Estimación de la polaridad reputacional mediante contextual word embeddings

Trabajo Fin de Máster

Eduardo Aarón Fernández Orallo

Trabajo de investigación para el

Máster en Lenguajes y Sistemas Informáticos

Universidad Nacional de Educación a Distancia

Dirigido por

Julio Antonio Gonzalo Arroyo

Junio 2020

Agradecimientos

Le doy gracias a mi novia y familia por su paciencia y apoyo en los momentos difíciles.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1. Motivación

Las redes sociales son un fenómeno que se ha expandido recientemente en todo el mundo y rápidamente atrajo a miles de millones de usuarios. Esta forma de comunicación electrónica a través de plataformas de redes sociales permite a los usuarios generar su contenido y compartirlo en varias formas de información, palabras, imágenes, audio y vídeos. Por lo tanto, la informática social se forma como un área emergente de investigación y desarrollo que incluye una amplia gama de temas como la semántica web, la inteligencia artificial, el procesamiento del lenguaje natural, el análisis de redes y el análisis de *Big Data* Farzindar and Inkpen (2013).

Twitter es una red social pensada para ofrecer una comunicación rápida. Más de 140 millones de usuarios activos publican más de 400 millones de tweets de 140 caracteres cada día. La velocidad y facilidad de publicación de Twitter lo han convertido en un importante medio de comunicación para personas de todos los ámbitos de la vida. Twitter ha desempeñado un papel destacado en eventos sociopolíticos, como la Primavera Árabe y el "*Wall Street movement*". Twitter también se ha utilizado para informar sobre desastres naturales, como el huracán Sandy Shamanth Kumar (2014).

La amplia y rápida difusión de las opiniones en Twitter ha hecho que el análisis y el monitoreo de medios sociales se haya convertido en una parte integral de la estrategia de marketing de las empresas de todo el mundo Peetz et al. (2016). En especial resulta de vital importancia las opiniones de personajes famosos, que son seguidos por muchas personas (*followers*) y dan lugar a una amplia difusión. Lo que puede influir en la opinión que el gran público tiene sobre las entidades sobre las que se opina.

Por las razones expuestas resulta de crucial importancia para empresas y marcas conocer el impacto que tiene un tweet en su reputación, lo que se conoce como polaridad reputacional. La polaridad reputacional de un tweet es una medida de cómo el tweet influye en la reputación de una marca o empresa Peetz

et al. (2016). La polaridad de reputación no se debe confundir con el sentimiento que expresa el tweet, aunque estén relacionados normalmente. Por ejemplo, un texto que expresa sarcasmo puede tener un sentimiento positivo o neutral, pero por el contrario se pretende expresar una opinión negativa.

El proceso de clasificación de tweets según su polaridad supone un problema de mayor complejidad en relación con otros tipos de clasificación de textos, como puede ser la clasificación de sentimientos. Para este proceso de clasificación, el sentimiento que exprese el tweet será importante pero no determinará la polaridad en sí. Influirán otros factores como el sarcasmo o la entidad sobre la que se está tratando (un tweet puede criticar a una marca de la competencia, lo que resulta beneficioso).

Otros factores para tener en cuenta y que añaden más dificultad, son problemas relacionados con la naturaleza de los tweets como la puntuación y las mayúsculas inconsistentes (o ausentes), que pueden dificultar la detección de los límites de las oraciones. Los emoticonos, la ortografía incorrecta o no estándar y las abreviaturas "desenfrenadas" complican la tokenización y el etiquetado de parte del discurso, entre otras tareas Farzindar and Inkpen (2013).

En el presente trabajo se considera la tarea de determinar automáticamente la polaridad reputacional de tweets. Para ello se empleará un modelo de redes neuronales empleando la técnica de *contextual word embeddings*. Como modelo de *contextual word embeddings* se ha seleccionado BERT (*Bidirectional Encoder Representations from Transformers*) Devlin et al. (2018), creado por Google, el cual ha demostrado ofrecer una gran mejora en diversos problemas de procesamiento del lenguaje natural, tales como clasificación de textos, reconocimiento de entidades nombradas, etc. BERT está basado en *transformers*, una novedosa técnica empleada en la mayoría de los modelos del lenguaje de última generación, que ha ofrecido mejoras sustanciales respecto a los métodos empleados anteriormente. Los resultados obtenidos por BERT lo han convertido en el estado del arte en lo que concierne a procesamiento del lenguaje natural. Hecho que ha llevado a varias empresas tecnológicas e instituciones a crear sus propios modelos basados en BERT.

