los mensajes etiquetados como de clase NEU:

Línea de base				
	Precision	**Recall**	**F_1-Score**	**Support**
N	66,43%	71,96%	69,09%	14109
NEU	38,19%	2,35%	4,43%	2339
NONE	65,62%	67,11%	66,36%	16681
P	76,44%	79,02%	77,71%	19087
total	68,57%	69,87%	68,47%	52216
accuracy	69,87%			

Unión sólo positivos, C=0.5, chi2=7000				
	Precision	**Recall**	**F_1-Score**	**Support**
N	66,84%	74,71%	70,56%	14109
NEU	44,26%	4,62%	8,36%	2339
NONE	69,69%	65,78%	67,68%	16681
P	76,01%	81,47%	78,65%	19087
total	70,09%	71,19%	69,81%	52216
accuracy	71,19%			

TABLA 5.7.2 - MEJORA DEL CLASIFICADOR LÍNEA DE BASE POR CLASES

[43] https://machinelearningmastery.com/feature-selection-machine-learning-python/

5.8 Conclusiones y valoración final

En este apartado hemos visto cómo crear un sistema de clasificación de mensajes extraídos de la red social Twitter de manera automática y en varias categorías de sentimiento. El algoritmo de máquinas de vectores de soporte es, sin ninguna duda y corroborando lo manifestado en multitud de trabajos de investigación, el mejor de los clasificadores para la realización de este tipo de tareas. En este caso, hemos probado su eficacia mediante el entrenamiento con *unigramas*, debido también a que los mejores resultados se suelen obtener con esta clase de características. Las técnicas de ponderación normalizada contribuyen a aumentar la eficacia del modelo, mientras que la reducción de características mediante *stemming*, aunque no de manera espectacular, también aporta una mejoría en el rendimiento global.

Complementar la colección de características con otras distintas a los propios términos de los mensajes es fundamental a la hora de diseñar un sistema de clasificación de textos, sobre todo en el caso de que el corpus de entrenamiento no sea muy extenso. Los métodos basados en etiquetas gramaticales se presentan como los más prometedores y que mejores incrementos de rendimiento pueden aportar, pero para ello es esencial que las palabras se clasifiquen de la manera más acertada posible. Los diccionarios de sentimiento son otra de las herramientas que mejores resultados suelen ofrecer, pero su eficacia depende del número de términos que contengan y de otros mecanismos complementarios como el tratamiento de la negación. El recuento de las menciones, hashtags y URLs de los mensajes parece que puede ayudar en el proceso de clasificación, cuestión muy interesante que merece de un estudio en mayor profundidad. Por último, métodos como la captura de emoticonos, emojis y la risa escrita, aunque no hemos visto en este caso una mejoría significativa, suelen ser recursos con una gran carga de polaridad de sentimiento, por lo que son técnicas recomendables en los sistemas de clasificación automatizada.

El porcentaje de exactitud del modelo final es del 71,79%. Pudiese parecer que este porcentaje no es lo suficientemente alto como para afirmar que el sistema posee un buen rendimiento, pero nada más lejos de la realidad. En el análisis de sentimientos se considera que un sistema presenta un buen nivel de precisión cuando alcanza un valor del 70% de acierto[44] [45]. Esto se debe a que ni siquiera las personas somos capaces de ponernos de acuerdo hasta en un 30% de las veces a la hora de clasificar los textos en categorías. De hecho, un vistazo rápido al corpus usado durante este TFM confirma que muchos de sus mensajes no tendrían la misma categoría si fuésemos nosotros quienes tuviésemos que clasificarlos. De esta forma, si tuviésemos un modelo que acertase un 100% de las veces, aún estaríamos en desacuerdo con su predicción en el 30% de las ocasiones.

[44] https://brnrd.me/posts/social-sentiment-sentiment-analysis
[45] https://en.wikipedia.org/wiki/Sentiment_analysis

En el análisis de los resultados también hay que tener en cuenta que el corpus utilizado tiene una categoría que no suele ser estándar: la categoría NEU. Normalmente, los modelos de clasificación usan tres categorías posibles: positiva, negativa y neutra. Esta categoría neutra se refiere a mensajes que no expresan ningún tipo de sentimiento, es decir, es nuestra categoría NONE. Esta clase NEU sí posee sentimiento y se aplica a mensajes en los que sus palabras expresan un sentimiento neutro (AGREEMENT) o a aquellos en donde se mezclan sentimientos positivos y negativos al mismo tiempo (DISAGREEMENT). Esta categoría es realmente difícil de modelar y la situación empeora debido a que nuestro corpus sólo cuenta con un 9% de mensajes de este tipo, es decir, hay una fuerte desproporción entre el número de mensajes de cada clase. Si hacemos la prueba de eliminar la clase NEU y sólo nos quedamos con los mensajes etiquetados como POS, NEG y NONE, el resultado sube hasta un magnífico 74,34% de exactitud (74,22% para el valor-F_1).

Para terminar y como muestra del buen rendimiento del clasificador obtenido, se han ejecutado las pruebas de la Tarea 1 (3 levels + NONE) del Taller de Análisis de Sentimientos en la SEPLN de 2012 [46] (TASS 2012). El resultado ha sido del 67,78% de exactitud lo que representa el cuarto puesto dentro de una lista de veinte participantes.

Run Id	Group	Precision[47]
pol-elhuyar-1-3l	Elhuyar Fundazioa	71.12%
pol-l2f-1-3l	L2F - INESC	69.05%
pol-l2f-3-3l	L2F - INESC	69.04%
TFM	**José Carlos Sobrino**	**67,78%**
pol-l2f-2-3l	L2F - INESC	67.63%
pol-atrilla-1-3l	La Salle - Universitat Ramon Llull	61.95%
pol-sinai-4-3l	SINAI - UJAEN	60.63%
pol-uned1-1-3l	LSI UNED (tamara & jorge)	59.03%
pol-uned1-2-3l	LSI UNED (tamara & jorge)	58.77%
pol-uned2-1-3l	LSI UNED 2 (angel, juan manuel & ana)	50.08%
pol-imdea-1-3l	IMDEA	45.95%
pol-uned2-2-3l	LSI UNED 2 (angel, juan manuel & ana)	43.61%
pol-tudelft-5y6-3l	TUDELFT	43.61%
pol-uned2-4-3l	LSI UNED 2 (angel, juan manuel & ana)	41.20%
pol-uned2-3-3l	LSI UNED 2 (angel, juan manuel & ana)	40.43%

[46] http://www.sepln.org/workshops/tass/2012/tasks.php
[47] Aunque la fuente original usa el término “Precision” para referirse a los resultados, en realidad quiere decir “Exactitud” (o Accuracy, en inglés)

pol-tudelft-3y4-3l	TUDELFT	40.27%
pol-tudelft-1y2-3l	TUDELFT	38.45%
pol-uma-1-3l	UMA	37.61%
pol-sinai-2-3l	SINAI - UJAEN	35.83%
pol-sinai-1-3l	SINAI - UJAEN	35.58%
pol-sinai-3-3l	SINAI – UJAEN	35.11%

TABLA 5.8.1 - RESULTADOS TAREA 1 DE TASS 2012

6. Aplicaciones del análisis de sentimientos en Twitter

Ahora que ya hemos visto cómo crear un sistema automatizado para medir el sentimiento de los usuarios de Twitter por medio de sus mensajes, pensemos de qué manera puede ser utilizado. En este apartado hablaremos de las posibles aplicaciones de este tipo de clasificadores en la vida real y de qué manera se pueden beneficiar personas, empresas y todo tipo de organizaciones mediante su uso.

6.1 Introducción

Mediante la monitorización de Twitter y la captura y análisis de los mensajes que en él se publican, se podría obtener la misma información que en la actualidad se consigue mediante encuestas de opinión, pero de manera más inmediata, menos costosa y con actualizaciones en tiempo real. Es cierto que para elaborar estas encuestas se seleccionan grupos de sujetos que representan de manera fiel a la población en estudio y esto no es sencillo de implementar sobre Twitter. Pero si no nos importa sacrificar parte de la precisión debido a que lo que en realidad nos interesa es conocer una tendencia, el análisis de sentimientos es una herramienta magnífica para obtener esa información con poco esfuerzo y en el menor tiempo posible

El éxito de estas tareas depende principalmente de dos cuestiones. Por una parte, de poder detectar en qué mensajes se encuentra la información necesaria para nuestra tarea. Estos mensajes pueden ser capturados por medio de la búsqueda de ciertas palabras, hashtags o menciones y teniendo en cuenta todos los usuarios de la red o sólo algunos especializados en un tema concreto. Por otra parte, es fundamental saber manejar el llamado *opinion* spamming (Jindal, N., & Liu, B, 2008), es decir, poder averiguar qué mensajes contienen información preparada para desvirtuar la opinión general sobre un tema determinado y así evitar su contabilización.

