Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Estudio de la portabilidad de un sistema de análisis de sentimiento de Tweets en castellano para el catalán

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Adrián Davia García

Tutor: María José Castro Bleda

Segundo tutor: Fernando García Granada

Curso 2020-2021

Resumen

El análisis de sentimiento en redes sociales, especialmente en Twitter, Facebook o Instagram ha sido objeto de central interés en los últimos años. Este análisis es clave para las compañías que buscan crecer su influencia, alcance y resultados. Para ello hay que recopilar datos de las redes y/o campañas sociales que después se puedan usar para dar seguimiento a tu rendimiento y así mejorarlo.

Hay mucho trabajo relacionado para idiomas de uso extendido como el inglés o el español. Sin embargo, para otros idiomas más minoritarios, como el catalán, no existen tales herramientas. El objetivo de este TFG será desarrollar un sistema para tweets en catalán. Se realizará un doble enfoque para obtener el objetivo deseado: usar herramientas de traducción automática para utilizar modelos entrenados para otros idiomas (en particular, en español) y, adicionalmente, entrenar nuevos modelos específicos para el catalán a partir de datos originalmente creado en catalán y/o traducidos de datos en español. Finalmente, se compararán ambas aproximaciones y se generará un modelo híbrido.

Palabras clave: Redes sociales, Tweets, Análisis de sentimiento, Traducción Automática

Resum

L'anàlisi de sentiment en xarxes socials, especialment en Twitter, Facebook o Instagram ha sigut objecte de central interés en els últims anys. Aquesta anàlisi és clau per a les companyies que busquen créixer la seua influència, abast i resultats. Per a això cal recopilar dades de les xarxes i/o campanyes socials que després es puguen usar per a donar seguiment al teu rendiment i així millorar-ho.

Hi ha molta faena relacionada per a idiomes d'ús estés com l'anglés o l'espanyol. No obstant això, per a altres idiomes més minoritaris, com el català, no existeixen tals eines. L'objectiu d'aquest TFG serà desenvolupar un sistema per a tuits en català. Es realitzarà un doble enfocament per a obtindre l'objectiu desitjat: usar eines de traducció automàtica per a utilitzar models entrenats per a altres idiomes (en particular, en espanyol) i, addicionalment, entrenar nous models específics per al català a partir de dades originalment creat en català i/o traduïts de dades en espanyol. Finalment, es compararan totes dues aproximacions i es generarà un model hibride.

Paraules clau: Xarxes socials, Tweets, Anàlisis de sentiment, Traducció Automàtica

Abstract

Sentiment analysis on social networks, especially on Twitter, Facebook or Instagram, has been of central interest in recent years. This analysis is key for companies looking to grow their influence, reach and results. To do this you need to collect data from social networks and/or campaigns that can then be used to track and improve your performance.

There is a lot of related work for widely used languages such as English or Spanish. However, for other more minority languages, such as Catalan, there are no such tools. The objective of this TFG will be to develop a system for tweets in Catalan. A two-fold approach will be carried out to obtain the desired goal: using machine translation tools to use models trained for other languages (in particular, Spanish) and, additionally, training new models specific for Catalan from data originally created in Catalan and/or translated from Spanish data. Finally, both approaches will be compared and a hybrid model will be generated.

Keywords: Social networks, Tweets, Sentiment analysis, Machine Translation

Índice general

Índice de figuras

Índice de tablas

Capítulo 1
Introducción

En la actualidad, con el avance de la tecnología, ha aumentado el número de personas que comparten parte de su vida diaria en redes sociales. Gracias a esto, podemos encontrar una diversidad de contenidos que nos permiten conocer las opiniones, gustos y aficiones de usuarios de todo el mundo. Estas aplicaciones nos ofrecen una cantidad enorme de datos que pueden ser utilizados para conocer la opinión pública sobre ciertos temas.

El análisis de sentimiento en redes sociales, especialmente en Twitter, Facebook o Instagram ha sido objeto de central interés en los últimos años. Este análisis es clave para las compañías que buscan crecer su influencia, alcance y resultados. Para ello hay que recopilar datos de las redes y/o campañas sociales para que después se puedan usar para dar seguimiento al rendimiento y así mejorarlo.

Esté trabajo se centrará en la creación de un corpus en catalán y en el desarrollo de un analizador de sentimiento que nos permita clasificar datos extraídos de Twitter – tweets – utilizando un cierto modelo de aprendizaje automático ya que existen muchos trabajos relacionados para idiomas de uso extendido como en el caso del inglés o el español, pero no para lenguajes minoritarios como el catalán o valenciano.

En este trabajo se describe el proceso empleado para el desarrollo de un analizador de sentimientos a partir de datos provenientes de Twitter utilizando para ello un clasificador basado en aprendizaje automático. El análisis de sentimientos es un campo de estudio que analiza la opinión de las personas, sentimientos, evaluaciones, actitudes y emociones desde el lenguaje escrito. Esta es una de las investigaciones más activas en el área de procesamiento de lenguaje natural.

Por lo general, para llevar a cabo un clasificador automático, es necesario un conjunto de muestras de entrenamiento. Analizando las muestras de entrenamiento se pueden encontrar los parámetros que hacen que el clasificador maximice su rendimiento en el proceso de entrenamiento. Para ello debemos tener un corpus invariable – conjunto de datos, en el caso de este trabajo, tweets – en el cual cada tweet esté convenientemente etiquetado [1].

El trabajo se centrará en la creación de un corpus de tweets recogidos del Tass del 2017 [2] y el CatSent [3], en castellano y en catalán respectivamente, los datos en catalán se obtendrán mediante la traducción automática de cuatro traductores: SaltGva [4], SoftCatala [5], Apertium [6] y Google [7], que será utilizado para realizar un estudio con un clasificador del tipo máquinas de vectores soporte.

1.1. Motivación

Gracias a la evolución de la tecnología en las últimas décadas, esto nos ha permitido que la comunicación entre las personas sea más sencilla, permitiendo conocer otras culturas, personas y dando acceso a un sinfín de información a las personas de a pie. Gracias a esta facilidad a la hora de acceder a la red han ido surgiendo diversas redes sociales que buscan conectar a las personas entre ellas. Encontramos desde aplicaciones para compartir fotografías, subir textos escritos por ti mismo y un largo etcétera, pero en este trabajo nos interesan aquellas redes sociales que nos permiten dar nuestra opinión a miles de personas sobre todo tipo de temas diferentes, desde hablar de la gastronomía hasta de la política.

Dado que el catalán es un lenguaje minoritario no hay gran cantidad de datos y herramientas. Por ello, este proyecto fue pensado a partir de la necesidad de tener un conjunto de datos para el catalán que permita, si es necesario, que se pueden utilizar en futuros proyectos.

1.2. Objetivos

El objetivo principal de este trabajo consiste en desarrollar un analizador de sentimiento para tweets – datos utilizados en este caso – en catalán. Para esto se realizarán varios pasos previos al desarrollo del analizador.

En primer lugar, se utilizarán herramientas de traducción automática para poder crear un corpus con datos en catalán que nos permita disponer de un conjunto de datos en catalán a partir de datos en castellano.

Una vez tengamos esté conjunto de datos, entrenaremos nuevos modelos específicos para el catalán utilizando los datos obtenidos anteriormente y otros conjuntos de datos creados originalmente en catalán.

Finalmente se compararán ambas aproximaciones y se generará un modelo híbrido para futuros proyectos.

1.3. Estructura de la memoria

El presente trabajo, que constituye la memoria del Trabajo de Fin de Grado, se divide en seis capítulos, los cuales pasaremos a describir a continuación; en la descripción de cada capítulo indicaremos de manera breve aquellos aspectos que se tratarán en cada capítulo, de modo que el lector pueda tener una visión global del texto que encontrarán posteriormente [1].

- **Capítulo 1. Introducción:** el primer capítulo pretende dar al lector una visión global de la temática del trabajo, en el cual se indican las motivaciones que han llevado a realizar dicho trabajo, así como los objetivos que se esperan alcanzar a la finalización del TFG.

- **Capítulo 2. Estado del arte:** en este capítulo introduciremos el concepto de procesamiento computacional del lenguaje natural, explicando conceptos como lenguaje y lenguaje natural, hablaremos sobre el análisis de sentimientos y más específicamente en el análisis de sentimiento en Twitter, hablaremos del conjunto de herramientas que han sido investigadas para el trabajo e introduciremos el concepto de corpus.

- **Capítulo 3. Creación del corpus:** en este capítulo analizaremos el problema de reutilizar datos extraídos de Twitter describiendo el diseño de la solución desarrollada, así como los detalles más importantes de la etapa de desarrollo.

- **Capítulo 4. Diseño de los experimentos:** en el cuarto capítulo describiremos el proceso por el cual hemos llevado a diseñar los diferentes experimentos y expondremos los objetivos que se quieren conseguir con los mismos.

- **Capítulo 5. Experimentación:** en este capítulo analizaremos los resultados de los experimentos definidos en el capítulo anterior, aportando los datos y estadísticas de estos.

- **Capítulo 6. Conclusiones y trabajo futuro:** en el último capítulo, expondremos el grado de cumplimiento de los objetivos marcados en el primer capítulo; así mismo también se expondrán algunas propuestas para mejorar o ampliar lo realizado en este trabajo y que no se hayan podido llevar a cabo.

Capítulo 2

Estado del arte

2.1. Procesamiento computacional del lenguaje natural

2.1.1. Lenguaje y Lenguaje Natural

Para entender en qué consiste el Procesamiento del Lenguaje Natural (PLN) primero debemos introducir el significado de lenguaje y posteriormente el concepto de lenguaje natural.

En primer lugar, el lenguaje es un sistema de signos que utiliza el ser humano, básicamente, para comunicarse con los demás o para reflexionar consigo mismo. Este sistema de signos puede ser expresado oralmente o por medio de la escritura [8].

Por su parte, el lenguaje natural es aquel que ha evolucionado con el tiempo para adaptarse a la comunicación humana de ese momento, como el español o el alemán [9]. Estos lenguajes continúan su evolución sin considerar la gramática, cualquier regla se desarrolla después de haber sucedido la evolución. En contraste, los lenguajes formales están definidos por reglas preestablecidas, y por tanto se rigen con todo rigor a ellas.

El lenguaje natural es el medio que utilizamos de manera cotidiana para establecer nuestra comunicación con las demás personas de nuestro entorno. El lenguaje natural ha venido perfeccionándose a partir de la experiencia a tal punto que puede ser utilizado para analizar situaciones altamente complejas y razonarlas muy sutilmente. Los lenguajes naturales tienen un gran poder expresivo y su función tiene un gran valor como una herramienta para el razonamiento y el entendimiento. Por otro lado, la sintaxis de un lenguaje natural puede ser modelada fácilmente por un lenguaje formal, similar a los utilizados en las matemáticas y la lógica [10].