Como conjunto de datos de entrenamiento y prueba se empleará RepLab 2013. El conjunto de datos RepLab 2013 utiliza datos de Twitter en inglés y español (más de 142000 tweets). El equilibrio entre ambos idiomas depende de la disponibilidad de datos para cada una de las entidades incluidas en el conjunto de datos. El corpus consiste en una colección de tweets que se refieren a un conjunto seleccionado de 61 entidades de cuatro dominios: automotriz, banca, universidades y música/artistas. La selección del dominio se realizó para ofrecer una variedad de escenarios para estudios de reputación Amigó et al. (2013).

1.2. Objetivos y plan de trabajo

En el presente trabajo se pretende estudiar cómo y cuánto puede mejorarse la predicción la polaridad reputacional empleando la técnica de *contextual word embeddings*. Como modelo de *contextual word embeddings* se ha decidido

emplear BERT (*Bidirectional Encoder Representations from Transformers*), por ser el más usados y uno de los que mejores resultados ha ofrecido en diversas tareas de PLN (procesamiento del lenguaje natural). Como conjunto de datos de entrenamiento y testeo se usará el conjunto de datos Replab 2013. Este *dataset* está formado por más de 142000 tweets en inglés y en español. Fue diseñado específicamente para la tarea de monitorear la reputación de entidades y en la actualidad corresponde una de las mayores colecciones de datos anotadas para tal cometido.

El método que se propone en este trabajo está formado por una red neuronal compuesta de las diversas capas que forman el modelo BERT, más la capa de salida necesaria para poder clasificar (*softmax*). Existen diferentes variantes de BERT y diferentes posibilidades de mejora, no obstante, emplearemos la versión estándar, sin modificaciones, con la que veremos que es suficiente para mejorar la tarea que nos atañe. Solo se variará la versión de BERT en base al idioma para el que fue desarrollado.

Para probar el método propuesto se realizarán diferentes experimentos donde se entrenará el modelo empleando el conjunto de datos de RepLab de entrenamiento y se testará con el conjunto de datos de test. Las variantes de estos experimentos dependerán del idioma de los tweets, el dominio de estos y la versión de BERT, que dependerá del idioma de los tweets (*bert-base-cased* y *bert-base-multilingual-cased*). Los experimentos serán evaluados empleando diversas métricas que permitirán comparar los resultados con los obtenidos en estudios anteriores.

1.3. Estructura del documento

La estructura que seguirá el presente trabajo será la expuesta a continuación:

Capítulo 1. Introducción. En este capítulo se describen los principales motivos que han llevado a la realización de este trabajo, así como la problemática y el estado actual del análisis de la polaridad reputacional. Además, se indican los objetivos que se pretenden cumplir.

Capítulo 2. Estado del arte. En este capítulo se describe en mayor detalle el análisis de polaridad reputacional, presentando su origen y su historia hasta los últimos estudios realizados, indicando en cada caso las debilidades y ventajas de las distintas técnicas empleadas. Además, describiremos la técnica de *contextual word embeddings* y nos centraremos en el modelo más empleado en estos momentos que emplea dicha técnica: BERT.

Capítulo 3. Diseño experimental. En este capítulo se describirá el *dataset* empleado para hacer los experimentos, así como las métricas de evaluación empleadas y las distintas variantes del sistema que se van a probar sobre ese *dataset*. Por otro lado, se describirá en detalle el método planteado para la clasificación de la polaridad reputacional.

Capítulo 4. Resultados experimentales. En este capítulo se mostrarán los resultados obtenidos en los experimentos realizados y se analizarán de forma cuantitativa y cualitativa.

Capítulo 5. Discusión. En este capítulo se discuten los resultados mostrados en el apartado anterior, comparándolos con estudios anteriores y se mostrarán las fortalezas y debilidades de la técnica propuesta. Además, se intentará comprender porque la técnica empleada llega a los resultados mostrados.

Capítulo 6. Conclusiones y trabajo futuro. En este capítulo se recopilan las conclusiones extraídas y se proponen líneas de investigación futuras.

Capítulo 2

Estado del arte

Existe una gran cantidad de métodos y técnicas, desarrolladas con anterioridad, para la clasificación de textos, aplicadas para diferentes tareas en las que el trabajo de clasificar "a mano" resultaría demasiado tedioso.

En el caso de microtextos como los tweets, el problema de clasificación se ha centrado en el análisis de sentimientos. Una evolución lógica surgida a partir del análisis de sentimientos es el análisis de polaridad reputacional.