6.2 Aplicaciones en empresas y negocios

Una de las entidades qué mayor beneficio puede obtener de la evaluación del sentimiento expresado en redes sociales como Twitter son sin duda las empresas de bienes y servicios. Para cualquier compañía de este tipo es fundamental poder conocer qué piensan sobre ellas sus potenciales clientes y poder detectar lo antes posible cualquier deterioro en su imagen corporativa y en la manera en la que el público percibe su marca.

Precisamente, la reputación de marca es uno de los valores principales que más cuidan las empresas, ya que de ella depende que sus productos y servicios sean percibidos como positivos por los consumidores. Una

determinada firma podría conocer qué percepción tienen de ella los usuarios de Twitter. Para esto se extraerían aquellos mensajes que contengan el nombre de la empresa que se quiere evaluar y a continuación, mediante el sistema desarrollado en el apartado anterior, se clasificarían por su sentimiento, obteniendo así una medida del afecto o rechazo que tiene la gente sobre la marca. De la misma forma, se podría aplicar este sistema para saber en qué situación se encuentran sus competidores directos y, si fuese necesario, diseñar algún tipo de campaña de marketing para tratar de mejorar su reputación.

En relación a esto último, medir el retorno de la inversión de las campañas de marketing sería posible mediante la monitorización de la red social durante el tiempo que dure su ejecución. El sistema extraería por intervalos de tiempo, horas o días, los tweets en donde se haga referencia a la organización, producto o servicio, los clasificaría por su sentimiento y, por medio de una proyección de esta información en gráficos de líneas, se podría observar si dicha campaña está realmente funcionando.

El análisis de sentimientos en Twitter también permite saber qué aceptación tiene entre las personas un producto o servicio concreto, el cual puede ser desde un bien de consumo tangible hasta una película, programa o serie de televisión. Los hashtags con los que suelen acompañarse los lanzamientos de nuevos productos o las cuentas de usuario de los servicios de atención al cliente aglutinan gran cantidad de mensajes en donde los consumidores vierten sus opiniones sobre los productos en cuestión. Un sistema de clasificación de sentimientos podría conocer en tiempo real qué grado de aprobación les produce dicho producto o servicio y, en el caso de que el sentimiento fuese negativo, ofrece la posibilidad de actuar lo antes posible para solventar los problemas que pudiesen estar sucediendo.

La gestión de una crisis también es otro escenario del que puede beneficiarse el análisis de sentimiento en Twitter. Una mala decisión no detectada a tiempo puede suponer un daño irreparable para una organización. Ejemplos de este tipo de crisis pueden ser huelgas de trabajadores, filtraciones de información comprometida a la opinión pública, decisiones empresariales que impliquen destrucción del medio ambiente, etc. Cuando las empresas se enfrentan a este tipo de situaciones, se puede saber si la crisis está siendo gestionada de manera correcta analizando el sentimiento de los mensajes de Twitter en tiempo real y, en el caso de que no sea así, tratar de rectificar para no dañar más la imagen de la empresa.

Por último, no cabe duda de que la publicidad personalizada es otra de las áreas beneficiadas de este tipo de técnicas. Mediante el análisis de los mensajes publicados por usuarios concretos se puede saber si poseen sentimientos positivos o negativos sobre determinados bienes y servicios. Los usuarios se podrán clasificar en base a la afinidad o simpatía que muestren por ciertos productos o marcas y, de esta forma, mostrarles publicidad acorde a sus gustos personales. Con este tipo de técnicas, también es posible detectar

su disconformidad con algún servicio que tengan contratado con una empresa rival y ofrecerle otros productos alternativos, ya que son potenciales consumidores de tales servicios.

6.3 Aplicaciones en política y gestión de gobierno

El mundo de la política es uno de los que más y mejor beneficio está obteniendo del análisis de sentimientos en redes sociales. Es conocido su uso en las campañas electorales de Obama en 2012 y Trump en 2016 para la presidencia de EEUU, y con el que ambos obtuvieron la victoria en sus respectivas citas, por lo que es fácil imaginarse que muchos grupos políticos y Gobiernos de todo el mundo están prestando una atención especial a este tipo de herramientas ocupando un lugar destacado dentro de sus campañas de propaganda. Sólo hay que pensar por un momento en la gran cantidad de dinero que se invierte en la elaboración de encuestas electorales para saber que este es un recurso de gran potencial para conocer el sentir de la población sobre Gobiernos, líderes políticos, partidos y en la toma de decisiones de Estado.

El uso más simple y básico es conocer la evolución de la aceptación que los ciudadanos tienen de un Gobierno a lo largo del tiempo y ver cómo sus decisiones afectan en este sentido. Por ejemplo, la búsqueda en Twitter de palabras clave como “Gobierno”, “Rajoy” o “PP” permitirán detectar los potenciales mensajes que expresen opiniones sobre el Gobierno de España. Estos mensajes serán posteriormente clasificados mediante el modelo diseñado en el apartado anterior y la información resultante analizada para saber cómo ha ido variando el sentir de los ciudadanos con respecto al Gobierno a lo largo del tiempo. Los cambios bruscos de sentimiento, a nivel global o por regiones geográficas, estarían provocados por las decisiones tomadas durante la legislatura, así como la gestión de determinadas crisis o la publicación de ciertas informaciones. Por tanto, conocer estos datos en tiempo real puede ayudar de manera sustancial a los Gobiernos a la hora de tomar las decisiones más adecuadas en cada momento y no dañar su imagen y credibilidad.

Una aplicación de gran interés es poder analizar la evolución del voto indeciso durante una campaña electoral. Conocer la afinidad política de un usuario de Twitter es, en muchos casos, bastante sencillo. Basta con realizar una serie de consultas con los nombres de los partidos políticos sobre la lista de los mensajes que el usuario ha publicado y determinar cuál presenta mayores valoraciones positivas. En base a esto, se puede seleccionar un grupo concreto de usuarios que no manifiesten ninguna inclinación importante y ver si ésta va apareciendo a medida que avance la campaña electoral. Si así ocurre, esto significa que uno de los candidatos está acertando en su manera de enfocar la campaña por lo que esos mensajes son los que movilizarán al electorado indeciso a la hora de decantarse por una opción un otra. Con esta herramienta incluso sería posible predecir los resultados finales.

Por último, el análisis de la polaridad de sentimiento permite saber qué cargos públicos, como ministros o responsables políticos, gozan de una mayor aceptación por parte de la población, pudiendo detectar a tiempo qué personas y sus decisiones causan rechazo entre los votantes y actuar en consecuencia antes de que el daño sea irreparable. La búsqueda y clasificación de los mensajes que contengan ciertos nombres puede ayudar a la hora de detectar este tipo de problemas y tendencias.

6.4 Aplicaciones en financias y economía

Ya se ha probado en múltiples ocasiones que es posible predecir la evolución en bolsa de un determinado activo a partir del sentimiento que diversos usuarios especializados muestran en sus publicaciones de Twitter. Los cambios en las tendencias suelen estar precedidos por alteraciones más o menos bruscas en el sentimiento expresado en mensajes de la red social. En base a esta premisa, monitorizar la actividad de ciertos usuarios puede ayudar a la hora de conocer cómo evolucionará el valor de una empresa u organización a corto y medio plazo. De nuevo, la búsqueda, captura y clasificación de los tweets que contengan ciertos términos puede ser usado para invertir de manera acertada en un valor u otro.

Dentro del área de la economía, una aplicación de este tipo de clasificadores es la predicción de la evolución del coste de bienes esenciales como la energía eléctrica, los carburantes, la vivienda, e incluso los alimentos. De manera análoga al caso de las variaciones en bolsa, la idea es capturar aquellos mensajes que hagan referencia al precio de la luz, de la gasolina, de los alquileres e hipotecas, alimentos y materias primas para su elaboración. Estos mensajes, que pueden provenir de usuarios anónimos o de organismos oficiales como Ministerios, bancos u otras entidades, deben ser monitorizados durante cierto tiempo y, en base a sus cambios de sentimiento, extraer las conclusiones que permitan predecir tendencias y cambios en los precios

Ya para terminar, quizás también sea posible predecir la entrada de la economía en un período de recesión mediante el análisis de tweets en donde se incluyan palabras como “paro”, “precios”, “despidos”, “trabajo” y otras similares. Al igual que ocurre en las conversaciones cara a cara, las redes sociales se llenan de mensajes negativos cuando la economía no va bien para sus usuarios por lo que es sencillo imaginar que existirán variaciones de sentimiento perceptibles para los clasificadores automatizados y en relación a este tipo de temas.

6.5 Aplicaciones en temas generales y opinión pública

Después de leer los apartados anteriores, es fácil advertir que las herramientas de clasificación de sentimientos se pueden usar para saber qué es lo que opinan otras personas sobre prácticamente cualquier tema. La cuestión clave es saber qué mensajes son los que se deben capturar, someterlos al proceso de clasificación y analizar la tendencia del sentimiento de manera acertada.