2.1.2. Procesamiento del lenguaje natural

Una vez introducidos los términos de lenguaje y lenguaje natural, procederemos a explicar en qué consiste el procesamiento del lenguaje natural.

Una de las tareas fundamentales de la Inteligencia Artificial es la manipulación de lenguajes naturales usando herramientas de computación. Por Procesamiento de Lenguaje Natural (PLN, denominado también NLP por sus siglas en inglés) se entiende por la habilidad de la máquina para procesar la información comunicada, no simplemente la transcripción de lo comunicado [11]. En otras palabras, el PLN consiste en la utilización de un lenguaje natural para comunicarnos con el computador, teniendo éste que entender el mensaje que le sea proporcionado [10].

El objetivo principal que es perseguido es el de la comprensión del lenguaje humano por parte de un computador. La persecución de un objetivo tan ambicioso, del que todavía se está muy lejos, supondría una auténtica revolución. Por una parte, los ordenadores podrían tener por fin acceso al conocimiento humano, y por otra, una nueva generación de interfaces, en lenguaje natural, facilitaría en gran medida la accesibilidad a sistemas complejos [12].

2.1.3. Arquitectura de un sistema PLN

Para cumplir su objetivo, un sistema de PLN necesitará hacer uso de conocimiento acerca de la estructura del lenguaje. Este conocimiento del lenguaje natural se puede estructurar en niveles:

a. **Nivel morfológico:** mediante el cual se determina las palabras que componen el texto analizándolas para conocer cómo se construyen a partir de unas unidades de significado más pequeñas llamadas morfemas.

b. **Nivel sintáctico:** fija el papel estructural que cada palabra juega en la oración y que sintagmas son parte de otros sintagmas formando así oraciones.

c. **Nivel semántico:** trata el significado de las palabras y como al unirlas en oraciones le dan un significado independientemente del contexto.

d. **Nivel pragmático:** trata de ver cómo las oraciones obtienen significados diferentes según en la situación que se utilicen y ver como se ven afectadas por las oraciones anteriores.

Figura 1. Arquitectura de un sistema de Procesamiento del Lenguaje Natural

2.2. Análisis de sentimiento

El análisis de sentimiento (sentiment analysis en inglés), también llamado minería de opinión (opinion mining en inglés), es el campo de estudio que analiza las opiniones de las personas, sentimientos, evaluaciones, aptitudes y emociones con respecto a entidades como productos, servicios, organizaciones, individuos, eventos, temas, y sus atributos. Mientras que en el área industrial el término análisis de sentimiento es más comúnmente utilizado, en el área académica se utilizan tanto análisis de sentimiento como minería de opinión [13].

2.2.1. Estado del arte del análisis de sentimiento

El término análisis de sentimiento quizás apareció por primera vez en [14], y el término minería de datos apareció por primera vez en [15]. Sin embargo, la investigación sobre sentimientos y opiniones aparecieron antes [16, 17, 18, 19, 20, 21].

Pocas investigaciones fueron hechas sobre las opiniones y sentimientos de las personas antes del año 2000. Desde entonces, este campo de estudio se ha convertido en un área de investigación muy activa. Hay muchas razones para esto. Por ejemplo,

en este momento en la historia de la humanidad, nosotros tenemos un gran volumen de datos sobre opiniones en las redes sociales gracias a internet y su fácil accesibilidad. Sin estos datos, muchas de las investigaciones no podrían ser posibles. No es sorprendente que el rápido crecimiento de esta área vaya ligada al crecimiento de las redes sociales.

El análisis de sentimiento tiene un profundo impacto en la gestión, las ciencias políticas, económicas y sociales ya que están afectadas por las opiniones de las personas.

Aunque la investigación sobre el análisis de sentimiento principalmente empezó cerca del año 2000, hay algunos trabajos previos sobre interpretación de metáforas, adjetivos que denotan sentimiento, la subjetividad, los puntos de vista, y sobre el afecto [13, 22, 23, 24, 25].

2.2.2. Análisis de sentimiento en Twitter

Los mensajes publicados en Twitter, también llamados Tweets, constituyen un material de gran interés para detectar tendencias de opinión entre los usuarios. El hecho de que se hagan públicas opiniones, ideas y debates se asemeja en gran medida a una conversación informal. En el contexto de la comunicación política, el análisis de contenido y los estudios cuantitativos de los mensajes de Twitter permiten identificar patrones de comportamiento entre los usuarios y puntos de inflexión en las corrientes de opinión [26].

Sin embargo, la investigación en comunicación requiere complementar los análisis cuantitativos con consideraciones de orden cualitativo. Dado el gran volumen de mensajes de Twitter que habitualmente hay que evaluar, conviene desarrollar métodos que procesen textos de forma automática con una fiabilidad de precisión aceptable. De esta forma, el investigador estaría en condiciones de cualificar mejor las opiniones y los datos extraídos de la conversación entre los usuarios. El análisis de sentimiento en Twitter surge como respuesta a esta necesidad [27].

Analizar el sentimiento en Twitter supone asignar a cada mensaje publicado un valor relacionado con la carga emocional que transmite. Con relación a esta carga emocional se pueden distinguir algunos tipos de alcances diferentes [28]:

- Polaridad: algunos métodos y recursos tienen el propósito de extraer información polarizada. Los métodos orientados a la polaridad normalmente devuelven variables cuyos posibles valores son positivos, negativos y neutrales. Por otro lado, los recursos léxicos están compuestos por listas de palabras cuyos valores son positivo y negativo.

- Intensidad: algunos métodos y recursos proporcionan niveles de intensidad conforme a la polaridad de sentimientos. Los métodos orientados a la intensidad nos aportan valores numéricos indicando la intensidad de los sentimientos de positividad o negatividad expresados en extractos de textos. Los recursos léxicos orientados en la intensidad se componen de listas de palabras junto con valores de intensidad con respecto a la positividad o negatividad de estas.

- Emoción: estos métodos están enfocados en la extracción de la emoción o el estado de ánimo de un extracto de texto. Estos métodos suelen clasificar el mensaje en categorías emocionales como tristeza, sorpresa, aburrimiento y otras más. Los recursos léxicos orientados en la emoción nos proporcionan una lista de palabras y/o expresiones etiquetadas según los diferentes estados de emoción.

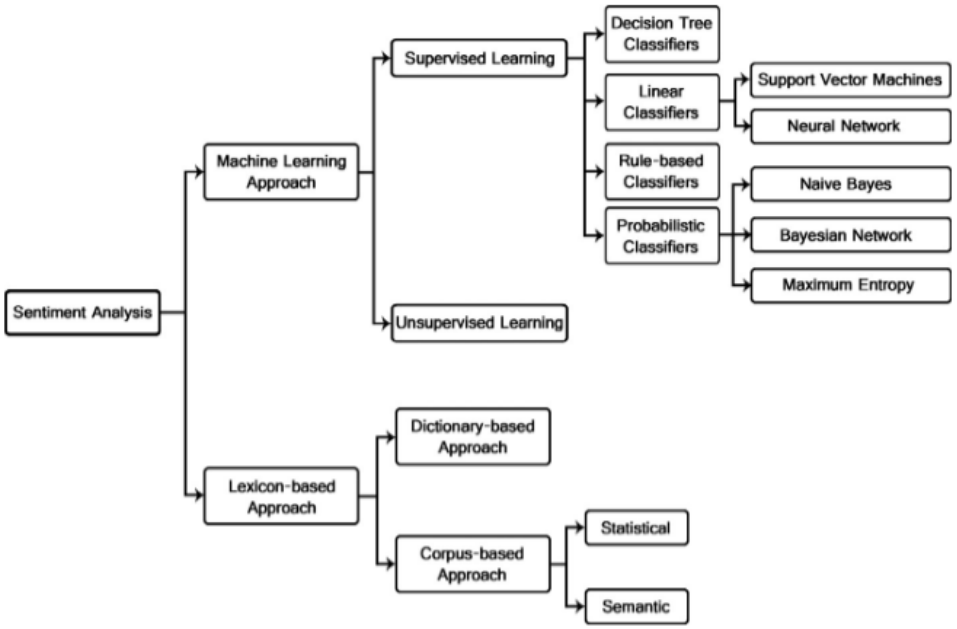

Figura 2. Técnicas de clasificación de sentimientos.

Fuente: [29]

Según Medhat [29]. Las principales técnicas de análisis de sentimiento se dividen en dos grandes grupos:

- Las basadas en aprendizaje automático (machine learning approach): el aprendizaje automático es un tipo de inteligencia artificial (AI) que proporciona a las computadoras la capacidad de aprender, sin ser programadas explícitamente. El aprendizaje automático se centra en el desarrollo de programas informáticos que pueden cambiar cuando se exponen a nuevos datos [30].

- Las basadas en diccionarios (lexicon-based approach): las palabras que expresan opinión son empleadas en varias tareas de clasificadores de sentimiento. Las palabras de opinión positivas son usadas para expresar estados de interés mientras que las negativas son utilizadas para expresar estados de desacuerdo [29].

2.3. Conjunto de herramientas

En esta sección se pretende abordar todas aquellas herramientas que se han estudiado y/o usado a lo largo del desarrollo del trabajo. Cabe destacar que el lenguaje de programación utilizado ha sido Python, el cual es un lenguaje de programación interpretado, orientado a objetos de alto nivel y con semántica dinámica. Su sintaxis hace énfasis en la legibilidad del código, lo que facilita su depuración y, por tanto, favorece la productividad. Ofrece la potencia y la flexibilidad de los lenguajes compilados con una curva de aprendizaje suave. Aunque Python fue creado como lenguaje de programación de uso general, cuenta con una serie de librerías y entornos de desarrollo para cada una de las fases del proceso de Data Science. Esto, sumado a su potencia, su carácter open source y su facilidad de aprendizaje le ha llevado a tomar la delantera a otros lenguajes propios de la analítica de datos por medio de Machine Learning [31].

Por un lado, encontramos diferentes herramientas para el procesado del lenguaje natural. Siendo algunas de ellas:

- NLTK - Natural Language Toolkit: conjunto de herramientas gratuitas diseñadas para los desarrollos que trabajan con lenguaje natural. Provee interfaces de fácil uso con más de 50 recursos léxicos y corpus con un grupo de librerías de procesamiento de texto de clasificación, tokenización, estemización, etiquetado, parseo y razonamiento semántico [32].