A pesar de que existen diferencias sustanciales entre la polaridad reputacional y el análisis de los sentimientos, las dos tareas tienen algunas similitudes. En esta sección, primero presentamos trabajos relacionados con el análisis de sentimientos y posteriormente mostraremos trabajos previos sobre el análisis de reputación y, más específicamente, sobre el análisis de polaridad reputacional. Posteriormente plantearemos la técnica *contextual word embeddings* con la que se pretende tratar el problema de la polaridad reputacional.

La realización de este capítulo se ha apoyado principalmente en los trabajos realizados por Giachanou et al. (2019), Medhat et al. (2014) y Vaswani et al. (2017).

2.1. Análisis de opiniones y sentimientos.

El análisis de sentimientos se centra en determinar la polaridad de los sentimientos (positiva, neutral, negativa) que se expresan en un texto. Los primeros estudios se centraron principalmente en la minería de opinión sobre reseñas de productos Turney (2002) y de películas Pang et al. (2002).

Con la aparición de plataformas de microblogging, en especial Twitter, donde millones de usuarios exponen sus opiniones sobre diferentes dominios y entidades en forma de textos, surgió rápidamente el interés de los investigadores por aplicar técnicas para extraer sentimientos y emociones de estos textos Bollen et al. (2011); Davidov et al. (2010); Go et al. (2009); Kiritchenko et al. (2014).

Las técnicas de clasificación de sentimientos se pueden dividir,

aproximadamente, en tres clases en función del enfoque que se le dé: enfoque de aprendizaje automático, enfoque basado en léxico y enfoque híbrido Medhat et al. (2014).

Los enfoques basados en aprendizaje automático, emplea algoritmos de ML (*machine learning*) para resolver la clasificación de sentimientos como un problema de clasificación de textos empleando diferentes *features* (características de los textos que se pueden emplear como entrada a un modelo de ML) de los textos. Por su parte, el problema de clasificación de textos se puede definir como: dado un conjunto de registros de entrenamiento $D = \{X_1, X_2, ..., X_n\}$, donde cada registro es etiquetado con una clase, un modelo ML de clasificación es entrenado para relacionar las *features* de los textos, con cada una de las etiquetas de clase. Posteriormente el modelo se emplea para predecir las etiquetas de clase de los textos no etiquetados. Las *features* empleadas en la clasificación de textos, van desde el estudio de la frecuencia de aparición de ciertos términos (n-gramos, por ejemplo), hasta el empleo de características sintácticas y/o lingüísticas. Agarwal et al. (2011) realizó un estudio donde se analizan diferentes *features* de textos para el análisis de sentimientos.

La evolución de los algoritmos de ML y en especial de la computación que permite ejecutar tales algoritmos, ha permitido que investigadores de diferentes ramas los apliquen a diferentes tareas. El caso de PLN no es una excepción y por lo tanto se han aplicado estos modelos en diferentes tareas. Una técnica de ML que permite obtener un mejor rendimiento que solo usando un clasificador es *ensemble learning*, que consiste en emplear varios modelos en conjunto con el objetivo de obtener un mejor rendimiento predictivo que el que se podría obtener empleando cada modelo por separado. Está técnica se aplicó en Ankit and Saleena (2018) para el análisis de sentimientos.

Una de las mayores revoluciones en cuanto a técnicas de ML la han supuesto la evolución de los algoritmos basados en redes neuronales. Las redes neuronales, en especial las multicapa, permiten afrontar mejor el problema de clasificación con datos que presentan no linealidades como se suele dar en los problemas de clasificación de textos. Algunos estudios emplearon redes neuronales para clasificar textos desde los años 90 Ng et al. (1997); Ruiz and Srinivasan (1999). Tang et al. (2014) propuso un método basado en redes neuronales que emplea la técnica de *word embedding* para la clasificación de sentimientos de tweets. Por su parte Xu et al. (2016) propuso un método basado en redes LSTM (*Long Short-Term Memory*) para la clasificación de sentimientos de textos largos.