Algunas consultas podrían estar relacionadas con la satisfacción que los usuarios muestran sobre ciertos productos de consumo, como televisores, ordenadores, teléfonos móviles o cualquier cosa que se pueda comprar. Estas consultas también podrían ser sobre los servicios de restaurantes, hoteles, aerolíneas o cualquier tipo de negocio. Obras de teatro, exposiciones, museos, películas, libros, conciertos, personajes públicos, videojuegos, series de televisión, destinos turísticos… No hay límites en cuanto a los temas sobre los que podamos conocer qué opina la gente en general o alguna persona en particular. Como se dijo anteriormente, lo importante es saber cuáles son los mensajes que contienen la información que nos interesa y, una vez detectados, hacer uso del sistema de clasificación de sentimientos desarrollado.

7. Conclusiones y cierre del TFM

En el capítulo final de este TFM se expondrán algunas ideas y conclusiones sobre los diversos temas en él tratados y sobre su realización. Por simplificación, no se hará referencia a las conclusiones obtenidas a partir del estudio práctico puesto que éstas se encuentran ya detalladas en el apartado correspondiente. En esta sección hablaremos de los avances del PLN, de lo que ha supuesto la aparición del análisis de sentimientos para la sociedad y de cómo se presenta el futuro para este tipo de sistemas. Para terminar, haremos una reflexión a modo de autoevaluación sobre el desarrollo del TFM.

7.1 Conclusiones y futuro del análisis de sentimientos

Entre el Experimento de Georgetown-IBM en el año 1954 y el lanzamiento del servicio Google Translate en 2006 han transcurrido poco más de 50 años. En términos generales, probablemente no haya nadie que piense que medio siglo es un período de tiempo corto; pero si nos paramos un segundo a analizar el colosal salto que existe entre ambas tecnologías, quizás 50 años nos parezca un tiempo insignificante en relación al hito conseguido. En ese tiempo hemos pasado de un sistema de traducción entre dos idiomas concretos y en un contexto muy limitado a contar con un servicio con capacidad para traducir textos de cualquier tipo entre más de 100 idiomas diferentes y, aunque no pueden sustituir a los traductores humanos si es necesaria una alta precisión, es más que suficiente para las tareas del común de los mortales. Atrás quedaron aquellos sistemas conversacionales tan rudimentarios como ELIZA, algoritmos que generaban una falsa ilusión de inteligencia, pero que se les veía el cartón y las costuras a medida que la conversación avanzaba y ésta se volvía falsa y simulada. 50 años después, sistemas tan evolucionados como Cortana o Siri cuentan con capacidad para llevar a cabo profundos análisis semánticos y pragmáticos con el objetivo de comprender las peticiones de sus usuarios y el contexto en el que éstas se producen, generando, ahora sí, una percepción de conversación real y auténtica entre el humano y la máquina. En sólo 50 años, el PLN ha experimentado una evolución asombrosa e inimaginable, difícil de adivinar hace medio siglo y, a pesar de que nos hallamos sumergidos en la etapa tecnológica más fructífera nunca vivida por la humanidad, seguramente todavía no sepamos hasta dónde seremos capaces de llegar en esta área de desarrollo e investigación.

En contraste con el PLN, la historia del análisis de sentimientos es todavía muy corta, apenas 15 años. En este breve período de tiempo hemos asistido al nacimiento de un fenómeno imparable que avanza implacable a toda velocidad y que ha revolucionado por completo a la sociedad, a sus gentes y sus relaciones. Internet se ha convertido en el escaparate público en el que millones de personas exhiben sus opiniones y sentimientos en cada instante. Redes sociales, sitios de compras online, blogs de todo tipo y tema, foros de debate; lugares en donde expresamos lo que nos gusta y lo que nos desagrada. Una ventana a nuestros deseos y necesidades que abrimos en cualquier momento y desde cualquier lugar. Es su

inmediatez y facilidad lo que hace que cada vez publiquemos más sobre nosotros mismos, de un modo espontáneo, tanto que empresas y organizaciones ya se han dado cuenta del gran valor que posee toda esta información en bruto. La extracción y clasificación a gran escala del sentimiento de los millones de textos que cada día se escriben en Internet será sin duda una de las áreas de mayor actividad tecnológica en el futuro más cercano.

Saber qué es lo que opina y siente cada persona en cada momento es una idea que suena casi como el tener superpoderes. Y en realidad es así. Es un arma potente e irresistible que permitirá a las empresas conocer qué productos son los que los consumidores quieren, cuáles son los que les desagradan y, sobre todo, qué es lo que necesitan en cada momento. Y no sólo es poderosa para empresas. Organizaciones no relacionadas con la venta de productos y servicios también ven en el análisis de sentimientos una gran oportunidad para obtener ventajas frente a sus competidores. Gobiernos y partidos políticos son otros de los grupos que mayor interés muestran sobre este campo y es un hecho conocido que utilizan estas herramientas para saber qué es lo que piensan de ellos y de sus decisiones sus potenciales votantes. El análisis de sentimientos permitirá mantener a la población sometida a un sondeo de opinión permanente y en donde los cambios en dichas opiniones se percibirán y obtendrán en tiempo real.

En este TFM hemos podido conocer las bases teóricas en las que se fundamenta el análisis de sentimientos y de qué diferentes maneras es posible crear un sistema automatizado para evaluar textos y clasificarlos en base a su polaridad. Así mismo, se ha mostrado con un ejemplo práctico la construcción de un clasificador basado en un sistema de aprendizaje supervisado, una de las formas más extendidas y efectivas de construir este tipo de sistemas. Pero a pesar de los buenos resultados, estas soluciones evolucionarán hacia desarrollos más ambiciosos y eficaces, en donde no sólo las palabras escritas serán usadas para evaluar los sentimientos. La información no textual que acompaña a los mensajes y publicaciones posee un gran valor que puede ayudar en el análisis de la polaridad. Vídeos, imágenes, animaciones y otros contenidos multimedia son muy habituales en redes sociales y pueden facilitar enormemente la tarea de clasificación de los sentimientos. Las páginas que se insertan en los mensajes en forma de URLs también contienen información importante que no debe ser desechada. Y, por supuesto, los *likes*, retweets, las respuestas a mensajes y otras maneras de interactuar permiten conocer mucho más de lo que sólo percibimos en el texto escrito.

En el futuro, problemas clásicos del análisis de sentimientos y del PLN como la detención del sarcasmo y de la ironía, la ambigüedad o la dependencia del contexto, se irán mejorando hasta alcanzar, quizás no una solución definitiva, pero sí tolerable. Sin embargo, al mismo tiempo empezarán a aparecer con fuerza otros problemas para los que habrá que buscar soluciones. Bajo mi punto de vista, uno de los más preocupantes es el spam de opiniones, mensajes y publicaciones que tratan de manipular el sentimiento global que las

personas tienen sobre un producto, un servicio o sobre cualquier tema general. En el caso de las redes sociales como Twitter, es una práctica cada vez más habitual que varias personas publiquen mensajes de manera sincronizada y organizada con determinadas ideas para influir en el sentimiento del resto de usuarios. También, cada vez sospechamos más de la autenticidad de ciertas reseñas publicadas en sitios web de restaurantes o de compras online. Tener un control sobre este tipo de comportamientos es fundamental para que el análisis de sentimientos pueda ser considerado una herramienta de verdadero valor. Por tanto, no hay duda de que la lucha contra el spam de opinión será una de las batallas a las que asistiremos próximamente.

El nivel de detalle con el que ahora se miden los sentimientos también cambiará en un futuro próximo. Atrás quedará la escala tan básica de lo positivo, negativo y neutro. Los nuevos sistemas de análisis podrán saber cómo se sienten las personas y darán el salto desde los sentimientos hasta las emociones. Miedo, alegría, tristeza, sorpresa, ira o confianza serán algunos de los estados de ánimo que los clasificadores sabrán distinguir a partir de nuestras palabras escritas. Y no sólo eso. Estos sistemas analizarán nuestra voz y nuestras imágenes mediante dispositivos móviles, ordenadores y otros equipos domésticos que ya conviven entre nosotros y forman parte de nuestra vida cotidiana. Toda esta información permitirá construir sistemas totalmente personalizados, adaptados a nuestros gustos, criterios y a nuestra forma de pensar. La personalización será tal que dejaremos atrás las cifras de precisión del 70% de las que hemos hablamos y que son virtualmente imposibles de mejorar para los clasificadores actuales. Los modelos nos conocerán tan bien que sabrán sin ninguna duda si para nosotros, como individuos concretos, algo es positivo o negativo. No hay duda de que el análisis de sentimientos del futuro nos abrirá las puertas de un mundo fascinante e inquietante a partes iguales.

7.2 Reflexiones acerca del desarrollo del TFM

No me gustaría terminar este TFM sin ofrecer una reflexión personal sobre lo que ha supuesto para mí su realización, así como una crítica sincera sobre el resultado final.