- FreeLing: librería de C++ la cual provee diferentes funcionalidades para el análisis del lenguaje (análisis morfológico, parseo, etc) en una variedad de lenguajes como español, inglés o catalán entre otros [33]. Estás herramientas están desarrolladas y mantenidas por TALP Research Center que se encuentra en la Universitat Politècnica de Catalunya siendo la librería de libre acceso.

- MeaningCloud: es un conjunto de APIs de pago que ofrecen diferentes herramientas para analizar texto proporcionado en modo SaaS (Software as a Service) y on-premises. Es posible integrarlo de manera rápida gracias a los SDKs entres lenguajes: Java, Python y PHP. Permitiendo añadir a la aplicación un entorno de trabajo (framework en inglés) con el que incluir de manera sencilla los análisis. Encontramos diferentes APIs que nos permiten llevar a cabo análisis de sentimiento,

extracción de tópicos y análisis de estructuras de documentos entre otros [34]. Es necesario apuntar que, aunque para obtener los servicios completos es necesario obtener la versión de pago, existe un plan gratuito que incluye 20 000 peticiones por mes a las API públicas que abordan todos los aspectos de la analítica de textos con APIs especializadas en extracción de temas, clasificación de textos, análisis de sentimiento, identificación del lenguaje, lematización y análisis sintáctico, reputación corporativa, agrupamiento de textos, obtención automática de resúmenes o análisis de la estructura de documentos a un ritmo de 2 peticiones por segundo, así como el soporte técnico necesario. Además de que cada petición a las API permite analizar hasta 500 palabras [35].

- Cloud Natural Language: API de Google que proporciona a los desarrolladores tecnologías de compresión del lenguaje natural, incluido el análisis de opinión, el análisis de entidades, el análisis de opiniones sobre entidades, la clasificación de contenidos y el análisis sintáctico. Está API forma parte de la familia de Cloud Machine Learning [36]. Igual que la anterior, también es una API de pago, aunque puede usarse una cuenta de prueba para realizar algunos análisis.

- Azure API de Microsoft (Text Analytics): servicio de inteligencia artificial para minería de textos y análisis, incluidos: análisis de sentimientos, minería de opiniones, detección de lenguaje y reconocimiento de entidades con nombre. Está API forma parte de Azure Cognitive Services, una colección en la nube de algoritmos de aprendizaje automático y de inteligencia artificial [37]. Es una API de pago.

- Watson Natural Language Undertanding: analizador de pago de IBM, provee diferentes análisis de textos pudiendo utilizarse tanto en Java, Curl, Node, Python, Go, .NET, Ruby, Swift y Unity. Entre sus opciones encontramos un analizar de conceptos, de emociones, de entidades, de metadatos, de relaciones, de roles semánticos y de sentimientos entre otros [38].

- Sumy [39]: es una biblioteca simple que es posible utilizarla por desde un terminal mediante comandos que es utilizada para extraer resúmenes de texto sin formato o de páginas HTML. También contiene un marco de evaluación simple para resúmenes de texto. Algunos de los métodos de resumen implementados son:

- SumBasic: método que se utiliza a menudo como referencia en la literatura. Es un sistema que produce resúmenes genéricos de varios documentos. Su diseño está motivado por la observación de que las palabras que aparecen con mayor frecuencia en el grupo de documentos también lo hacen con mayor probabilidad en los resúmenes hechos por humanos que aquellas palabras que aparecen con menor frecuencia [40].

- Reducción: resumen basado en grafos, en la que la relevancia de una frase se calcula como la suma de los pesos de sus aristas con respecto a otras frases.

- KL-SUM: Método que agrega con avidez oraciones a un resumen siempre que disminuya la divergencia de KL que representa la divergencia entre la verdadera distribución y la distribución aproximada [41].

De todas las herramientas expuestas anteriormente solo algunas de ellas tienen la opción de trabajar con el idioma catalán, entre ellas están FreeLing y MeaningCloud.

2.4. Corpus

Según la RAE (Real Academia de la lengua Española), el corpus es un conjunto lo más extenso y ordenado posible de datos o texto científicos, literarios, etcétera, que pueden servir de base en una investigación [42].

En el caso de este trabajo los corpus se deben componer de al menos dos campos. Uno de ellos nos debe ofrecer el etiquetado de los datos y el otro, los datos, que en nuestro caso son publicaciones de Twitter (Tweets).

Después de realizar una búsqueda exhaustiva hemos encontrado gran cantidad de corpus para español, inglés o alemán. Estos son algunos de ellos:

- TASS: El cual es un taller de análisis semántico en la SEPLN (Sociedad española para el procesamiento del lenguaje natural) en el cual podemos encontrar corpus desde el 2012 hasta el 2020. Todos estos corpus se basan en datos recopilados de Twitter en castellano [2].

- SemEval: Es una serie de talleres internacionales de investigación de procesamiento de lenguaje natural cuya misión es avanzar en el estado actual del análisis semántico y ayudar a crear conjuntos de datos anotados de alta calidad en una variedad de problemas cada vez más desafiantes en la semántica del lenguaje natural. El taller de cada año presenta una colección de tareas compartidas en las que se presentan y comparan sistemas de análisis semántico computacional diseñados por diferentes equipos [43]. En este caso los datos los encontrábamos en inglés y únicamente para la tarea 4 del año 2016 [44].

- GermEval: es una serie de campañas de evaluación de tareas compartidas que se centran en el procesamiento del lenguaje natural para el idioma alemán [45].

Sin embargo, la cantidad de corpus que podemos encontrar para lenguajes minoritarios se reduce muchísimo, ya que, después de realizar una búsqueda exhaustiva para encontrar corpus en catalán, únicamente hemos encontrado el corpus de CatSent, el cual se compone de un conjunto de tweets en catalán de mano de Pau Balaguer, experto en Ciencia de datos en Eurecat – Centro tecnológico de Cataluña. Este corpus está compuesto por 50000 tweets en catalán [3].

Capítulo 3

Creación del corpus

3.1. Introducción

En esta sección explicaremos el proceso que hemos llevado a cabo para conseguir crear un corpus en formato csv en catalán con el objetivo de llevar a cabo un análisis de sentimiento con datos extraídos de Twitter (Tweets). Para conseguirlo hemos realizado una búsqueda de corpus orientados para esta área de investigación y hemos encontrado dos que nos han parecido interesante utilizar.

Uno de estos corpus ha sido obtenido del TASS [2], el cual consiste en un conjunto de tweets en castellano. Por otro lado, hemos considerado utilizar el corpus en catalán que utilizó Pau Balaguer en su trabajo sobre análisis de sentimiento en catalán (CatSent) [3].

Por otro lado, para llevar a cabo nuestro objetivo, teníamos que obtener la máxima cantidad de datos en catalán, por ello, una parte del proceso de desarrollo ha consistido en la utilización de traductores automáticos disponibles para el idioma catalán que nos permitiese obtener los datos en el idioma deseado en este trabajo.

3.2. Corpus para análisis de sentimiento

3.2.1. Corpus en castellano

Como hemos mencionado anteriormente, en este trabajo hemos obtenido los datos del TASS [46] del año 2017. En concreto, utilizaremos los conjuntos de datos de InterTASS [2]. El corpus lo podemos encontrar en formato XML compuesto por etiquetas de tipo <tweet> que se componen de las siguientes etiquetas:

- <tweetid> que corresponde al identificador del tweet.
- <user> que indica el usuario que publicó el tweet
- <content> contiene el contenido del tweet.
- <date> indica la fecha y hora de publicación.
- <lang> indica el idioma en el que fue publicado.
- <sentiment> en la que encontramos la etiqueta <polarity> que corresponde a la clasificación que pertenece a ese tweet. Pudiendo estar etiquetados con cuatros diferentes valores: Positivo (P), Negativo (N), Neutral (NEU) y sin polaridad (NONE).

```
<tweet>
  <tweetid>768224728049999872</tweetid>
  <user>caval100</user>
  <content>Se ha terminado #Rio2016 Lamentablemente no arriendo las ganancias al pueblo brasileño por la
penuria que les espera Suerte y solidaridad</content>
  <date>2016-08-23 23:13:42</date>
  <lang>es</lang>
  <sentiment>
    <polarity><value>N</value></polarity>
  </sentiment>
</tweet>
```

Figura 3. Composición de los datos del corpus InterTASS

El corpus está compuesto por tres conjuntos de datos:

- Conjunto de entrenamiento con un total de 1008 tweets.
- Conjunto de desarrollo con un total de 506 tweets.
- Conjunto de test con un total de 1889 tweets.

3.2.2. Corpus en catalán

En este caso, mientras se llevaba a cabo la investigación para el desarrollo del trabajo encontramos un corpus etiquetado para el idioma catalán. Este lo podemos encontrar en el repositorio de Pau Balaguer [3].

El corpus consiste en un fichero en formato txt de 50000 líneas que se componen de la polaridad de los datos, la cual puede tomar el valor 0 (Negativo) o 1 (Positivo), y el contenido del tweet.

3.3. Creación de las conexiones con los traductores

Uno de los primeros pasos para conseguir obtener el corpus que necesitamos consiste en crear las conexiones con los traductores automáticos que se van a utilizar para la conversión de los datos de castellano a catalán.

En este caso se ha decidido trabajar con cuatro traductores diferentes, los cuales son:

- El traductor de Google (del inglés Google Translate) es un sistema multilenguaje de traducción automática, desarrollado y proporcionado por Google, para la traducción de texto, voz, imágenes o video en tiempo real. Ofrece una interfaz web, así como interfaces para móviles IOS y Android, y una API, que los desarrolladores pueden utilizar para construir extensiones de navegación, aplicaciones y otros softwares. Dispone de más de 100 idiomas en distintos niveles para la traducción [7].

- El traductor SALT (Salt.usu) es una herramienta de traducción de valenciano a castellano y de castellano a valenciano, y de corrección de textos en valenciano. Se basa en un software libre adaptado por la Generalitat Valenciana (GVA) [4].

- El traductor de Softcatalà es un traductor automático entre el catalán y las siguientes lenguas: castellano, inglés, portugués, francés, occitano (aranés)

y aragonés. Para la traducción inglés-catalán se usa también un traductor neuronal desarrollado por Softcatalà [5].

- El traductor Apertium es uno de los sistemas de traducción automática de código abierto que se originó dentro del proyecto" Open-Source Machine Translation for the Languages of Spain" ("Traducción automática de código abierto para las lenguas del estado español"). Es un sistema de traducción automática de transferencia superficial, inicialmente diseñado para la traducción entre pares de idiomas relacionados, aunque algunos de sus componentes también se han utilizado en la arquitectura de transferencia profunda (Matxin) [6]. Cabe destacar que los traductores de la Generalitat Valenciana y el de Softcatalà utilizan el sistema apertium.