Otro enfoque empleado a la hora de diseñar métodos para clasificar textos en función del sentimiento, es el enfoque basado en el léxico. Este se basa en un léxico de sentimientos, es decir, una colección de palabras que expresan sentimiento positivo o negativo. Por norma general, las palabras de sentimiento positivo se emplean para expresar emociones como alegría, simpatía, etc., mientras que las palabras de sentimiento negativo se usan para expresar algunos estados como enfado, discordia, etc. Además de palabras que expresan sentimiento, existen ciertas expresiones denominadas idiomáticas, formadas por secuencias de palabras cuyo significado no es compositivo, es decir, el significado de la expresión no se deriva del de sus componentes, pero que juntas dan lugar,

en algunos casos, a la expresión de sentimientos que las palabras que la forman por sí solas no darían lugar. Las palabras y expresiones descritas forman el léxico que se empleará posteriormente para estimar el sentimiento de textos con el enfoque descrito. Existen diferentes técnicas para crear los léxicos de sentimientos. La forma más simple es la manual, que consiste simplemente en anotar de forma manual palabras positivas y negativas. Otros métodos más complejos emplean técnicas automáticas, por ejemplo Kim and Hovy (2004); Hu and Liu (2004) presentaron la estrategia principal del enfoque basado en "diccionarios" donde un pequeño conjunto de palabras que expresan sentimiento se recopila manualmente. Posteriormente, se buscan sinónimos y antónimos de estas palabras en corpus conocidos (*WordNet* o *thesaurus*). Las palabras encontradas se agregan a la lista inicial y luego comienza la siguiente iteración. El proceso iterativo se detiene cuando no se encuentran nuevas palabras. Una vez completado el proceso, se puede realizar una inspección manual para eliminar o corregir errores. Al final se tiene un gran diccionario de palabras que expresan sentimiento enriquecido con sinónimos y antónimos. Otras técnicas para crear el léxico de sentimientos se basan en corpus, como el método planteado por Hatzivassiloglou and McKeown (1997) donde se comienza con una lista de adjetivos que expresan sentimiento y se emplean junto con un conjunto de restricciones lingüísticas para identificar palabras de sentimiento adicionales y sus orientaciones. Esta idea se llama *sentiment consistency*.

Uno de los problemas que plantea el análisis de sentimientos en microblogs, como es el caso de Twitter, es el lenguaje especial que se emplea. En concreto se suelen emplear expresiones coloquiales, abreviaturas, emoticonos, etc. Por este motivo algunos investigadores crearon léxicos especiales como Nielsen (2011) que propuso el léxico AFINN, el cual contiene acrónimos y palabras de la *"jerga"* empleadas en Twitter. Otros autores como Thelwall et al. (2010) propusieron un léxico que además de palabras contenía una lista de emoticonos, negaciones y *boosting words*.

2.2. Análisis de polaridad reputacional.

Las impresas disponen de varios métodos para determinar como de satisfechos se encuentran los clientes con sus productos. Un método clásico son las encuestas, que ofrecen en algunos casos, la posibilidad de asignar una nota a un producto o, en otros casos, comentar sobre el mismo. La gran desventaja de las encuestas es que sueles resultar "pesadas" para los usuarios y estos responden en muchos casos de forma arbitraría.

La extensión de sistemas de microblogging como Twitter han permitido a los usuarios expresar su opinión sobre ciertas entidades, sin estar sometidos a la molestia de las encuestas, por lo que los usuarios pueden opinar cuando les apetezca. Este hecho unido al gran número de usuarios que poseen estos sistemas de microblogging, ha suscitado el rápido interés de las empresas por analizar su reputación en la red, y por ende, de los investigadores, que intentan desarrollar herramientas que sean capaces de procesar la gran cantidad de datos

disponibles en forma de texto, para extraer información relevante sobre las empresas interesadas.

Una de las primeras campañas para la gestión de la reputación online fue Replab 2012 Amigó et al. (2012). RepLab se centró en la reputación de empresas y solicitó a los participantes que anotaran diferentes tipos de información en los tweets que contenían los nombres de varias empresas. Se propusieron dos tareas: una tarea de "creación de perfiles", donde los tweets tenían que ser anotados por relevancia y polaridad reputacional, y una tarea de "monitoreo", donde los tweets tenían que agruparse temáticamente y los grupos tenían que ordenarse por prioridad. Los primeros estudios sobre polaridad reputacional que emplearon RepLab2012 se basaron en el análisis de sentimientos, dado en parte por la gran similitud con el problema de clasificación de sentimientos. Kaptein (2012) propuso un método para extraer las características de sentimientos de los tweets. Por su parte Yang et al. (2012) empleó *happiness feature* (basada en el nivel de "felididad" que expresa un tweet) para detectar la polaridad reputacional. Los sistemas que mejores resultaron obtuvieron fueron los creados por medio de herramientas comerciales. Villena-Román et al. (2012) empleó un herramienta de análisis de sentimientos, además de características lingüísticas, por su parte Karlgren et al. (2012) consideró la idea de una semántica referente a la satisfacción del cliente, que consiste en términos seleccionados manualmente además de una ampliación semiautomática a través de un modelo semántico.