En primer lugar, creo que es justo decir que todos los objetivos marcados al comienzo de este TFM han sido conseguidos en mayor o menor medida. Los dos principales, la exposición de los fundamentos del análisis de sentimientos y la construcción de un sistema automatizado para la clasificación de mensajes de Twitter, fueron desarrollados en los apartados 2-4 y 5 respectivamente. Además, en éstos están todos los temas marcados como subobjetivos, a excepción de los ejemplos de uso del análisis de sentimientos, que se encuentran en el apartado 6. Respecto a este objetivo, es necesario decir que su alcance fue alterado durante el desarrollo del TFM. La idea original era extraer tweets reales mediante la API de Twitter y clasificarlos con el sistema del apartado anterior. A medida que el trabajo iba avanzando se hizo evidente

que no sería posible cumplirlo debido a varios motivos. En primer lugar, el tiempo disponible no lo permitiría. Esta tarea no consistía sólo en extraer y clasificar, sino que los temas de los mensajes tenían que ser seleccionados con cierto cuidado para mostrar con claridad las distintas opiniones de los usuarios, se tendrían que preparar para poder ser clasificados después, sería necesario crear algún programa para poder trabajar con la API y el clasificador al mismo tiempo...en definitiva, demasiadas tareas para tan poco tiempo. Por otra parte, Twitter limita mucho la información que se puede obtener con su API de manera gratuita. Por ejemplo, sólo permite acceder a mensajes de los últimos 6-9 días. Además, bloquea ciertas funciones de búsqueda, no garantiza fidelidad en la información que devuelve, posee restricciones en el número de peticiones que se pueden hacer al día, etc. Debido a todas estas razones, se tomó la decisión de no incluir ejemplos prácticos y se le comunicó al tutor, restando 20 horas de dedicación al apartado 6 y sumándolas al 5.

A pesar de que todos los objetivos han sido conseguidos, una cuestión con la que no me encuentro completamente satisfecho es con la revisión ofrecida del estado del arte. Es cierto que a lo largo de la memoria se han presentado diversos estudios que, bajo mi punto de vista, son los más relevantes debido a cuentan con un gran número de citas en buscadores especializados como Google Scholar; pero aun así me hubiese gustado poder explicar con detalle muchas más publicaciones, sus semejanzas y diferencias y ofrecer una comparativa con mayor profundidad. Lamentablemente, la cantidad de trabajos es inmensa y elaborar un estudio de ese tipo requeriría tanto tiempo que sería un TFM en sí mismo.

Para terminar las reflexiones sobre los objetivos, otra sensación que tengo es que la memoria contiene mucha más teoría que práctica. Si bien es cierto, para la práctica es necesaria antes la teoría, pero quizás algunas partes podrían haber sido reducidas o incluso eliminadas. Es el caso algunas cuestiones sobre el PLN, de las tareas del análisis de sentimientos o las medidas para evaluar los clasificadores. Quizás hubiese sido mejor haber hecho algunas referencias externas y dedicar ese tiempo a otras cuestiones.

Respecto a la planificación comprometida y exceptuando el cambio de alcance comentado, he podido cumplir los plazos de entrega tal y como aparecen en el diagrama de Gantt del primer capítulo. Sin embargo, sí que debo destacar que el número de horas necesarias para su desarrollo ha estado muy por encima de las 300. Las razones de este desfase son dos: el desconocimiento del tema elegido y la dificultad para encontrar información clara y organizada. Respecto a la primera cuestión, la realidad es que partía del cero absoluto en prácticamente cualquiera de las dimensiones del TFM. He necesitado mucho tiempo para poder ir avanzando y entendiendo cada uno de los conceptos de los que consta esta memoria. Por otra parte, otro problema es que la principal fuente de información se encuentra en los estudios publicados por universidades. Es cierto que cada vez aparecen más libros sobre el análisis de sentimientos y que existen diversas páginas web que tratan este tema, pero no es tan popular y accesible como lo pueden ser las bases

de datos o el desarrollo de software. Encontrar información organizada y que, además, como ocurre en los estudios consultados, no se contradiga entre ella, no ha sido sencillo. Quizás la metodología seguida ha sido demasiado simple y debería haber usado un enfoque más estratégico. También reconozco que no he sabido valorar con acierto el tiempo y el esfuerzo de todo el trabajo acordado.

Y ya para finalizar, hay varias cuestiones que no he podido probar y que me hubiese gustado, por lo que quedarían pendientes como futuros trabajos. Una de ellas es el uso de algoritmos de regresión para la clasificación de los mensajes. Mediante este tipo de algoritmos seguramente podría haber construido un sistema para saber en qué grado un texto es positivo o negativo, abriendo así la posibilidad de medir la intensidad de los sentimientos. También me hubiese gustado experimentar con métodos “ensemble”, es decir, sistemas compuestos por múltiples algoritmos de aprendizaje automático y que permiten construir clasificadores más robustos. Las redes neuronales y el aprendizaje profundo, tan de moda en estos momentos, son otros de los campos que querría haber tratado, aunque hubiese sido sólo desde un punto de vista teórico. Los sistemas de clasificación no supervisados y el uso de recursos como WordNet o SentiNet sobre los que se apoyan muchas de las soluciones de este tipo, son otras de las líneas de investigación que se han quedado sin tiempo para ser exploradas. Por último, pienso que con los conocimientos ahora adquiridos en materia de análisis de datos podría haber construido un sistema de clasificación mejor, pero estoy seguro de que en un futuro cercano podré dedicarme a este campo profesionalmente y hacer uso de todo lo aprendido durante este tiempo.

Glosario

Bibliografía

Alessia, D., Ferri, F., Grifoni, P., & Guzzo, T. (2015). Approaches, tools and applications for sentiment analysis implementation. International Journal of Computer Applications, 125(3).

Anta, A. F., Chiroque, L. N., Morere, P., & Santos, A. (2013). Sentiment analysis and topic detection of Spanish tweets: A comparative study of of NLP techniques. Procesamiento del lenguaje natural, 50, 45-52.

Biagioni, R. (2016). The SenticNet sentiment lexicon: Exploring semantic richness in multi-word concepts (Vol. 4). Springer.

Cambria, E., & Hussain, A. (2012). Sentic computing: Techniques, tools, and applications (Vol. 2). Springer Science & Business Media.

Carbonell, J. G. (1979). Subjective Understanding: Computer Models of Belief Systems (No. RR-150). YALE UNIV NEW HAVEN CONN DEPT OF COMPUTER SCIENCE.

Dave, K., Lawrence, S., & Pennock, D. M. (2003, May). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In Proceedings of the 12th international conference on World Wide Web (pp. 519-528). ACM.

Esuli, A., & Sebastiani, F. (2007). SentiWordNet: a high-coverage lexical resource for opinion mining. Evaluation, 1-26.

Gabrilovich, E., & Markovitch, S. (2004, July). Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4. 5. In Proceedings of the twenty-first international conference on Machine learning (p. 41). ACM.

Gamallo, P., & Garcia, M. (2014). Citius: A naive-bayes strategy for sentiment analysis on english tweets. In Proceedings of the 8th international Workshop on Semantic Evaluation (SemEval 2014) (pp. 171-175).

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1(12).

Hardeniya, N. (2015). NLTK essentials. Packt Publishing Ltd.

Hatzivassiloglou, V., & McKeown, K. R. (1997, July). Predicting the semantic orientation of adjectives. In Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics (pp. 174-181). Association for Computational Linguistics.

Hu, M., & Liu, B. (2004, August). Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 168-177). ACM.

Kumar, A., & Sebastian, T. M. (2012). Sentiment analysis on twitter. IJCSI International Journal of Computer Science Issues, 9(3), 372-378.

Kumar, A., & Teeja, M. S. (2012). Sentiment analysis: A perspective on its past, present and future. International Journal of Intelligent Systems and Applications, 4(10), 1.

Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, 5(1), 1-167.

Martínez Cámara, E. (2016). Análisis de opiniones en español. Tesis Doctoral. Universidad de Jaén.

Martínez-Cámara, E., Martín-Valdivia, M. T., Urena-López, L. A., & Montejo-Ráez, A. R. (2014). Sentiment analysis in Twitter. Natural Language Engineering, 20(1), 1-28.

Martínez-Cámara, E., Martín-Valdivia, M. T., Ureña-López, L. A., & Mitkov, R. (2015). Polarity classification for Spanish tweets using the COST corpus. Journal of Information Science, 41(3), 263-272.

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, 5(4), 1093-1113.

Molina-González, M. D., Martínez-Cámara, E., Martín-Valdivia, M. T., & Perea-Ortega, J. M. (2013). Semantic orientation for polarity classification in Spanish reviews. Expert Systems with Applications, 40(18), 7250-7257.

O'Reilly, T. (2004). The architecture of participation.

Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In LREc (Vol. 10, No. 2010).

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1–2), 1-135.

Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.

Perez-Rosas, V., Banea, C., & Mihalcea, R. (2012, May). Learning Sentiment Lexicons in Spanish. In LREC (Vol. 12, p. 73).

Perkins, J. (2014). Python 3 text processing with NLTK 3 cookbook. Packt Publishing Ltd.

Read, J. (2005, June). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In Proceedings of the ACL student research workshop (pp. 43-48). Association for Computational Linguistics.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. Computational linguistics, 37(2), 267-307.

Tan, S., & Zhang, J. (2008). An empirical study of sentiment analysis for chinese documents. Expert Systems with applications, 34(4), 2622-2629.