Se ha diseñado un programa en Python en el cual se han desarrollado cinco métodos diferentes, uno por cada traductor más un método auxiliar que nos permite llamar a los métodos que se encargan de la traducción de forma sencilla.

Para conseguir utilizar los traductores desde nuestros desarrollos en Python, se han utilizado diferentes herramientas.

Para los traductores de Google, el de la GVA y el de Softcatlà se han llevado a cabo mediante peticiones de tipo GET, se ha utilizado la API request, la cual permite enviar solicitudes HTTP/1.1 [47], estas solicitudes son los medios por los cuales se intercambian datos entre servidores y clientes. Hay dos tipos de mensajes: peticiones, enviadas por el cliente al servidor para pedir el inicio de una acción; y respuestas, que son el resultado que devuelve el servidor a la petición enviada [48].

Para la utilización del traductor Apertium, se ha utilizado la APIs subprocess que permite generar nuevos procesos, conectarse a las tuberías de entrada, salida o error y obtener los valores de retorno [49], permitiendo así ejecutar comandos desde los ficheros Python ya que en este caso se ha utilizado el comando que proporciona la API de apertium.

```
echo 'This is a test sentence' | apertium xxx-yyy
```

Figura 4. Ejemplo de uso de la API de apertium

Una vez estaban funcionando los traductores, se llevó a cabo una comprobación del estado de las traducciones. Para ello se utilizó un conjunto de 1000 tweets del corpus de interTASS.

Este análisis nos indica si las traducciones son iguales entre sí, comparando cada traducción con las traducciones de los siguientes traductores. Los resultados obtenidos nos mostraron que el 83,2% de las traducciones diferían en alguna palabra o en cómo estaba montada la oración, a raíz de esto decidimos llevar a cabo un nuevo análisis. En este análisis, analizamos el vocabulario que obtenemos al procesar cada traducción por separado, guardando en un fichero cada palabra, junto al número de veces que aparecía esa palabra.

Estos resultados nos hicieron darnos cuenta de que no podíamos desechar ninguno de los traductores, ya que al dar resultados diferentes no podíamos saber cuál de ellos es más efectivo.

3.4. Tratamiento de los datos

3.4.1. Preprocesamiento de los datos del corpus interTASS

El primer paso para crear nuestro corpus es obtener el id, el contenido y la polaridad de los tweets y guardarlos en ficheros txt. En este caso se ha utilizado la API xml.dom.minidom [50], siendo una implementación mínima de la interfaz Document Object Model. Es una implementación más simple y pequeña que DOM.

Con esta herramienta se consigue acceder a los datos del xml de una forma sencilla que permita tratar los datos de forma rápida. En este paso remplazamos los saltos de líneas por espacios en blanco para que no tengamos los datos de cada tweet en una línea.

3.4.2.Preprocesamiento de los datos del corpus CatSent

En este caso, no es necesario tratar de forma especial los datos, ya que los datos se encuentran en un fichero txt. Únicamente se ha eliminado los caracteres especiales

como salto de línea (\n), tabulaciones (\t) y similares que dificultarían la traducción y posterior tratamiento de los datos.

3.5. Creación del corpus. Fichero CSV

Una vez funcionando las conexiones, nos disponemos a crear el corpus tratando tantos los tweets traducidos como los originales.

El primer paso es limpiar el tweet de todo aquello que no queramos que sea traducido, en nuestro caso esas partes son: los hashtags, las urls y los usuarios. Estas partes no deben traducirse ya que el cambio no es adecuado, pues, por ejemplo, en el hashtag #Perro no habrá los mismos tweets que en el hashtag #Gos, al igual que los usuarios o las urls, si se traducen no conducirán a la misma página o al mismo usuario que en su versión original. Para esto, hacemos una búsqueda por el contenido del tweet y remplazamos con @CUSER, @CURL y @CHASHTAG. Además, los valores originales son guardados para que una vez se haya finalizado la traducción se puedan volver a remplazar con los valores reales. En este paso averiguamos que el traductor de Google a la hora de traducir reemplazaba la primera letra con una mayúscula si esa primera letra era una vocal, por ello se decidió que los reemplazos tuviesen una “C” al principio.

Otro asunto para tener en cuenta son las comillas (“”), las cadenas de caracteres en los lenguajes de programación se identifican porque se entrecomillan, pues para Google y apertium teníamos que comprobar que sólo existiesen las comillas del principio o del final, ya que, por ejemplo, para Google si encontraba un punto, terminaba la traducción y únicamente nos devolvía hasta el mismo.

Una vez tengamos el tweet limpio, nos disponemos a traducirlo. Mediante el método auxiliar del programa donde creamos las conexiones a los traductores, pasándole el nombre del traductor, el tweet limpio y el idioma de origen, esto nos devuelve el resultado de la traducción de ese tweet. En el momento en el que se ha obtenido la traducción se vuelven a poner los hashtags, urls y los usuarios originales en el tweet traducido y se guarda. Este proceso se repetirá por cada uno de los cuatro traductores.

Una vez realizadas las traducciones, nos disponemos a estemizar los resultados de la traducción y el texto original. El estemizado consiste en el procesado de palabras para

obtener la raíz de las mismas llamadas lexema, obteniendo al final un conjunto de lexemas de todo el texto. Está parte se lleva a cabo con la API pyStemmer la cual provee acceso a algoritmos eficientes para calcular una forma "derivada" de una palabra. Esta es una forma en la que se han eliminado la mayoría de las terminaciones morfológicas comunes; con suerte, representa una forma de base lingüística común [51], la cual es una de las pocas herramientas que procesa palabras en catalán. Aunque se intentó utilizar el API freeling [33], está solo nos procesaba los textos sin opción de estemizar, por ello se decidió descartar esta opción. A parte, se llevará a cabo la tokenización de los datos. Esto consiste en mantener únicamente las palabras que sean relevantes a la hora de analizar los datos.

Está parte se lleva a cabo para una vez experimentemos, podamos comprobar si los resultados con las raíces de las palabras dan valores mejores que únicamente utilizando las palabras en su forma original.

Para llevar a cabo esta parte, se han eliminado stopwords, que son palabras muy frecuentes en un idioma que carecen de valor, por ejemplo, las preposiciones. También es importante eliminar los hashtags, usuarios y urls. También se debió tener en cuenta los signos de puntuación y los apóstrofes para el lenguaje catalán, ya que a raíz de las traducciones puede que no se hayan realizado correctamente.

Una vez se han llevado a cabo estos procesos, los guardaremos en un fichero csv, con codificación UTF-8, el cual es un archivo de texto que tiene un formato específico que permite guardar los datos en un formato de tabla estructurada. En el encontraremos el id del tweet, el nombre del corpus, el tipo, el cual corresponde a si son datos de test, desarrollo, entrenamiento o si los datos son, en origen, del corpus de CatSent. Seguidamente encontramos el idioma original, el idioma al que se ha llevado a cabo la traducción, el traductor utilizado, la polaridad, la traducción del tweet, el tweet traducido, la traducción estemizada, el original estemizado, la traducción no estemizada y, por último, el original no estemizado. Estos últimos cuatro campos están tokenizados y en formato de lista, ya que así en el momento de la experimentación nos resultará más sencillo utilizarlo.

Una vez finalizado todo el proceso, nuestro corpus cuenta con 213612 tweets, ya que hemos traducido con los cuatro traductores los 1008 tweets del conjunto de entrenamiento del TASS, los 506 tweets del conjunto de desarrollo del TASS, los 1889 tweets del conjunto de test del TASS y los 50000 tweets del corpus CatSent. Esto hace

que al final tengamos un corpus, de un tamaño considerable, multilenguaje ya que encontramos la versión original en castellano de los tweets del TASS junto a sus cuatro versiones traducidas, una por cada traductor, al catalán además de los datos originales en catalán de CatSent junto a sus cuatro versiones traducidas, una por cada traductor, al castellano.

Capítulo 4

Diseño de los experimentos

4.1. Experimentos

En esta sección, describiremos los experimentos que se han llevado a cabo. El preproceso usado es el descrito en el capítulo anterior. Como clasificador, se ha empleado SVM (Support Vector Machine en inglés, o Máquina de vectores soporte en español) lineal.

Como entrada al clasificador, se han usado diferentes agrupaciones de palabras, N-gramas y SkipGramas [52, 53]. La forma en la que extraemos los N-gramas se tiene que adaptar al ámbito que estamos estudiando y al objetivo que tenemos en mente. Por ejemplo, en el estudio del lenguaje natural podríamos construir los N-gramas sobre la base de distintos tipos de unidades tales como fonemas, sílabas, letras, o palabras. Algunos sistemas procesan las cadenas de texto eliminando los espacios, pero otros no. En nuestro caso, todos los N-Gramas utilizados han sido mediante texto que ya había sido procesado:

a. Unigramas: únicamente se utilizarán grupos de una sola palabra.

b. Bigramas: únicamente se utilizarán grupos de dos palabras consecutivas.

c. Unigramas y bigramas: en este caso se utilizan los dos casos anteriores.

d. 1-6 gramas: en este caso se utilizarán grupos de una, dos, tres, cuatro, cinco y seis palabras respectivamente.

e. Unigramas y SkipGramas: se utilizarán grupos de unigramas y grupos de SkipGramas que consisten en grupos de palabras en los que se salta la palabra

que hay entre medias, por ejemplo, de la siguiente lista '["casa", "roja", "vieja"]', obtendremos '["casa", "vieja"]'.

Una vez introducidos los tratamientos de los datos que se llevarán a cabo en todos los experimentos, introduciremos los experimentos que hemos llevado a cabo:

- Experimento 1a. Se utilizarán los datos del corpus TASS en castellano (idioma original) utilizando únicamente las particiones de test y entrenamiento, este último se compone de 1008 tweets. Se unirán las etiquetas de polaridad con valor NONE y NEU, quedándonos únicamente con datos etiquetados con el valor NEU. El objetivo es ver si los resultados son comparables a los resultados obtenidos en el SemEval [54].

- Experimento 1b. En esta parte, consideramos eliminar las etiquetas NEU y NONE siendo el objetivo observar si una vez hechos los experimentos, el resultado obtenido mejora. Para este experimento se utilizan un total de 637 tweets como partición de entrenamiento.

- Experimento 2. Se utilizarán los datos del corpus TASS en castellano con las traducciones del corpus de CatSent traducidas del catalán. La partición de entrenamiento se compondrá de la partición de entrenamiento del TASS con todos los datos del CatSent traducido, consiguiendo que tengamos una partición de 204032 tweets, y, por otro lado, la partición de test se compondrá de la partición test del TASS. Se eliminarán las muestras etiquetadas con NEU o NONE. El objetivo de este experimento es comprobar si mejoran los resultados obtenidos en el experimento 1b añadiendo datos obtenidos mediante los traductores empleados durante el trabajo.