Posteriormente se creó la campaña RepLab 2013 Amigó et al. (2013). En este caso se propusieron, en gran medida, técnicas de análisis de sentimientos y características textuales para la tarea de clasificación por polaridad reputacional. Además se emplearon otras técnicas de amplia difusión en problemas de PLN, como son: *bag-of-words* Filgueiras and Amir (2013); López et al. (2013), *clustering* Hangya and Farkas (2013) o TF-IDF (*Term frequency - Inverse document frequency*) Cossu et al. (2013). Más adelante, Peetz et al. (2016) propuso un enfoque de clasificación para estimar la polaridad de reputación de los tweets basado en *features* de tres dimensiones: la fuente del tweet, el contenido del tweet y la recepción del tweet, es decir, cómo se percibe el tweet. Sus experimentos empleando RepLab mostraron que las features que representaban cómo se percibía un tweet resultaban trascendentes y superaban a la mayoría de las otras *features* examinadas.

Otros enfoques se basan en la propagación de sentimientos. Giachanou et al. (2017) se plantea la hipótesis de que para determinar la polaridad reputacional se puede propagar el sentimiento de los textos a otros textos que tratan un mismo tema. Su método se basa en dos enfoques, por un lado, propagar directamente el sentimiento a textos similares y por otro, aumentar el léxico de polaridad. Los resultados obtenidos demostraron que construir léxicos de polaridad específicos del dominio mejora los resultados. Por su parte Giachanou et al. (2019) propuso dos enfoques para propagar señales de sentimiento para estimar la polaridad de reputación de los tweets. El primer enfoque se basa en el aumento de los léxicos de los sentimientos, mientras que el segundo se basa en la propagación directa de los sentimientos a los tweets que tratan un mismo tema. Además, presentó el método denominado *polar fact filter*, que puede diferenciar entre

polar facts y tweets de reputación neutral. Los resultados obtenidos demostraron que la anotación débilmente supervisada de la polaridad reputacional es factible y que las señales de sentimiento pueden propagarse para estimar efectivamente la polaridad reputacional de los tweets. Por último, se mostró que aprender los valores de PMI (*Pointwise Mutual Information*) de los datos de entrenamiento es el enfoque más efectivo para el análisis de polaridad de reputación.

2.3. Contextual word embeddings

Desde que existen las computadoras los investigadores han querido diseñar formas de insertar palabras en estas, de forma que puedan diferenciarlas, por lo que se crearon varios sistemas de codificación para tal efecto. En una computadora, la representación más simple de un fragmento de texto es una secuencia de caracteres (dependiendo de la codificación, un carácter puede ser un solo byte o varios). Una palabra se puede representar como una cadena (lista ordenada de caracteres), pero comparar si dos cadenas son idénticas es costoso.

Posteriormente se creó un sistema de representación de palabras, donde a cada tipo de palabra se le daba un valor entero único (y más o menos arbitrario) no negativo. Esto tenía las ventajas de que (1) cada tipo de palabra se almacenaba en la misma cantidad de memoria, y (2) las estructuras de datos basadas en matrices podían usarse para indexar otra información adicional del tipo de palabra. El vocabulario podría expandirse continuamente a medida que se encontraran nuevos tipos de palabras. La gran ventaja que aporta este sistema de representación empleando número enteros (representación discreta), es que el comprobar si dos enteros son idénticos tiene un coste computacional muy bajo. Por contra, la gran desventaja, es que a dos tipos de palabras con significados relacionados se les pueden asignar enteros distantes, y dos tipos de palabras "adyacentes" en la asignación pueden no tener nada que ver entre sí. El uso de enteros solo es una conveniencia. La representación de palabras de forma discreta no es conveniente si se pretende conocer la semejanza entre palabras. En múltiples tareas de PLN es necesario conocer dicha semejanza, por lo tanto, surgió la idea de realizar representaciones alternativas al método discreto.