Turney, P. D. (2002, July). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th annual meeting on association for computational linguistics (pp. 417-424). Association for Computational Linguistics.

Vapnik, V. N., & Kotz, S. (1982). Estimation of dependences based on empirical data (Vol. 40). New York: Springer-Verlag.

Villena-Román, Julio, Lana-Serrano, Sara, Martínez-Cámara, Eugenio, González-Cristobal, José Carlos. 2013. Revista de Procesamiento del Lenguaje Natural, 50, pp 37-44. http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/4657.

Wang, H., & Castanon, J. A. (2015, October). Sentiment expression via emoticons on social media. In Big Data (Big Data), 2015 IEEE International Conference on (pp. 2404-2408). IEEE.

Wilks, Y., & Bien, J. (1983). Beliefs, points of view, and multiple environments. Cognitive Science, 7(2), 95-119.

Xia, R., Zong, C., & Li, S. (2011). Ensemble of feature sets and classification algorithms for sentiment classification. Information Sciences, 181(6), 1138-1152.

Zafra, J., María, S., Martínez Cámara, E., Valdivia, M., Teresa, M., & Ureña López, L. A. (2014). SINAI-ESMA: An unsupervised approach for Sentiment Analysis in Twitter. In Proc. of the TASS workshop at SEPLN (pp. 16-19).

Jindal, N., & Liu, B. (2008, February). Opinion spam and analysis. In Proceedings of the 2008 International Conference on Web Search and Data Mining (pp. 219-230). ACM.

Enlaces consultados

Nota: Esta lista contiene los enlaces web más relevantes que han sido consultados durante el desarrollo de este TFM y, aunque no posible determinar con exactitud su fecha de consulta, ésta se encuentra dentro del período de tiempo transcurrido entre las fechas 02/2018 y 06/2018.

Análisis de sentimiento - Wikipedia
https://es.wikipedia.org/wiki/Análisis_de_sentimiento

Sentiment Analysis - Wikipedia
https://en.m.wikipedia.org/wiki/Sentiment_analysis

The importance of Neutral Class in Sentiment Analysis - Datumbox
http://blog.datumbox.com/the-importance-of-neutral-class-in-sentiment-analysis/

Text Classification and Sentiment Analysis - Ahmet Taspinar
http://ataspinar.com/2015/11/16/text-classification-and-sentiment-analysis/

Multiclass classification - Wikipedia
https://en.m.wikipedia.org/wiki/Multiclass_classification

Twitter Sentiment Analysis Training Corpus (Dataset) - Thinknook
http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22

Precision and recall - Wikipedia
https://en.m.wikipedia.org/wiki/Precision_and_recall

Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog
http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures

Text Classification with NLTK and Scikit-Learn - Libelliinar
https://bbengfort.github.io/tutorials/2016/05/19/text-classification-nltk-sckit-learn.html

Automate Machine Learning Workflows with Pipelines in Python and scikit-learn - Machine Learning Mastery
https://machinelearningmastery.com/automate-machine-learning-workflows-pipelines-python-scikit-learn

Python NLP - NLTK and scikit-learn
http://billchambers.me/tutorials/2015/01/14/python-nlp-cheatsheet-nltk-scikit-learn.html

Machine Learning & Sentiment Analysis: Text Classification using Python & NLTK - Mukesh Chapagain Blog
http://blog.chapagain.com.np/machine-learning-sentiment-analysis-text-classification-using-python-nltk

Python Programming Tutorials
https://pythonprogramming.net/sklearn-scikit-learn-nltk-tutorial

Dive Into NLTK, Part IV: Stemming and Lemmatization - Text Mining Online
http://textminingonline.com/dive-into-nltk-part-iv-stemming-and-lemmatization

Text Classification for Sentiment Analysis – Precision and Recall - StreamHacker
https://streamhacker.com/2010/05/17/text-classification-sentiment-analysis-precision-recall

Como hacer Análisis de Sentimiento en español - Pybonacci
https://www.pybonacci.org/2015/11/24/como-hacer-analisis-de-sentimiento-en-espanol-2

Essentials of Machine Learning Algorithms (with Python and R Codes)
https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms

A Tour of Machine Learning Algorithms - Machine learning mastery
https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms

Sentiment analysis in Spanish
http://blog.manugarri.com/sentiment-analysis-in-spanish

Using Pipelines and FeatureUnions in scikit-learn - Michelle Fullwood
https://michelleful.github.io/code-blog/2015/06/20/pipelines

Sentiment Analysis: Concept, Analysis, and Applications - DZone AI
https://dzone.com/articles/sentiment-analysis-concept-analysis-and-applicatio

Classification Accuracy is Not Enough: More Performance Measures You Can Use - Machine Learning Mastery
https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use

Big Data & Data Science Blog: Los 2 tipos de aprendizaje en Machine Learning: supervisado y no supervisado
http://data-speaks.luca-d3.com/2017/11/que-algoritmo-elegir-en-ml-aprendizaje.html?m=1

Clasificador bayesiano ingenuo - Wikipedia, la enciclopedia libre
https://es.wikipedia.org/wiki/Clasificador_bayesiano_ingenuo

Machine Learning: Text feature extraction (tf-idf) – Part II - Terra Incognita
http://blog.christianperone.com/2011/10/machine-learning-text-feature-extraction-tf-idf-part-ii

Procesamiento de lenguajes naturales - Wikipedia
https://es.wikipedia.org/wiki/Procesamiento_de_lenguajes_naturales

Análisis de sentimiento, ¿qué es, cómo funciona y para qué sirve?
http://www.itelligent.es/es/analisis-de-sentimiento

Experimento de Georgetown - Wikipedia
https://es.wikipedia.org/wiki/Experimento_de_Georgetown

ELIZA - Wikipedia
https://es.wikipedia.org/wiki/ELIZA

The future of natural language processing
http://www.expertsystem.com/future-natural-language-processing

Overfitting in Machine Learning: What It Is and How to Prevent It
https://elitedatascience.com/overfitting-in-machine-learning

Definición de norma Euclídea - Wikipedia
https://es.wikipedia.org/wiki/Norma_vectorial#Definición_de_norma_euclídea

TF-IDF - Wikipedia
https://es.wikipedia.org/wiki/Tf-idf

Validación cruzada - Wikipedia
https://es.wikipedia.org/wiki/Validación_cruzada

Universal POS tags - Universal Dependencies
http://universaldependencies.org/u/pos/index.html

On Social Sentiment and Sentiment Analysis – brnrd.me
https://brnrd.me/posts/social-sentiment-sentiment-analysis

The Future of Sentiment Analysis – Terrific data
http://terrificdata.com/2017/04/07/future-sentiment-analysis

Publicidad por estados de ánimo – Puro marketing
https://www.puromarketing.com/9/28734/publicidad-estados-animo-inquietante-futuro-vuelta-esquina.html

SVM (Support Vector Machine) — Theory – Machine Learning 101 – Medium
https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

Anexos

La memoria de este TFM se complementa con dos recursos adicionales: el código de fuente de los modelos construidos en la parte práctica y una hoja de cálculo con los resultados obtenidos a partir de las pruebas realizadas. Ambos recursos se explican con mayor detalle en esta sección.

Anexo I – Código fuente

Todas las pruebas presentadas en el apartado 5 de esta memoria han sido escritas en lenguaje de programación Python v3.6[48] y con ayuda de las siguientes librerías gratuitas y de código abierto:

- **NLTK v3.2.5[49]:** facilita la realización de múltiples tareas para el procesamiento de lenguajes naturales.

- **Scikit-Learn v0.19.1[50]:** librería que implementa diversos algoritmos de aprendizaje automático y ofrece recursos para el análisis y la minería de datos.

- **Stanford Part-Of-Speech Tagger v3.8**[51]: utilidad para el análisis sintáctico de textos y que, entre otras cosas, permite clasificar las palabras en base a categoría gramatical.

- **Pandas v0.22[52]:** conjunto de utilidades para la manipulación y análisis de datos mediante lenguaje de programación Python

El proyecto está adjunto como anexo a este documento de memoria y además puede ser consultado en el siguiente proyecto alojado en GitHub:

`https://github.com/jcsobrino/TFM-Analisis_sentimientos_Twitter-UOC`

La estructura que sigue el proyecto es la siguiente:

- **Carpeta raíz:** contiene dos archivos con la implementación del estudio comparativo para la obtención del clasificador línea de base (*baseline_model.py*) y para la mejora del rendimiento del clasificador anterior (*improved_model.py*).

[48] https://www.python.org/
[49] http://www.nltk.org/
[50] http://scikit-learn.org/stable/
[51] https://nlp.stanford.edu/software/tagger.shtml
[52] https://pandas.pydata.org/

- **Carpeta "datasets":** contiene todos los corpus de mensajes que han sido utilizados durante la parte práctica del TFM. Por una parte, la carpeta "tass" tiene los archivos originales con del General Corpus, Politics Corpus e International TASS Corpus. Por otro lado, la carpeta raíz contiene el corpus global creado a partir de la unión de las tres colecciones anteriores. Este último tiene dos presentaciones adicionales señaladas con los sufijos 5 y 30. Este valor hace referencia al porcentaje de mensajes del corpus global que ha sido usado como entrenamiento en varias pruebas realizadas. Por último, los archivos *estandard_dataset.csv* contienen los mensajes usados para la tarea 1 de TASS 2012.