- Experimento 3. Se utilizarán los datos del corpus TASS en valenciano, traducido del castellano. Se utilizarán las particiones de test y entrenamiento, esta última partición con un total de 4032 tweets. En este experimento volveremos a unir las muestras con las etiquetas NONE y NEU.

- Experimento 4. Se utilizarán los datos del corpus TASS en valenciano, traducido del castellano, más todos los datos de CatSent, consiguiendo que

tengamos una partición de 54032 tweets. Se utilizarán las mismas particiones que en el experimento 2 y se eliminarán las muestras con etiquetado NONE y NEU. El objetivo consiste en comprobar si hay una mejora en los resultados obtenidos en el experimento 3.

- Experimento 5. Se utilizarán los datos del corpus CatSent en una proporción 75% para la partición de test y 25% para la partición de entrenamiento.

- Experimento 6. Se utilizarán los datos del corpus CatSent traducidos del catalán con la misma proporción que en el experimento anterior.

4.2. Máquinas de vectores soporte

En esta sección hablaremos de las máquinas de vectores soporte (SVM, Support Vector Machine en inglés), ya que es la técnica utilizada para entrenar nuestros modelos de análisis de sentimiento.

Antes de nada, debemos introducir el significado de aprendizaje supervisado, el cual es el tipo de algoritmo de Machine Learning más frecuentemente utilizado. Utiliza un conjunto de datos conocidos (denominado conjunto de datos de entrenamiento) para entrenar un algoritmo con un conjunto de datos de entrada conocidos (denominados características) y respuestas conocidas para realizar predicciones. El conjunto de datos de entrenamiento incluye datos de entrada etiquetados que se emparejan con los valores de salida o de respuesta deseados. A partir de esto, el algoritmo de aprendizaje supervisado intenta crear un modelo estableciendo relaciones entre las características y los datos de salida para realizar predicciones acerca de los valores de respuesta para un nuevo conjunto de datos [55].

Este tipo de aprendizaje se utiliza en muchos problemas de clasificación y regresión, incluidas aplicaciones médicas de procesamiento de señales, procesamiento del lenguaje natural y reconocimiento de imágenes o voz.

El objetivo del algoritmo SVM es encontrar un hiperplano que separe de la mejor forma posible dos o más clases diferentes de puntos de datos. “De la mejor forma posible” implica que el hiperplano tenga el margen lo más amplio entre las dos clases, representado por los signos más y menos en la siguiente figura. El margen se define como la anchura máxima de la región paralela al hiperplano que no tiene puntos de datos interiores. El algoritmo sólo puede encontrar este hiperplano en problemas que permiten separación lineal [56].

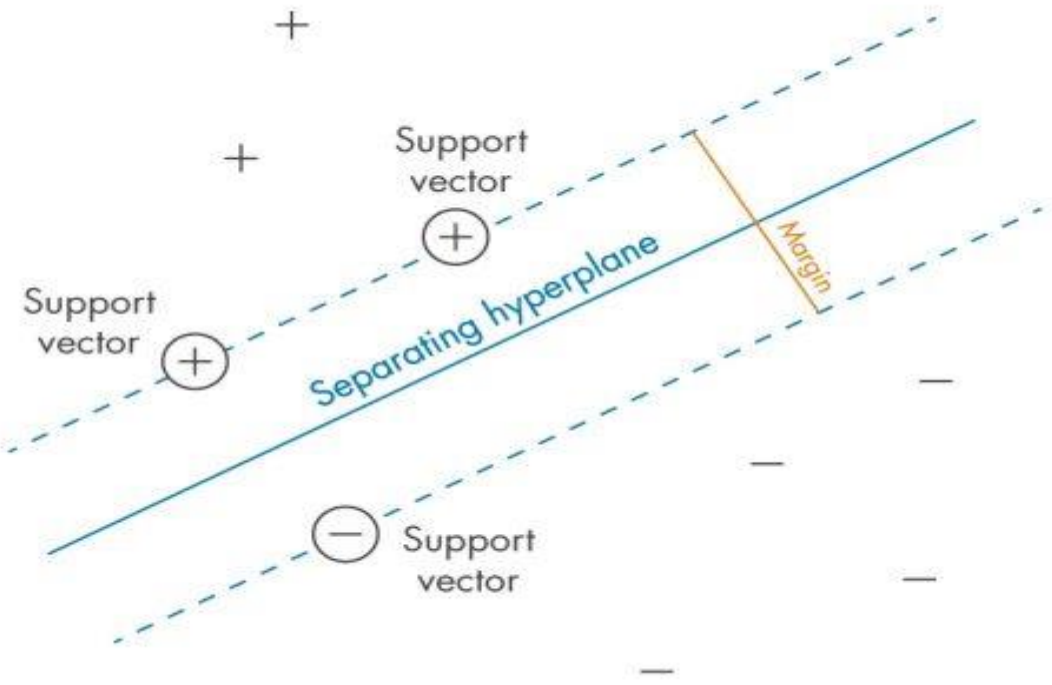

Figura 5. Definición del “margen” entre las clases: el criterio que los SVM intentan optimizar.
Fuente: [56]

En este trabajo se ha utilizado un SVM lineal mediante la biblioteca sklearn[1] que es una biblioteca para aprendizaje automático de software libre para el lenguaje de programación Python. Incluye varios algoritmos de clasificación, regresión y análisis de grupos entre los cuales están máquinas de vectores de soporte, bosques aleatorios, Gradient boosting, K-means y DBSCAN [57].

Una parte del proceso para el análisis consiste en la vectorización de las palabras, el cual es un proceso que convierte una colección de documentos de texto en vectores de características numéricas. Hay muchos métodos para convertir datos textuales en

[1] https://scikit-learn.org/stable/

vectores que el modelo pueda entender, pero el método más popular se llama TF-IDF. Este es un acrónimo que significa “Frecuencia de términos - Frecuencia de documento inverso”, que son los componentes de las puntuaciones resultantes asignadas a cada palabra. Resaltando así las palabras más interesantes [58].

La configuración que hemos utilizado para el SVM la podemos ver en la siguiente figura.

$$svm.SVC(C = 1.0, kernel =' linear', degree = 3, gamma =' auto')$$

Figura 6. Parametrización del SVM.

El parámetro C puede ser definido como un parámetro de regularización. Este es el único parámetro libre de ser ajustado en la formulación de la SVM. El ajuste de este parámetro puede hacer un balance entre la maximización del margen y la violación a la clasificación [59].

El parámetro kernel corresponde a la función de clasificación que se utilizará durante el proceso, en este caso será una función kernel lineal la cual sirve para proyectar la información a un espacio de características de mayor dimensión gracias al cual aumenta la capacidad computacional de las máquinas de aprendizaje lineal.

$$< x, x' >$$

Figura 7. Función kernel lineal.

El parámetro degree es utilizado en las funciones kernel de tipo polinómicas.

$$(\gamma < x, x' > + r)^d$$

Figura 8. Función kernel de tipo polinómica.

El parámetro gamma define cuánta influencia tiene un solo ejemplo de entrenamiento.

4.3. Medidas de evaluación

En este capítulo detallaremos cómo hemos procedido a la hora de evaluar las mediciones obtenidas a partir de los experimentos detallados en el capítulo anterior. Nos guiaremos a partir del siguiente artículo [54].

La siguiente figura muestra la matriz de confusión obtenida. La celda XY indica el número de tweets que han sido clasificados como X y que deberían haber sido clasificados como Y, donde P, U y N se refieren a Positivo, Neutral y Negativo respectivamente.

		Actual		
		Pos	Neu	Neg
Predicho	Pos	PP	PU	PN
	Neu	UP	UU	UN
	Neg	NP	NU	NN

Figura 9. Matriz de confusión.

Como una medida de evaluación utilizamos macroaverage recall:

$$\rho^{PN} = \frac{\rho^{Pos} + \rho^{Neg}}{2}$$

Figura 10. Ecuación macroaverage recall

donde ρ^{Pos} and ρ^{Neg} son las clases positivas y negativas, respectivamente, ρ^{PN}en el rango [0,1], donde 1 es conseguido únicamente por un clasificador perfecto (aquel clasificador que es capaz de clasificar correctamente todas las muestras), 0 es cuando el clasificador está pervertido (todas las muestras han sido clasificadas erróneamente), mientras que un valor de 0,5 es lo esperado en un clasificador random.

La ventaja de ρ^{PN} sobre la precisión estándar es que es más robusta en cuanto desequilibrio de las clases dado que para una precisión estándar, la puntuación del clasificador de la clase mayoritaria es la frecuencia relativa, o prevalencia, de la clase mayoritaria que puede ser mucho más alto de 0,5 si las clases están desequilibradas. Además, también es más robusta en cuanto a la precisión F_1 dado que el valor del clasificador puede ser más alto que 0,5 si el conjunto de test está desequilibrado.

En este trabajo utilizaremos la siguiente medida de evaluación F_1:

$$F_1^{PN} = \frac{F_1^{Pos} + F_1^{Neg}}{2}$$

Figura 11. Ecuación $\boldsymbol{F_1}$

F_1^{PN} se define:

- obteniendo ρ^{Pos} como la fracción de muestras positivas que fueron predecidas que serían positivas. Fijándonos en la figura 9, esto significa:

$$\rho^{Pos} = \frac{PP}{PP + UP + NP}$$

Figura 12. Ecuación $\boldsymbol{\rho^{Pos}}$

- obteniendo π^{Pos} como la fracción de muestras positivas que se predijeron como positivas y lo son, eso se traduce como:

$$\pi^{Pos} = \frac{PP}{PP + UP + PN}$$

Figura 13. Ecuación $\boldsymbol{\pi^{Pos}}$

- obteniendo F_1^{Pos}:

$$F_1^{Pos} = \frac{2 * \pi^{Pos} * \rho^{Pos}}{\pi^{Pos} + \rho^{Pos}}$$

Figura 14. Ecuación $\boldsymbol{F_1^{Pos}}$

Para calcular la otra parte de la ecuación F_1^{PN}necesitamos la ecuación de F_1^{Neg} que es exactamente igual a las de las etiquetas en positivo, pero utilizando los valores de las negativas.

Capítulo 5

Experimentación

Esta sección pretende exponer los resultados obtenidos de los experimentos descritos en el capítulo anterior.

5.1. Experimento 1

Para comenzar, el experimento 1a que se llevó a cabo utilizando el conjunto de datos obtenidos del corpus TASS en castellano uniendo las etiquetas NONE y NEU en únicamente una sola etiqueta, pasando los datos etiquetados como NONE a estar etiquetados como NEU.