Las máquinas por sí solas no tienen la capacidad de discernir la semejanza de palabras, pero los humanos, especialmente los que estudian la ciencia del lenguaje humano, conocen dicha información. Por lo tanto, serían capaces de diseñar estructuras de datos que la codifiquen explícitamente. Un ejemplo de tal esfuerzo es WordNet Fellbaum (1998), que es una base de datos léxica que almacena palabras y relaciones entre ellas, como la sinonimia e hiponimia. Además, captura los diferentes sentidos que puede tener una palabra. Otra forma de diferenciar palabras que ofrecen las teorías lingüísticas, es la referente a la estructura de las oraciones (sintaxis) en forma de categorías como "sustantivos" y "verbos".

Teniendo en cuenta las posibles categorizaciones de palabras que hemos comentado y muchas otras no comentadas, es posible representar una palabra como un vector en lugar de un simple entero. La dimensionalidad del vector

dependerá del propósito de uso. En PLN abundan los ejemplos de la asignación de dimensiones a vectores que representan tipos de palabras, o a secuencias de varias palabras. El término técnico utilizado para estas dimensiones es *feature* (característica). Las *features* pueden ser diseñadas por expertos, o pueden derivarse utilizando algoritmos automatizados.

En un corpus muy grande podemos recoger información de las formas en las que cierta palabra es usada, por ejemplo, el número de veces que aparece cerca de otro tipo de palabras. Cuando comenzamos a observar la distribución completa de contextos (palabras cercanas o secuencias de palabras) en un corpus donde se encuentra la palabra a estudiar, estamos teniendo una visión distribuida del significado de las palabras. De este enfoque surgió un método altamente exitoso para derivar automáticamente *features*, el método es conocido como *clustering*. Uno de los primeros algoritmos de este tipo fue Brown et al. (1992), el algoritmo de *clustering* organizó automáticamente las palabras en grupos en función de los contextos en los que aparecen en un corpus dado. Las palabras que solían aparecer en los mismos contextos "vecinos" se agruparon en un grupo. Estos grupos podrían fusionarse en grupos más grandes. La jerarquía resultante, aunque de ninguna manera idéntica a la estructura de datos elaborada por expertos en WordNet, fue sorprendentemente interpretable y útil. De este modo, los grupos de palabras adecuados se pueden construir por separado para textos de noticias, artículos biomédicos o tweets, por ejemplo.

A medida que los corpus crecieron, la escalabilidad se convirtió en un desafío, porque la cantidad de contextos observables también creció. Relacionado con la alta dimensionalidad debida en parte por el alto número de contextos distintos, surgió el problema de colocar el valor adecuado en cada dimensión del vector. Este problema se trató como un parámetro que se optimizará, junto con todos los demás parámetros, para ajustarse mejor a los patrones observados de las palabras en los datos. Dado que consideramos estos parámetros como valores continuos, y la noción de "ajustar los datos" puede tratar como una función objetiva continua y suave, la selección de los valores de los parámetros se realiza mediante algoritmos iterativos basados en el descenso del gradiente. Una de las colecciones de algoritmos más famosas al respecto es *word2vec* Mikolov et al. (2013). De este modo surgió una gran exploración de métodos para obtener vectores de palabras distribuidas, más conocidas como *distributional word vectors*. La técnica en si fue denominada como *word embedding*.

De todo lo expuesto hasta el momento se ha asumido que cada tipo de palabra se representaría usando un objeto de datos fijo (un entero o un vector). Esto resulta útil, pero hace algunas suposiciones sobre el lenguaje que no encajan con la realidad. La más importante es que las palabras tienen diferentes significados en diferentes contextos. Es difícil obtener un acuerdo general sobre cuántos significados deben asignarse a diferentes palabras, o en los límites entre un sentido y otro. Los sentidos de las palabras pueden ser fluidos. De hecho, en muchos programas de PLN basados en redes neuronales lo primero que hacen es pasar el vector que determina el tipo de cierta palabra, por una función que lo transforma dependiendo del contexto cercano, dando una nueva versión del vector de palabra, de este modo se tiene un vector adaptado a un contexto

particular.

Con la representación vectorial descrita anteriormente y debido a que las palabras significan cosas diferentes en contextos diferentes, se exige que las representaciones vectoriales capturaran todas las posibilidades. Pasar los vectores de palabras por funciones que capturen solo el significado en un contexto dado simplifica las cosas. Por las mismas razones por las que la colección de contextos en los que se encuentra un tipo de palabra proporciona pistas sobre su/s significado/s, el contexto de una palabra particular proporciona pistas sobre su significado específico.

Con la idea de adaptar la representación vectorial al contexto surgió ELMo (*embeddings from language models*) Peters et al. (2018). ELMo trajo un poderoso avance, los denominados *contextual word vectors*. Estos vectores permitieron por primera vez, diferenciar una misma palabra en función del contexto al que pertenece.