- **Carpeta "extractor":** cada fichero Python es uno de los extractores de características usados en el apartado de mejora del clasificador línea de base.

- **Carpeta "lexicon":** contiene los diccionarios de palabras clasificadas por sentimiento y que se usan durante la extracción de características mediante lexicón. Esta carpeta también incluye los diccionarios originales con los que se ha creado el de la práctica: iSOL y Spanish Sentiment Lexicon.

- **Carpeta "sentiment-symbols":** son los listados de emoticones y emojis clasificados por polaridad de sentimiento y usados durante la extracción de características por símbolos de sentimiento.

- **Carpeta "stanford-postagger":** contiene el archivo .JAR con la utilidad que permite la clasificación de las palabras de los mensajes con su categoría gramatical. Esta utilidad está preparada para hacer uso de dos sistemas de clasificación: EAGLES, mediante el archivo *spanish-distsim.tagger* y Universal Dependencies si se usa el fichero *spanish-ud.tagger*. Este último sistema es el que ha sido utilizado para la realización de esta práctica. Debido a que el tiempo de etiquetado es muy elevado, se ha precalculado esta información y guardado en el archivo *precalculated_ud_postags.pkl* para todos los mensajes del corpus global.

- **Carpeta "util":** tiene tres clases Python de utilidades. *Preprocesor.py* implementa los métodos descritos para el preprocesamiento de los mensajes antes de la fase de entrenamiento, *DatasetHelper.py* que es una utilidad para manipular los diferentes corpus de esta práctica y *PartsOfSpeechHelper.py* gestiona la comunicación con el etiquetador Stanford Part-Of-Speech Tagger.

A continuación, se muestra el código fuentes de cada uno de los archivos anteriores:

/analisis-sentimientos/baseline_model.py

```python
import csv

import pandas as pd
from nltk import TweetTokenizer
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import GridSearchCV, StratifiedKFold
from sklearn.naive_bayes import MultinomialNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.pipeline import Pipeline
from sklearn.svm import LinearSVC
from sklearn.tree import DecisionTreeClassifier

from util.DatasetHelper import DatasetHelper
from util.Preprocessor import Preprocessor

# global corpus 30% for cross-validation
message, label = DatasetHelper.csv_to_lists("datasets/train_dataset_30.csv")

# spanish stop words
spanish_stopwords = stopwords.words('spanish')

# metrics
scoring = {'accuracy': 'accuracy',
           'precision_macro': 'precision_macro',
           'recall_macro': 'recall_macro',
           'f1_macro': 'f1_macro',
           'precision_micro': 'precision_micro',
           'recall_micro': 'recall_micro',
           'f1_micro': 'f1_micro',
           'precision_weighted': 'precision_weighted',
           'recall_weighted': 'recall_weighted',
           'f1_weighted': 'f1_weighted'}

# pipeline
pipeline = Pipeline([('vectorizer', None),
                     ('classifier', None)])

# Tokenizer
tokenizer = TweetTokenizer().tokenize

# feature weights
bow_binary_term_ocurrences = CountVectorizer(binary=True, tokenizer=tokenizer)
bow_absolute_term_ocurrences = CountVectorizer(binary=False, tokenizer=tokenizer)
bow_term_frequency = TfidfVectorizer(use_idf=False, tokenizer=tokenizer)
bow_tfidf = TfidfVectorizer(use_idf=True, tokenizer=tokenizer)

parameters = [{
    'vectorizer': (bow_binary_term_ocurrences,
                   bow_absolute_term_ocurrences,
                   bow_term_frequency,
                   bow_tfidf),
    'vectorizer__preprocessor': (Preprocessor(twitter_features=Preprocessor.REMOVE).preprocess,
                                 Preprocessor(twitter_features=Preprocessor.REMOVE, stemming=True).preprocess,
                                 Preprocessor(twitter_features=Preprocessor.NORMALIZE).preprocess,
                                 Preprocessor(twitter_features=Preprocessor.NORMALIZE, stemming=True).preprocess),
    'vectorizer__stop_words': (None, spanish_stopwords),
    'classifier': (MultinomialNB(), LinearSVC(), DecisionTreeClassifier(), KNeighborsClassifier(n_neighbors=200))
}]

if __name__ == '__main__':
    skf = StratifiedKFold(n_splits=10, shuffle=True)
    grid_search = GridSearchCV(pipeline, param_grid=parameters, n_jobs=-1, cv=skf, verbose=5, scoring=scoring,
                               refit='f1_weighted', return_train_score=False)
    grid_search.fit(message, label)
    print("best_score:", grid_search.best_score_)
    pd.DataFrame(grid_search.cv_results_).to_csv(path_or_buf='baseline.csv',
                                                 quoting=csv.QUOTE_NONNUMERIC)
```

/analisis-sentimientos/improved_model.py

```
from nltk import TweetTokenizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_selection import VarianceThreshold, SelectKBest, mutual_info_classif, chi2, f_classif
from sklearn.metrics import classification_report, accuracy_score
from sklearn.pipeline import Pipeline, FeatureUnion
from sklearn.svm import LinearSVC

from extractors.CharacterExtractor import CharacterExtractor
from extractors.LaughExtractor import LaughExtractor
from extractors.LexiconExtractor import LexiconExtractor
from extractors.PartsOfSpeechExtractor import PartsOfSpeechExtractor
from extractors.PartsOfSpeechPatternExtractor import PartsOfSpeechPatternExtractor
from extractors.SentimentSymbolExtractor import SentimentSymbolExtractor
from extractors.TwitterExtractor import TwitterExtractor
from util.DatasetHelper import DatasetHelper
from util.PartsOfSpeechHelper import PartsOfSpeechHelper
from util.Preprocessor import Preprocessor

# global corpus 30% for training
message_train, label_train = DatasetHelper.csv_to_lists("datasets/train_dataset_30.csv")
message_test, label_test = DatasetHelper.csv_to_lists("datasets/test_dataset_30.csv")

# standard corpus TASS 2012
#message_train, label_train = DatasetHelper.csv_to_lists("datasets/standard_train_dataset.csv")
#message_test, label_test = DatasetHelper.csv_to_lists("datasets/standard_test_dataset.csv")

# Tokenizer
tokenizer = TweetTokenizer().tokenize

# Preprocessor with stemming enabled
preprocessor = Preprocessor(twitter_features=None, stemming=True).preprocess

# Term Frequency
bow_term_frequency = TfidfVectorizer(use_idf=False, tokenizer=tokenizer, preprocessor=preprocessor)

pipeline = Pipeline([
    ('feats', FeatureUnion([
           ('vectorizer', bow_term_frequency),
           ('sentiment_symbol', SentimentSymbolExtractor()),
           ('parts_of_speech', PartsOfSpeechExtractor()),
           ('parts_of_speech_pattern', PartsOfSpeechPatternExtractor()),
           ('lexicon', LexiconExtractor()),
           ('laugh', LaughExtractor()),
           ('character', CharacterExtractor()),
           ('twitter', TwitterExtractor())
    ])),
    ("fs", SelectKBest(score_func=chi2, k=7000)),
    ('classifier', LinearSVC(C=0.5))
])


pipeline.fit(message_train, label_train)

y_prediction = pipeline.predict( message_test )

report = classification_report(label_test, y_prediction, digits=4)

print(report)
print(accuracy_score(label_test, y_prediction))
```

/analisis-sentimientos/extractors/CharacterExtractor.py

```python
import re
from itertools import groupby

from sklearn import preprocessing
from sklearn.base import BaseEstimator, TransformerMixin

from util.Preprocessor import Preprocessor


class CharacterExtractor(BaseEstimator, TransformerMixin):

    def __init__(self):
        pass

    def transform(self, data, y=None):
        result = []

        for tweet in data:
            tweet = Preprocessor.process_twitter_features(tweet, twitter_features=Preprocessor.REMOVE)
            result.append([
                tweet.count('!'),
                self._max_consecutive_equals_characteres(tweet),
                len(re.findall(r'[A-Z]', tweet))
            ])

        return preprocessing.normalize(result)

    def fit(self, df, y=None):
        return self

    def _max_consecutive_equals_characteres(self, text):
        if len(text) == 0:
            return 0

        groups = groupby(text)
        return max(num for char, num in [(char, sum(1 for _ in group)) for char, group in groups])
```

/analisis-sentimientos/extractors/LaughExtractor.py

```python
from sklearn import preprocessing
from sklearn.base import BaseEstimator, TransformerMixin

from util.Preprocessor import Preprocessor


class LaughExtractor(BaseEstimator, TransformerMixin):

    def __init__(self):
        pass

    def transform(self, data, y=None):
        result = []

        for tweet in data:
            tweet = Preprocessor.normalize_laughs(tweet)
            result.append([tweet.count(Preprocessor.LAUGH)])

        return preprocessing.normalize(result)

    def fit(self, df, y=None):
        return self
```