Primero, y dado que tenemos tres tipos de etiquetas, mostraremos las matrices de confusión correspondientes a cada tratamiento de los datos estemizados.

	N	**NEU**	**P**
N	2028	588	452
NEU	896	480	584
P	636	436	1496

Tabla 1. Matriz de confusión exp. 1a utilizando unigramas. Datos estemizados.

	N	**NEU**	**P**
N	1936	664	468
NEU	832	520	608
P	612	476	1480

Tabla 2. Matriz de confusión exp. 1a utilizando unigramas y bigramas. Datos estemizados.

	N	NEU	P
N	2028	588	452
NEU	896	480	584
P	636	436	1496

Tabla 3. Matriz de confusión exp. 1a utilizando 1-6 gramas. Datos estemizados.

	N	NEU	P
N	551	111	105
NEU	264	92	134
P	207	79	356

Tabla 4. Matriz de confusión exp. 1a utilizando unigramas y SkipGramas. Datos estemizados.

Una vez expuestas las matrices de confusión para los datos estemizados, nos disponemos a mostrar los resultados calculados a partir de estos datos.

	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Unigramas	0,58	0,6	0,59	51,82
Unigramas y bigramas	0,58	0,6	0,59	51,82
1-6 gramas	0,59	0,61	0,6	52,71
Unigramas y SkipGramas	0,58	0,62	0,6	52,61

Tabla 5. Resultados del exp. 1a con datos estemizados.

Observando los resultados cabe destacar que el mejor es el que obtenemos utilizando 1-6 gramas.

Una vez expuestas las matrices de confusión para los datos estemizados y los valores calculados a partir de ellas, pararemos a exponerles la de los datos no estemizados.

	N	NEU	P
N	1796	688	584
NEU	880	580	500
P	768	540	1260

Tabla 6. Matriz de confusión exp. 1a utilizando unigramas. Datos no estemizados.

	N	NEU	P
N	1936	600	532
NEU	940	492	528
P	792	420	1356

Tabla 7. Matriz de confusión exp. 1a utilizando unigramas y bigramas. Datos no estemizados.

	N	NEU	P
N	1996	552	520
NEU	1036	432	492
P	828	384	1392

Tabla 8. Matriz de confusión exp. 1a utilizando 1-6 gramas. Datos no estemizados.

	N	NEU	P
N	551	101	115
NEU	305	84	101
P	250	64	328

Tabla 9. Matriz de confusión exp. 1a utilizando unigramas y SkipGramas. Datos no estemizados.

Una vez expuestas las matrices de confusión para los datos no estemizados, nos disponemos a mostrar los resultados calculados a partir de estos datos.

	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Unigramas	0,51	0,55	0,53	47,87
Unigramas y bigramas	0,54	0,57	0,56	49,81
1-6 gramas	0,56	0,58	0,57	50,29
Unigramas y SkipGramas	0,55	0,58	0,57	50,71

Tabla 10. Resultados del exp. 1a con datos no estemizados.

Observando los resultados cabe destacar que el mejor es el que obtenemos utilizando unigramas y SkipGramas.

Para finalizar con el experimento 1a, hay que indicar que configuración de los mejores resultados, veremos que la experimentación utilizando datos estemizados y 1-6 gramas hace que mejore el resultado obtenido.

Los resultados del experimento 1a son similares al run3 con SVM presentado en [60] y también comparable con los resultados run1 y run2 que son aproximaciones basadas en Deep Learning con Redes Neuronales y embeddings.

El experimento 1a obtiene para cada una de las agrupaciones de N-Gramas y SkipGramas resultados similares a los obtenidos por [54] en el SemEval, que aunque son un poco inferiores, hay que tener en cuenta que aunque son corpus formados por tweets, los corpus son diferentes, y en este papel se ha utilizado otros recursos como el Jeffrey's lexicon que contiene un conjunto de palabras positivas y negativas y el NRC Emotion Lexicon que contiene un conjunto de palabras etiquetadas con un cero o un uno en función de la emoción que suscitan con ira, anticipación, asco, miedo, alegría, tristeza, sorpresa.

Continuaremos con el experimento 1b que se llevó a cabo utilizando el conjunto de datos obtenidos del corpus TASS en castellano, en este caso no se utilizarán los datos que estén etiquetados como NONE o NEU. En este experimento se ha obviado las etiquetas NEU y NONE del corpus TASS en castellano para equipararnos al corpus

CatSent que no emplea estas etiquetas y ver cómo funciona el clasificador con sólo estas dos etiquetas.

En este caso no es necesario mostrar las matrices de confusión, ya que únicamente tenemos etiquetas positivas y negativas. Para una mayor legibilidad, se mostrarán dos tablas de resultados, dependiendo de si los datos son usados estemizados o no estemizados.

	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Unigramas	0,67	0,76	0,72	72,32
Unigramas y bigramas	0,68	0,77	0,73	73,17
1-6 gramas	0,67	0,77	0,73	72,96
Unigramas y SkipGramas	0,67	0,76	0,73	72,60

Tabla 11. Resultados del exp. 1b con datos estemizados.

Observando los resultados podemos fijarnos en que los resultados utilizando datos estemizados mejora si utilizamos unigramas y bigramas.

	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Unigramas	0,63	0,75	0,7	70,19
Unigramas y bigramas	0,64	0,75	0,71	70,62
1-6 gramas	0,62	0,75	0,7	70,26
Unigramas y SkipGramas	0,64	0,75	0,7	70,4

Tabla 12. Resultados del exp. 1b con datos no estemizados.

Observando los resultados cabe destacar que la mejor configuración es la que obtenemos utilizando unigramas y bigramas.

Si comparamos los mejores resultados observamos que la experimentación utilizando datos estemizados junto a unigramas y bigramas hace que mejore el resultado obtenido.

A continuación, se muestran los resultados del corpus CatSent en catalán con SVM [61] donde se obtiene un 74% de F-Score y son similares al 73% obtenido en castellano del corpus TASS con sólo etiquetas positivas y negativas, teniendo en cuenta que el corpus es mucho más pequeño.

Table 7 SVM results

SVM	Precision	Recall	F-score
Negative	74%	75%	75%
Positive	74%	73%	74%
Total	74%	74%	74%

Figura 15. Resultados del corpus CatSent en catalán empleando como clasificador SVM.

Fuente: [61]

5.2. Experimento 2

Continuaremos con el experimento 2 que se llevó a cabo utilizando el conjunto de datos obtenidos del corpus TASS en castellano junto con los datos traducidos al castellano del corpus CatSent, en este caso no se utilizarán los datos que estén etiquetados como NONE o NEU.

En este caso no es necesario mostrar las matrices de confusión, ya que únicamente tenemos etiquetas positivas y negativas. Para una mayor legibilidad, se mostrarán dos tablas de resultados, dependiendo de si los datos son usados estemizados o no estemizados.

	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Unigramas	0,64	0,52	0,59	58,91
Unigramas y bigramas	0,65	0,53	0,60	60,33
1-6 gramas	0,66	0,57	0,62	62,24
Unigramas y SkipGramas	0,65	0,55	0,61	60,54

Tabla 13. Resultados del exp. 2 con datos estemizados.

Observando los resultados podemos fijarnos en que los resultados utilizando datos estemizados mejora si utilizamos 1-6 gramas.

	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Unigramas	0,64	0,51	0,58	58,41
Unigramas y bigramas	0,64	0,51	0,58	58,91
1-6 gramas	0,65	0,55	0,61	60,97
Unigramas y SkipGramas	0,66	0,54	0,61	60,68

Tabla 14. Resultados del exp. 2 con datos no estemizados.

Observando los resultados cabe destacar que la mejor configuración es la que obtenemos utilizando 1-6 gramas.

Si comparamos los mejores resultados de ambos experimentos, veremos que la experimentación utilizando datos estemizados utilizando 1-6 gramas hace que mejore el resultado obtenido.

Como se puede observar los datos mejoran a los resultados del experimento 1a al aumentar el número de muestras de entrenamiento, hay que hacer notar que incluir el corpus CatSent sólo incrementa el número de muestras positivas y negativas, pero no las Neutras (NEU) o NONE.

5.3. Experimento 3

El experimento 3 se llevó a cabo utilizando el conjunto de datos obtenidos del corpus TASS traducido en catalán uniendo las etiquetas NONE y NEU en únicamente una sola etiqueta, pasando los datos etiquetados como NONE a estar etiquetados como NEU.

El conjunto de entrenamiento se compone de 4032 tweets, ya que se han utilizado todos los traductores para que las pequeñas diferencias entre ellos enriquezcan el vocabulario del corpus.

Primero, y dado que tenemos tres tipos de etiquetas, mostraremos las matrices de confusión correspondientes a cada tratamiento de los datos estemizados.

	N	**NEU**	**P**
N	1881	740	517
NEU	888	551	521
P	786	521	1261

Tabla 15. Matriz de confusión exp. 3 utilizando unigramas. Datos estemizados.

	N	**NEU**	**P**
N	1915	634	519
NEU	903	563	494
P	707	448	1413

Tabla 16. Matriz de confusión exp. 3 utilizando unigramas y bigramas. Datos estemizados.

	N	**NEU**	**P**
N	1895	628	545
NEU	893	577	490
P	703	433	1432

Tabla 17. Matriz de confusión exp. 3 utilizando 1-6 gramas. Datos estemizados.

	N	NEU	P
N	1942	630	496
NEU	399	245	220
P	745	450	1373

Tabla 18. Matriz de confusión exp. 3 utilizando unigramas y SkipGramas. Datos estemizados.

Una vez expuestas las matrices de confusión para los datos estemizados, nos disponemos a mostrar los resultados calculados a partir de estos datos.

	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	SVM Accuracy Score
Unigramas	0,52	0,55	0,54	47,7
Unigramas y bigramas	0,57	0,58	0,58	51,22
1-6 gramas	0,57	0,58	0,58	51,4
Unigramas y SkipGramas	0,59	0,63	0,61	54,77

Tabla 19. Resultados del exp. 3 con datos estemizados.

Observando los resultados cabe destacar que el mejor es el que obtenemos utilizando unigramas y SkipGramas.

Una vez expuestas las matrices de confusión para los datos estemizados y los valores calculados a partir de ellas, pararemos a exponerles la de los datos no estemizados.

	N	NEU	P
N	1873	691	504
NEU	902	610	448
P	791	510	1267

Tabla 20. Matriz de confusión exp. 3 utilizando unigramas. Datos no estemizados.

	N	NEU	P
N	1953	621	494
NEU	911	577	472
P	758	417	1393

Tabla 21. Matriz de confusión exp. 3 utilizando unigramas y bigramas. Datos no estemizados.