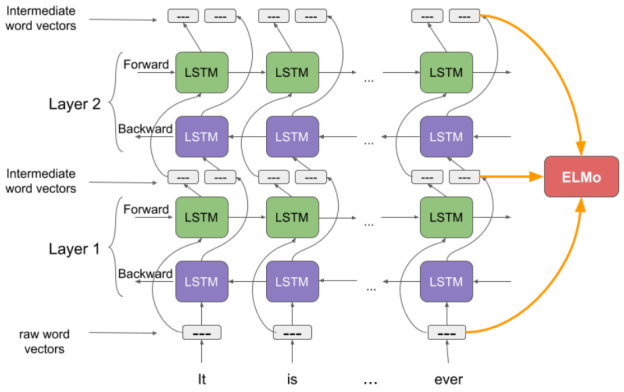

Figura 2.1: *Fuente: JOSHI (2020). Esquema de funcionamiento de ELMo*

Los vectores de palabras ELMo se calculan sobre un modelo de lenguaje bidireccional de dos capas (biLM), donde cada capa tiene 2 fases en el entrenamiento: avance y retroceso. Cada una de estas capas está formada por una concatenación de LSTM (biLSTM o LSTM bidireccionales) que son las que se encargan de aprender la información referente a los contextos izquierdo y derecho.

La arquitectura descrita utiliza una red neuronal convolucional a nivel de caracteres (CNN) para representar palabras de una cadena de texto en vectores de palabras sin procesar. Estos vectores de palabras sin procesar actúan como entradas a la primera capa de biLM. La fase de avance contiene información sobre una palabra determinada y el contexto (otras palabras) antes de esa palabra, es decir, se realiza un entrenamiento de LSTM desde el principio de la frase hasta la palabra en cuestión. La fase de retroceso contiene información sobre la palabra y el contexto posterior, es decir, ahora se entrena un LSTM desde el final de la frase hasta la palabra en cuestión. Este par de información

(fase de avance y fase de retroceso), forma los vectores de palabras intermedios. Estos vectores de palabras intermedios se alimentan a la siguiente capa de biLM. La representación final (ELMo) es la suma ponderada de los vectores de palabras sin procesar y los 2 vectores de palabras intermedias. La representación descrita se puede ver en la figura 2.1.

Como la entrada al biLM se calcula a partir de caracteres en lugar de palabras, ELMo captura la estructura interna de la palabra. Por lo tanto, ELMo podrá emplear pistas morfológicas para formar representaciones robustas para palabras que no están contenidas en el vocabulario de entrenamiento.

Los modelos del lenguaje basados en el funcionamiento descrito se denominan *contextual word embeddings*. Estos han supuesto una revolución en el PLN. Las precisiones obtenidas con este tipo de modelos en tareas típicas de PLN como traducción de textos, clasificación, etc., han permitido destronar a otras técnicas más clásicas por esta nueva metodología.

Desde el surgimiento de ELMo se han desarrollado varios modelos *contextual word embeddings*, cada uno con sus características peculiares. Como diferencia importante entre unos modelos y otros cabe destacar el número de capas ocultas y por lo tanto el número de parámetros que componen las redes neuronales. Desde los primeros modelos a los últimos se han pasado de miles a varios millones de parámetros. Otra diferencia importante hace referencia a la estructura empleada, que puede estar formada por LSTMs bidireccionales (como el caso de ELMo) o por trasnformadores. En el próximo punto se hablará más detalladamente de la estructura basada en trasnformadores, que ha supuesto una importante mejora en los modelos basados en *contextual word embeddings*.

2.4. Aplicación de transformers en contextual word embeddings

Las redes neuronales recurrentes (RNN), como por ejemplo las LSTM, se han establecido firmemente como enfoques de vanguardia en problemas de modelado que incluyan codificación y decodificación, tales como modelado del lenguaje (como se ha visto en el punto anterior). Los modelos basados en redes neuronales recurrentes suelen factorizar el cálculo a lo largo de las posiciones de los caracteres de las secuencias de entrada y salida. Al alinear las posiciones con los *steps* (pasos de entrenamiento) en el tiempo de cálculo, generan una secuencia de estados ocultos ht, en función del estado oculto anterior $(ht - 1)$ y la entrada para la posición t. Esta naturaleza secuencial impide la paralelización de cómputo a la hora de realizar el entrenamiento del modelo, lo que se vuelve crítico en longitudes de secuencia largas, ya que las restricciones de memoria limitan el procesamiento por lotes. Se han intentado desarrollar técnicas que mejoran la eficiencia computacional, basados especialmente en el uso de mecanismos *attention* (mecanismos de atención), pero la restricción fundamental del cómputo secuencial sigue siendo la paralelización, dado que los mecanismos *attention* se continúan empleando con redes recurrentes. Por

otro lado, este tipo de modelos basados en redes neuronales recurrentes, tienen el inconveniente de que el procesamiento secuencial limita que las neuronas solo puedan atender a palabras cercanas respecto de cada palabra.