/analisis-sentimientos/extractors/LexiconExtractor.py

```
import io

from nltk import TweetTokenizer
from nltk.util import ngrams
from sklearn import preprocessing
from sklearn.base import BaseEstimator, TransformerMixin

from util.Preprocessor import Preprocessor


class LexiconExtractor(BaseEstimator, TransformerMixin):

    NGRAM_LENGTH = 3
    REVERSE_WORDS = ['no', 'ni', 'tampoc', 'ningun']

    _tokenizer = TweetTokenizer()
    _preprocessor = Preprocessor(twitter_features=Preprocessor.REMOVE, stemming=True)

    def __init__(self):
        self._neg_words = self.file_to_list('lexicon/negative_words.txt')
        self._pos_words = self.file_to_list('lexicon/positive_words.txt')

    def transform(self, data, y=None):
        result = []

        for tweet in data:
            tweet = self._preprocessor.preprocess(tweet)
            result.append(self.count_polarity_words(tweet))

        return preprocessing.normalize(result)

    def count_polarity_words(self, text):
        num_pos_words = 0
        num_neg_words = 0

        list_ngrams = list(ngrams(self._tokenizer.tokenize(text), self.NGRAM_LENGTH,
pad_left=True))
        for ngram in list_ngrams:
            pre_words = ngram[:self.NGRAM_LENGTH-1]
            word = ngram[self.NGRAM_LENGTH-1]

            if word in self._pos_words:
                if any(w in pre_words for w in self.REVERSE_WORDS):
                    num_neg_words += 1
                else:
                    num_pos_words += 1

            elif word in self._neg_words:
                if any(w in pre_words for w in self.REVERSE_WORDS):
                    num_pos_words += 1
                else:
                    num_neg_words += 1

        return [num_pos_words, num_neg_words]

    def fit(self, df, y=None):
        return self

    def file_to_list(self, filename):
        return io.open(filename).read().splitlines()
```

/analisis-sentimientos/extractors/PartsOfSpeechExtractor.py

```
from collections import Counter

from nltk import TweetTokenizer
from sklearn.base import BaseEstimator, TransformerMixin
from sklearn.feature_extraction import DictVectorizer

from util.PartsOfSpeechHelper import PartsOfSpeechHelper


class PartsOfSpeechExtractor(BaseEstimator, TransformerMixin):

    IGNORE_TAGS = ['PUNCT', 'CCONJ']
    _vectorizer = None
    _tokenizer = TweetTokenizer(reduce_len=True)
    _pos_helper = PartsOfSpeechHelper()

    def __init__(self):
        pass

    def transform(self, data, y=None):
        result = []

        for tweet in data:
            result.append(self.pos_tag(tweet))

        if self._vectorizer == None :
            self._vectorizer = DictVectorizer(sparse=False)
            self._vectorizer.fit(result)

        return self._vectorizer.transform(result)

    def pos_tag(self, tweet):
        tokens = self._tokenizer.tokenize(tweet)
        pos_tweet = self._pos_helper.pos_tag(tokens)
        return Counter([t for w,t in pos_tweet if t not in
self.IGNORE_TAGS])
    def fit(self, df, y=None):
        return self
```

/analisis-sentimientos/extractors/PartsOfSpeechPatternExtractor.py

```python
from collections import Counter

from nltk import TweetTokenizer
from sklearn.base import BaseEstimator, TransformerMixin
from sklearn.feature_extraction import DictVectorizer

from util.PartsOfSpeechHelper import PartsOfSpeechHelper
from util.Preprocessor import Preprocessor


class PartsOfSpeechPatternExtractor(BaseEstimator, TransformerMixin):

    POS_PATTERNS = [('NOUN','ADJ'), ('NOUN','NOUN'), ('ADJ','NOUN'), ('VERB','NOUN'),
('AUX','NOUN'),     ('NOUN','PRON','NOUN'), ('VERB','PRON','NOUN'), ('AUX','PRON','NOUN')]
    IGNORE_TAGS = ['PUNCT']

    _vectorizer = None
    _tokenizer = TweetTokenizer(reduce_len=True)
    _processor = Preprocessor(stemming=True)
    _pos_helper = PartsOfSpeechHelper()

    def __init__(self):
        pass

    def transform(self, data, y=None):
        result = []

        for tweet in data:
            result.append(self.get_patterns(tweet))

        if self._vectorizer == None :
            self._vectorizer = DictVectorizer(sparse=False)
            self._vectorizer.fit(result)

        return self._vectorizer.transform(result)

    def get_patterns(self, tweet):
        result = []
        tokens = self._tokenizer.tokenize(tweet)
        pos_tags = self._pos_helper.pos_tag(tokens)
        if len(pos_tags) > 1:
            pos_tags = [p for p in pos_tags if p[1] not in self.IGNORE_TAGS]
            words, tags = zip(*pos_tags)

            for pattern in self.POS_PATTERNS:
                found = self.find_sublist(list(pattern), list(tags))
                for i,j in found:
                    # Added patterns instead of tokens
                    result.append('_'.join(list(pattern)))
                    # result.append(self._processor.preprocess(' '.join(words[i:j])))

        return Counter(result)

    def fit(self, df, y=None):
        return self

    def find_sublist(self, sl, l):
        results = []
        sll = len(sl)
        for ind in (i for i, e in enumerate(l) if e == sl[0]):
            if l[ind:ind + sll] == sl:
                results.append((ind, ind + sll))

        return results
```

/analisis-sentimientos/extractors/SentimentSymbolExtractor.py

```python
import io

from sklearn import preprocessing
from sklearn.base import BaseEstimator, TransformerMixin

from util.Preprocessor import Preprocessor


class SentimentSymbolExtractor(BaseEstimator, TransformerMixin):

    _preprocessor = Preprocessor(twitter_features=Preprocessor.REMOVE)

    def __init__(self):
        self._pos_symbols = self.file_to_list('sentiment-symbols/positive_symbols.txt')
        self._neg_symbols = self.file_to_list('sentiment-symbols/negative_symbols.txt')
        self._neu_symbols = self.file_to_list('sentiment-symbols/neutral_symbols.txt')

    def transform(self, data, y=None):
        result = []

        for tweet in data:
            tweet = self._preprocessor.preprocess(tweet)
            result.append([sum(tweet.count(symbol) for symbol in self._pos_symbols),
                           sum(tweet.count(symbol) for symbol in self._neg_symbols),
                           sum(tweet.count(symbol) for symbol in self._neu_symbols)])

        return preprocessing.normalize(result)

    def fit(self, df, y=None):
        return self

    def file_to_list(self, filename):
        return io.open(filename, encoding='utf-8').read().splitlines()
```

/analisis-sentimientos/extractors/TwitterExtractor.py

```python
from sklearn import preprocessing
from sklearn.base import BaseEstimator, TransformerMixin

from util.Preprocessor import Preprocessor


class TwitterExtractor(BaseEstimator, TransformerMixin):

    def __init__(self):
        pass

    def transform(self, data, y=None):
        result = []

        for tweet in data:
            tweet = Preprocessor.process_twitter_features(tweet, twitter_features=Preprocessor.NORMALIZE)
            result.append([tweet.count(Preprocessor.MENTION),
                           tweet.count(Preprocessor.URL),
                           tweet.count(Preprocessor.HASHTAG)])

        return preprocessing.normalize(result)

    def fit(self, df, y=None):
        return self
```