	N	NEU	P
N	2000	590	478
NEU	927	540	493
P	804	394	1370

Tabla 22. Matriz de confusión exp. 3 utilizando 1-6 gramas. Datos no estemizados.

	N	NEU	P
N	1970	616	482
NEU	395	269	200
P	823	402	1343

Tabla 23. Matriz de confusión exp. 3 utilizando unigramas y SkipGramas. Datos no estemizados.

Una vez expuestas las matrices de confusión para los datos no estemizados, nos disponemos a mostrar los resultados calculados a partir de estos datos.

	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Unigramas	0,53	0,56	0,55	49,37
Unigramas y bigramas	0,57	0,58	0,58	51,65
1-6 gramas	0,56	0,59	0,58	51,47
Unigramas y SkipGramas	0,58	0,63	0,61	55,11

Tabla 24. Resultados del exp. 3 con datos no estemizados.

Observando los resultados cabe destacar que el mejor es el que obtenemos utilizando unigramas y SkipGramas.

Para finalizar con el experimento 3, hay que indicar que configuración da los mejores resultados, veremos que la experimentación utilizando datos estemizados y 1-6 gramas hace que mejore el resultado obtenido.

Como se puede apreciar en la Tabla 19 de resultados al traducir el TASS al catalán no conlleva una bajada notable del Accuracy, e incluso para Unigramas y SkipGramas mejora los resultados, lo cual valida la traducción hecha por los traductores automáticos.

5.4. Experimento 4

Continuaremos con el experimento 4 que se llevó a cabo utilizando el conjunto de datos obtenidos del corpus TASS traducidos al catalán junto con los datos obtenidos del corpus CatSent, en este caso no se utilizarán los datos que estén etiquetados como NONE o NEU.

En este caso no es necesario mostrar las matrices de confusión, ya que únicamente tenemos etiquetas positivas y negativas. Para una mayor legibilidad, se mostrarán dos tablas de resultados, dependiendo de si los datos son usados estemizados o no estemizados.

	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Unigramas	0,64	0,55	0,6	60,13
Unigramas y bigramas	0,65	0,55	0,61	60,81
1-6 gramas	0,65	0,57	0,61	61,3
Unigramas y SkipGramas	0,63	0,45	0,56	60,54

Tabla 25. Resultados del exp. 4 con datos estemizados.

Observando los resultados podemos fijarnos en que los resultados utilizando datos estemizados mejora si utilizamos 1-6 gramas.

	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Unigramas	0,64	0,55	0,60	60,34
Unigramas y bigramas	0,65	0,55	0,61	60,73
1-6 gramas	0,65	0,55	0,61	60,68
Unigramas y SkipGramas	0,63	0,45	0,56	55,62

Tabla 26. Resultados del exp. 4 con datos no estemizados.

Observando los resultados cabe destacar que la mejor configuración es la que obtenemos utilizando unigramas y bigramas.

Si comparamos los mejores resultados de ambos experimentos, veremos que la experimentación utilizando datos estemizados utilizando 1-6 gramas hace que mejore el resultado obtenido.

Al aumentar el número de muestras de entrenamiento con el corpus CatSent ha enriquecido nuestro corpus de entrenamiento y se han mejorado los resultados del experimento 3.

5.5. Experimento 5

Continuaremos con el experimento 5 que se llevó a cabo utilizando el conjunto de datos obtenidos del corpus CatSent.

En este experimento evaluamos el corpus CatSent en catalán creando nosotros una partición de training 75% y otra de test 25% de forma aleatoria ya que los autores no especifican estas particiones en su trabajo [61].

En este caso no es necesario mostrar las matrices de confusión, ya que únicamente tenemos etiquetas positivas y negativas. Para una mayor legibilidad, se mostrarán dos

tablas de resultados, dependiendo de si los datos son usados estemizados o no estemizados.

	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Unigramas	0,79	0,78	0,79	78,68
Unigramas y bigramas	0,79	0,78	0,79	78,68
1-6 gramas	0,79	0,79	0,79	78,6
Unigramas y SkipGramas	0,70	0,79	0,79	78,96

Tabla 27. Resultados del exp. 5 con datos estemizados.

Observando los resultados podemos fijarnos en que los resultados utilizando datos estemizados mejora si utilizamos unigramas y SkipGramas.

	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Unigramas	0,79	0,78	0,79	78,75
Unigramas y bigramas	0,79	0,79	0,79	79,2
1-6 gramas	0,78	0,78	0,78	78,1
Unigramas y SkipGramas	0,8	0,8	0,8	79,65

Tabla 28. Resultados del exp. 5 con datos no estemizados.

Observando los resultados cabe destacar que la mejor configuración es la que obtenemos utilizando unigramas y SkipGramas.

Si comparamos los mejores resultados de ambos experimentos, veremos que la experimentación utilizando datos no estemizados utilizando unigramas y SkipGramas hace que mejore el resultado obtenido.

Como vemos los resultados mejoran los del artículo que son del 74% [61] (ver figura 15).

5.6. Experimento 6

Continuaremos con el experimento 6 que se llevó a cabo utilizando el conjunto de datos obtenidos del corpus CatSent traducidos al castellano.

En este caso no es necesario mostrar las matrices de confusión, ya que únicamente tenemos etiquetas positivas y negativas. Para una mayor legibilidad, se mostrarán dos tablas de resultados, dependiendo de si los datos son usados estemizados o no estemizados.

	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Unigramas	0,85	0,84	0,85	84,37
Unigramas y bigramas	0,9	0,9	0,9	90,21
1-6 gramas	0,9	0,91	0,91	90,53
Unigramas y SkipGramas	0,97	0,97	0,97	96,76

Tabla 29. Resultados del exp. 6 con datos estemizados.

Observando los resultados podemos fijarnos en que los resultados utilizando datos estemizados mejora si utilizamos unigramas y SkipGramas.

	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Unigramas	0,87	0,87	0,87	87,07
Unigramas y bigramas	0,91	0,91	0,91	90,74
1-6 gramas	0,91	0,91	0,91	90,72
Unigramas y SkipGramas	0,97	0,97	0,97	97,37

Tabla 30. Resultados del exp. 6 con datos no estemizados.

Observando los resultados cabe destacar que la mejor configuración es la que obtenemos utilizando unigramas y SkipGramas.

Si comparamos los mejores resultados de ambos experimentos, veremos que la experimentación utilizando datos no estemizados utilizando unigramas y SkipGramas hace que mejore el resultado obtenido.

Como se puede observar al traducir al castellano el corpus CatSent nos encontramos unos resultados muy elevados, que rondan el 97%. Esto nos hace pensar que el corpus en catalán tiene apostrofes y frases hechas que hacen que los traductores al castellano creen un corpus con un lenguaje mucho más formal.

5.7. Resumen experimentos

Para concluir, en las siguientes tablas podemos ver los mejores resultados de cada experimento, una por cada idioma.

	N-Gramas	**¿Están los datos estemizados?**	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Experimento 1a	1-6 gramas	SI	0,59	0,61	0,6	52,71
Experimento 1b	Unigramas y bigramas	SI	0,68	0,77	0,73	73,13
Experimento 2	1-6 gramas	SI	0,66	0,57	0,62	62,24
Experimento 6	Unigramas y SkipGramas	NO	0,97	0,97	0,97	97,37

Tabla 31. Resultados de los experimentos en castellano.

	N-Gramas	**¿Están los datos estemizados?**	F_1^{Pos}	F_1^{Neg}	F_1^{PN}	**SVM Accuracy Score**
Experimento 3	1-6 gramas	SI	0,57	0,58	0,58	51,4
Experimento 4	1-6 gramas	SI	0,65	0,57	0,61	61,3
Experimento 5	Unigramas y skypgramas	NO	0,8	0,8	0,8	79,65

Tabla 32. Resultados de los experimentos en catalán.

Como se puede observar en los resultados hemos conseguido traducir el corpus TASS a catalán con éxito y los experimentos nos muestran que los resultados son similares a los que se obtienen a los obtenidos en castellano. Además, obtenemos resultados similares a otros trabajos realizados con el mismo corpus TASS y con otros como el SemEval, por último, la incorporación del corpus de tweets CatSent, mejora los resultados obtenidos y nos valida el uso de los traductores automáticos.

Capítulo 6

Conclusiones y trabajo futuro

Este trabajo presenta los resultados del proceso que se ha seguido para conseguir obtener un modelo híbrido de análisis de sentimiento en tweets que nos permita analizar datos para los idiomas español y catalán.

Los corpus utilizados para el análisis de sentimientos con tweets son difíciles de procesar ya que en Twitter podemos encontrar muchos tweets con abreviaturas y emojis que no están debidamente tratados; como consecuencia esto dificulta la, ya de por sí, ambigua interpretación de los datos dificultando en gran medida el procesamiento de los datos para su posterior análisis.

A pesar de lo mencionado, podemos considerar que se han cumplido los objetivos descritos al inicio del trabajo, los cuales están indicados en el comienzo de la memoria, ya que conseguimos desarrollar un corpus en catalán con una extensión considerable de 213612 tweets en catalán y español. Además, cabe añadir que también se ha conseguido, a través de este corpus, obtener un analizador SVM que nos permite trabajar en ambos idiomas.

6.1. Relación con los estudios cursados

Durante todo el proceso del trabajo hemos podido utilizar una gran cantidad de conocimientos adquiridos durante los cuatro años que dura el grado en ingeniería informática, gracias a los cuales nos ayudaron a realizar el procesamiento de los datos y su posterior análisis. Daremos una pequeña explicación sobre los conocimientos que pensábamos fueron más relevantes a lo largo del proyecto.

Si empezamos desde el principio, debemos mencionar la asignatura de Introducción a la Informática y la Programación y la asignatura de Programación, ya que nos dieron

las bases de nuestro conocimiento, introduciéndonos los conceptos de listas de objetos, bucles, etcétera.

Por otro lado, y siendo más específicos, debemos hablar de la asignatura de Estructura de Datos y Algoritmos, ya que adquirimos los conocimientos para utilizar las listas y los diccionarios, que fueron de gran utilidad durante el desarrollo del trabajo.

Por último, en cuanto a los conocimientos adquiridos para el análisis de sentimiento utilizando un SVM y el aprendizaje automático, debemos mencionar las asignaturas de Percepción y Aprendizaje Automático, pues nos introdujeron estos conceptos y otros relacionados con la inteligencia artificial.

6.2. Trabajo futuro

Con lo mencionado anteriormente sobre los emojis y abreviaturas que modifican sustancialmente el significado, podemos decir que nos abre una nueva perspectiva para tener en cuenta para la mejora del sistema. Está mejora vendría a partir de una elaboración de diccionarios de emojis (tanto textual como visuales) y de abreviaturas que nos permitan no perder información a la hora de procesar el tweet para su posterior análisis.

También sería interesante seguir entrenando nuestro SVM para conseguir optimizar todavía más los parámetros utilizados a la hora de llevar a cabo el análisis.

Bibliografía

[1] V. A. Esteve, F. P. Santamaria y L. F. H. Oliver, Creación de corpus de artículos de prensa y categorización de noticias, Valencia, 2019.

[2] «TASS: Taller de análisis semántico en la SEPLN 2017,» [En línea]. Available: http://tass.sepln.org/2017/#about. [Último acceso: 29 Agosto 2021].

[3] P. Balaguer, «Github: Catalan Sentiment Analysis,» 3 Diciembre 2017. [En línea]. Available: https://github.com/pbalaguer19/catalan-sentiment-analysis. [Último acceso: 29 Agosto 2021].

[4] GVA, «Conselleria de Educación, Cuiltura y Deporte,» [En línea]. Available: https://ceice.gva.es/es/web/dgplgm/salt. [Último acceso: 29 Agosto 2021].

[5] SoftCatalà, «SoftCatalà,» [En línea]. Available: https://www.softcatala.org/traductor/. [Último acceso: 29 Agosto 2021].

[6] M. L. Forcada, B. I. Bonev, S. O. Rojas, J. A. P. Ortiz, G. R. Sánchez, F. S. Martínez, C. Armentano-Oller, M. A. Montava y F. M. Tyers, «Apertium,» 10 Marzo 2010. [En línea]. Available: https://wiki.apertium.org/w/images/d/d0/Apertium2-documentation.pdf. [Último acceso: 29 Agosto 2021].

[7] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Traductor_de_Google. [Último acceso: 2021 Agosto 2021].

[8] M. d. C. Ugalde, «El lenguaje caracterización de sus dos formas fundamentales: el código oral y el código escrito,» vol. 12, nº 2, pp. 47-56, 1988.

[9] J. G. Brookshear, Teoría de la computación, Addison-Wesley Iberoamericana, 1993.

[10] A. C. Vásquez, H. V. huerta, J. P. Quispe y A. M. Huayna, «Procesamiento de lenguaje natural,» vol. 6, nº 2, pp. 45-54, Diciembre 2009.

[11] A. Gelbukh, «Procesamiento de Lenguaje Natural y sus,» *Komputer Sapiens,* vol. 1, nº 2, pp. 6-32, 2010.

[12] J. Villares, *Aplicaciones del procesamiento del lenguaje natural en la recuperación de información en español,* Coruña: Universidade da Coruña, 2005.

[13] B. Liu, Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, 2012.

[14] T. Nasukawa y J. Yi, «Sentiment Analysis: Capturing Favorability Using Natural Language Processing,» 2003.

[15] K. Dave, S. Lawrence y D. Pennock, «Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews,» *Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews,* vol. 775152, 2003.

[16] S. Morinaga, K. Yamanishi, K. Tateishi y T. Fukushima, «Mining Product Reputations on the Web,» 2002.

[17] S. R. Das y M. Y. Chen, «Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web».

[18] B. Pang, L. Lee y S. Vaithyanathan, «Thumbs up? Sentiment Classification Using Machine Learning Techniques,» vol. 10, 2002.

[19] A. H. Y. TONG, M. EVANGELISTA, A. B. PARSONS, H. XU, G. D. BADER, N. PAGÉ, M. ROBINSON, S. RAGHIBIZADEH, C. W. V. HOGUE, H. BUSSEY, B. ANDREWS, M. TYERS y C. BOONE, «Systematic genetic analysis with ordered arrays of yeast deletion mutants. Science.,» 2001.

[20] P. Turney, «Thumbs Up or Thumbs Down? {S}emantic Orientation Applied to Unsupervised Classification of Reviews,» 2002.

[21] V. Hatzivassiloglou y J. M. Wiebe, «Effects of Adjective Orientation and Gradability on Sentence Subjectivity,» vol. 1, 2000.

[22] H. a. McKeown, «Predicting the Semantic Orientation of Adjectives,» 1997.

[23] M. A. Hearst, «Automatic Acquisition of Hyponyms from Large Text Corpora,» 1992.

[24] J. M. Wiebe, «Tracking Point of View in Narrative,» 1994.

[25] J. M. Wiebe, R. F. Bruce y T. P. O'Hara, «Development and Use of a Gold-Standard Data Set for Subjectivity Classifications,» 1999.

[26] A. Jungherr, «Twitter use in election campaigns: A systematic literature review,» *Journal of Information Technology & Politics,* vol. 13, pp. 72-91, 2016.

[27] T. Baviera, *Técnicas para el análisis del sentimiento en Twitter: Aprendizaje Automático Supervisado y SentiStrength,* Universitat de Valencia, 2017.

[28] F. Bravo-Marquez, M. Mendoza y B. Poblete, «Meta-level sentiment models for big social data analysis,» *Knowledge-Based Systems,* vol. 69, pp. 86-99, 2014.

[29] W. Medha, A. Hassan y H. Korashy, «Sentiment analysis algorithms and applications: A survey,» *Ain Shams Engineering Journal,* vol. 5, nº 4, pp. 1093-1113, 2014.

[30] «Search Data Center,» Enero 2007. [En línea]. Available: https://searchdatacenter.techtarget.com/es/definicion/Aprendizaje-automatico-machine-learning. [Último acceso: 29 Agosto 2021].

[31] «LUCA,» [En línea]. Available: https://luca-d3.com/es/data-speaks/diccionario-tecnologico/python-lenguaje.

[32] «NLTK,» [En línea]. Available: https://www.nltk.org. [Último acceso: 29 Agosto 2021].

[33] «Freeling,» [En línea]. Available: http://nlp.lsi.upc.edu/freeling/node/1. [Último acceso: 2021 08 29].

[34] «Meaning Cloud,» [En línea]. Available: https://www.meaningcloud.com/es/como-puedes-usar-meaningcloud. [Último acceso: 29 Agosto 2021].

[35] «Meaning Cloud: Analítica y minería de textos gratis,» [En línea]. Available: https://www.meaningcloud.com/es/productos/analitica-y-mineria-de-textos-gratis.

[36] «Google Cloud,» [En línea]. Available: https://cloud.google.com/natural-language/docs#docs. [Último acceso: 29 Agosto 2021].

[37] «Microsoft,» [En línea]. Available: https://docs.microsoft.com/es-es/azure/cognitive-services/text-analytics/overview. [Último acceso: 29 Agosto 2021].

[38] «IBM Watson Natural Language Undertanding,» [En línea]. Available: https://www.ibm.com/cloud/watson-natural-language-understanding. [Último acceso: 29 Agosto 2021].

[39] M. Belica, «PyPi,» [En línea]. Available: https://pypi.org/project/sumy/. [Último acceso: 29 Agosto 2021].

[40] L. Vanderwende, H. Suzuki, C. Brocketta y A. Nenkova, «Beyond SumBasic: Task-Focused Summarization with Sentence Simplification and Lexical Expansion,» vol. 43, nº 6, pp. 1606-1618, 2007.

[41] A. Haghighi y L. Vanderwende, «Exploring Content Models for Multi-Document Summarization,» de *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado, 2009.

[42] «Real Academia de la lengua Español,» [En línea]. Available: https://dle.rae.es/corpus. [Último acceso: 29 Agosto 2021].

[43] «SemEval,» [En línea]. Available: https://semeval.github.io.

[44] «SemEval-2016 Task 4: Sentiment Analysis in Twitter,» [En línea]. Available: https://alt.qcri.org/semeval2016/task4/.

[45] M. Wiegand, M. Siegel y J. Ruppenhofer, «Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language,» 2018.

[46] «TASS: Taller de análisis semántico en la SEPLN,» [En línea]. Available: http://tass.sepln.org.

[47] «Requests: HTTP for Humans,» [En línea]. Available: https://docs.python-requests.org/en/master/. [Último acceso: 29 Agosto 2021].

[48] «MDN Web Docs,» [En línea]. Available: https://developer.mozilla.org/es/docs/Web/HTTP/Messages. [Último acceso: 29 Agosto 2021].

[49] «Python Docs: Subprocess management,» [En línea]. Available: https://docs.python.org/3/library/subprocess.html. [Último acceso: 29 Agosto 2021].

[50] «Python Docs: Implementación mínima del DOM,» [En línea]. Available: https://docs.python.org/es/3.10/library/xml.dom.minidom.html. [Último acceso: 29 Agosto 2021].

[51] R. Boulton, «PyPi,» [En línea]. Available: https://pypi.org/project/PyStemmer/. [Último acceso: 29 Agosto 2021].

[52] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/N-grama. [Último acceso: 30 Agosto 2021].

[53] M. Struwig, «Not So Big Data Blog,» 2 Junio 2019. [En línea]. Available: https://notsobigdatablog.com/2019/01/02/what-is-a-skipgram/. [Último acceso: 3 Septiembre 2021].

[54] V. Martínez, F. Pla y L.-F. Hurtado, «DSIC-ELIRF at SemEval-2016 Task 4: Message Polarity Classification in Twitter using a Support Vector Machine Approach,» pp. 198-201, Junio 2016.

[55] «MathWorks,» [En línea]. Available: https://es.mathworks.com/discovery/supervised-learning.html. [Último acceso: 30 Agosto 2021].

[56] «MathWorks,» [En línea]. Available: https://es.mathworks.com/discovery/support-vector-machine.html. [Último acceso: 30 Agosto 2021].

[57] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Scikit-learn. [Último acceso: 30 Agosto 2021].

[58] G. Bedi, «A guide to Text Classification(NLP) using SVM and Naive Bayes with Python,» 9 Noviembre 2018. [En línea]. Available: https://medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-bayes-with-python-421db3a72d34.

[59] G. A. BETANCOURT, «LAS MÁQUINAS DE SOPORTE VECTORIAL (SVMs),» *Scientia et Technica,* vol. 6, nº 27, 2005.

[60] L. Hurtado Oliver, F. Pla y J. González Barba, «ELiRF-UPV at TASS 2017: Sentiment Analysis in Twitter based on Deep Learning,» 2017.

[61] P. Balaguer, I. Teixidó, J. Vilaplana, J. Mateo, J. Rius y F. Solsona, «CatSent: a Catalan sentiment analysis website,» *Multimed Tools Appl,* vol. 78, p. 28137–28155, 2019.

[62] «Wild Code School,» 20 Enero 2021. [En línea]. Available: https://www.wildcodeschool.com/es-ES/blog/tipos-de-lenguajes-de-programacion. [Último acceso: 29 Agosto 2021].