/analisis-sentimientos/util/Preprocessor.py

```python
import re

from nltk import TweetTokenizer
from nltk.stem import SnowballStemmer


class Preprocessor:

    NORMALIZE = 'normalize'
    REMOVE = 'remove'
    MENTION = 'twmention'
    HASHTAG = 'twhashtag'
    URL = 'twurl'
    LAUGH = 'twlaugh'

    DIACRITICAL_VOWELS = [('á','a'), ('é','e'), ('í','i'), ('ó','o'), ('ú','u'), ('ü','u')]
    SLANG = [('d','de'), ('[qk]','que'), ('xo','pero'), ('xa', 'para'), ('[xp]q','porque'),('es[qk]', 'es
que'),        ('fvr','favor'),('(xfa|xf|pf|plis|pls|porfa)', 'por favor'), ('dnd','donde'), ('tb', 'también'),
              ('(tq|tk)', 'te quiero'), ('(tqm|tkm)', 'te quiero mucho'), ('x','por'), ('\+','mas')]

    _stemmer = SnowballStemmer('spanish')
    _tokenizer = TweetTokenizer().tokenize

    def __init__(self, twitter_features=None, stemming=False):
        self._twitter_features = twitter_features
        self._stemming = stemming

    def preprocess(self, message):
        # convert to lowercase
        message = message.lower()
        # remove numbers, carriage returns and retweet old-style method
        message = re.sub(r'(\d+|\n|\brt\b)', '', message)
        # remove vowels with diacritical marks
        for s,t in self.DIACRITICAL_VOWELS:
            message = re.sub(r'{0}'.format(s), t, message)
        # remove repeated characters
        message = re.sub(r'(.)\1{2,}', r'\1\1', message)
        # normalized laughs
        message = self.normalize_laughs(message)
        # translate slang
        for s,t in self.SLANG:
            message = re.sub(r'\b{0}\b'.format(s), t, message)

        message = self.process_twitter_features(message, self._twitter_features)

        if self._stemming:
            message = ' '.join(self._stemmer.stem(w) for w in self._tokenizer(message))

        return message

    @staticmethod
    def process_twitter_features(message, twitter_features):
        message = re.sub(r'[\.\,]http','. http', message, flags=re.IGNORECASE)
        message = re.sub(r'[\.\,]#', '. #', message)
        message = re.sub(r'[\.\,]@', '. @', message)

        if twitter_features == Preprocessor.REMOVE:
            # remove mentions, hashtags and urls
            message = re.sub(r'((?<=\s)|(?<=\A))(@|#)\S+', '', message)
            message = re.sub(r'\b(https?:\S+)\b', '', message, flags=re.IGNORECASE)
        elif twitter_features == Preprocessor.NORMALIZE:
            # normalize mentions, hashtags and urls
            message = re.sub(r'((?<=\s)|(?<=\A))@\S+', Preprocessor.MENTION, message)
            message = re.sub(r'((?<=\s)|(?<=\A))#\S+', Preprocessor.HASHTAG, message)
            message = re.sub(r'\b(https?:\S+)\b', Preprocessor.URL, message, flags=re.IGNORECASE)

        return message
```

```
68
69      @staticmethod
70      def normalize_laughs(message):
71          message = re.sub(r'\b(?=\w*[j])[aeiouj]{4,}\b', Preprocessor.LAUGH, message, flags=re.IGNORECASE)
72          message = re.sub(r'\b(juas+|lol)\b', Preprocessor.LAUGH, message, flags=re.IGNORECASE)
73          return message
```

/analisis-sentimientos/util/PartsOfSpeechHelper.py

```
1  import hashlib
2  import os
3  import pickle
4
5  from nltk.tag.stanford import StanfordPOSTagger
6
7  os.environ['JAVAHOME'] = "C:/Program Files (x86)/Java/jre1.8.0_161/bin/java.exe"
8
9  class PartsOfSpeechHelper:
10
11      PATH_TO_MODEL = 'stanford-postagger/spanish-ud.tagger'
12      PATH_TO_JAR = 'stanford-postagger/stanford-postagger-3.8.0.jar'
13
14      _postag = StanfordPOSTagger(model_filename=PATH_TO_MODEL, path_to_jar=PATH_TO_JAR)
15
16      def __init__(self):
17          self.load_file_into_cache('stanford-postagger/precalculated_ud_postags.pkl')
18
19      def pos_tag(self, tokens):
20          key = self.key(tokens)
21
22          if key in self._cache:
23              return self._cache.get(key)
24
25          parts_of_speech = self._postag.tag(tokens)
26          self._cache[key] = parts_of_speech
27
28          return parts_of_speech
29
30      def key(self, tokens):
31          return hashlib.md5(''.join(tokens).encode('utf-8')).hexdigest()
32
33      def load_file_into_cache(self, filename):
34          with open(filename, 'rb') as f:
35              self._cache = pickle.load(f)
36
37      def save_cache_into_file(self, filename):
38          with open(filename, 'wb') as f:
39              pickle.dump(self._cache, f, pickle.HIGHEST_PROTOCOL)
40
41      def _pos_tag_batch(self, tokens_list):
42          parts_of_speech = self._postag.tag_sents(tokens_list)
43          for index, item in enumerate(parts_of_speech):
44              key = self.key(tokens_list[index])
45              self._cache[key] = item
```

/analisis-sentimientos/util/DatasetHelper.py

```python
import csv
import xml.etree.ElementTree as etree

from sklearn.model_selection import train_test_split


class DatasetHelper:

    @staticmethod
    def general_tass_to_list(filename):
        tree = etree.parse(filename)
        root = tree.getroot()
        data = []

        for tweet in root:
            tweetId = tweet.find('tweetid').text
            content = tweet.find('content').text
            polarityValue = tweet.find('sentiments/polarity/value').text
            data.append([tweetId, content.replace('\n',' '), polarityValue])

        return data

    @staticmethod
    def politics_tass_to_list(filename):
        tree = etree.parse(filename)
        root = tree.getroot()
        data = []

        for tweet in root:
            tweetId = tweet.find('tweetid').text
            content = tweet.find('content').text
            aux = next((e for e in tweet.findall('sentiments/polarity') if e.find('entity') == None), None)
            if aux != None:
                polarityValue = aux.find('value').text
                data.append([tweetId, content.replace('\n',' '), polarityValue])

        return data

    @staticmethod
    def intertass_tass_to_list(filename, qrel=None):
        tree = etree.parse(filename)
        root = tree.getroot()
        data = []

        for tweet in root:
            tweetId = tweet.find('tweetid').text
            content = tweet.find('content').text
            polarityValue = tweet.find('sentiment/polarity/value').text
            if polarityValue == None:
                polarityValue = qrel[tweetId]

            data.append([tweetId, content.replace('\n',' '), polarityValue])

        return data

    @staticmethod
    def gold_standard_to_dict(filename):
        with open(filename, 'r') as csvfile:
            reader = csv.reader(csvfile, delimiter='\t')
            data = {rows[0]: rows[1] for rows in reader}

        return data

    @staticmethod
    def list_to_csv(data, filename):
        with open(filename, 'w', encoding='utf-8') as csvfile:
            writer = csv.writer(csvfile, delimiter=',', lineterminator='\n', quoting=csv.QUOTE_NONNUMERIC)
            writer.writerows(data)

```

```
    @staticmethod
    def generate_train_test_subsets(data, size):
        codes = [d[0] for d in data]
        labels = [d[2] for d in data]
        codes_train, codes_test, labels_train, labels_test = train_test_split(codes, labels, train_size=size)
        train_data = [d for d in data if d[0] in codes_train]
        test_data = [d for d in data if d[0] in codes_test]
        return train_data, test_data

    @staticmethod
    def csv_to_lists(filename):
        messages = []
        labels = []
        with open(filename, 'r', encoding='utf-8') as csvfile:
            reader = csv.reader(csvfile, delimiter=',')
            for row in reader:
                messages.append(row[1])
                labels.append(row[2])
        return messages, labels


qrel = DatasetHelper.gold_standard_to_dict("../datasets/tass/intertass-sentiment.qrel")

data = []
data.extend(DatasetHelper.general_tass_to_list("../datasets/tass/general-test-tagged-3l.xml"))
data.extend(DatasetHelper.general_tass_to_list("../datasets/tass/general-train-tagged-3l.xml"))
data.extend(DatasetHelper.intertass_tass_to_list("../datasets/tass/intertass-development-tagged.xml"))
data.extend(DatasetHelper.intertass_tass_to_list("../datasets/tass/intertass-test.xml", qrel))
data.extend(DatasetHelper.intertass_tass_to_list("../datasets/tass/intertass-train-tagged.xml"))
data.extend(DatasetHelper.politics_tass_to_list("../datasets/tass/politics-test-tagged.xml"))

train, test = DatasetHelper.generate_train_test_subsets(data, size=0.3)

DatasetHelper.list_to_csv(data, '../datasets/global_dataset.csv')
DatasetHelper.list_to_csv(train, '../datasets/train_dataset_30.csv')
DatasetHelper.list_to_csv(test, '../datasets/test_dataset_30.csv')
```

Anexo II – Hoja de resultados

Se adjunta a este TFM una hoja de cálculo en formato Excel, de nombre *TFM - Análisis de sentimientos - Resultados.xlsx*, con todos los datos obtenidos a partir de las pruebas ejecutadas en el apartado 5 de esta memoria. Esta hoja de cálculo contiene varias pestañas:

- **K vecinos más cercanos:** contiene los resultados de la ejecución de *baseline_model.py* para el algoritmo k vecinos más cercanos. Se puede consultar la configuración seguida para cada una de las 32 tareas realizadas en el apartado 5.6 de este documento.
- **Naive Bayes**: igual que la pestaña anterior, pero para el algoritmo Naive Bayes.
- **Máquinas de vectores:** lo mismo, pero para el algoritmo máquinas de vectores de soporte.
- **Árboles de decisión:** de manera análoga que el punto anterior, pero para los árboles de decisión.
- **Unión algoritmos:** es la unión de los resultados de las cuatro pestañas anteriores.
- **Resultados baseline model:** es una simplificación ordenada de los datos de pestaña anterior en donde se muestran la información más relevante para cada algoritmo y tarea ejecutada
- **Gráficos baseline model:** información derivada de las anteriores pestañas para la generación de los gráficos mostrados en este documento.
- **Resultados improved model:** son los resultados de la parte práctica para la mejora del algoritmo línea de base.

[Página dejada en blanco intencionadamente]