Como alternativa a los modelos de *contextual word embeddings* basados en redes recurrentes se planteó la idea de *transformers* Vaswani et al. (2017). Las arquitecturas basadas en *transformers* evitan la recurrencia y los problemas inherentes a esta que hemos comentado. Esto es debido a que su estructura se basa completamente en un mecanismo *attention* (no emplean redes recurrentes), que le permite calcular una serie de representaciones del espacio vectorial de los caracteres de entrada en función del contextos en una sola pasada (el comportamiento secuencial tenía que procesar palabra por palabra). La falta de recurrencia y por tanto de comportamiento secuencial, les permite ser "omnidireccionales", es decir, todos los puntos de la red se fijan en todos los demás simultáneamente, lo que quiere decir que se puede contextualizar directamente a partir de todos los contexto de una palabra a izquierda y a derecha. Este hecho hace que los *transformers* permitan el modelado más efectivo de las dependencias a largo plazo entre las palabras en una secuencia dada y el entrenamiento más eficiente del modelo en general, al eliminar la dependencia secuencial de las palabras anteriores y permitiendo mayor paralelización en el cómputo.

Los *transformers* son modelos de arquitectura basados en la idea "codificador-decodificador" que utiliza mecanismos *attention* para enviar una imagen más completa de la secuencia dada al decodificador, en vez de hacerlo secuencialmente como en el caso de los modelos basados en redes recurrentes. Las capas *attention* que componen el *transformer*, codifican cada palabra de una frase en función del resto de la secuencia de palabras, permitiendo así introducir el contexto en la representación matemática del texto, razón por la cual a los modelos basados en *transformers* se les denomina también *Contextual Embeddings*. La arquitectura de *transformers* incluye otras innovaciones, como los *positional embeddings*, que permiten al algoritmo conocer la posición relativa de cada palabra del texto y poder recuperar de esta forma información de posición.

Los modelos basados en *transformers* suelen funcionar en dos etapas pre-entrenado y *fine-tuning*. En la fase de pre-entrenado, el modelo aprende cómo se estructura el lenguaje de forma general, además de conseguir un conocimiento genérico del significado de las palabras. En la fase de *fine-tuning*, habiendo pre-entrenado antes, se le añaden ciertas capas a la arquitectura de la red para adaptar los modelos a tareas concretas, realizando el entrenamiento de la red resultante. De este modo el modelo aprende sobre la tarea en cuestión.

En la actualidad los modelos del lenguaje basados en *transformers* están ofreciendo mejores resultados que los basados en redes recurrentes por las razones ya descritas Estos modelos avanzan hacia arquitecturas progresivamente más grandes, que son entrenadas con la mayor cantidad de textos disponible. Un problema colateral de esto es que los modelos más grandes son también más lentos, aun permitiendo cierta paralelización en el procesado como se ha comentado. Esto ha permitido que surja una nueva línea de investigación, que es

la relativa a intentar hacer modelos basados en la arquitectura de *transformers* que sean más ligeros sin sacrificar efectividad, a la vez que permitan codificar secuencias de texto más largas, pues una de las principales limitaciones es el exceso de consumo de recursos de computación y el tamaño limitado de las posibles secuencias de entrada.

En el presente trabajo se pretende aplicar uno de los últimos modelos *contextual word embeddings,* que además emplea el concepto de *transformers*, en concreto se pretende aplicar BERT (*Bidirectional Encoder Representations from Transformers*) con el que se han obtenido algunos de los mejores resultados hasta el momento en varios problemas de clasificación típicos del procesamiento del lenguaje natural, como es la clasificación de sentimientos.

2.5. BERT

BERT Devlin et al. (2018) es un modelo de representación de lenguaje que significa *Bidirectional Encoder Representations from Transformers*. A diferencia de otros modelos de representación del lenguaje (como ELMo), BERT emplea *transformers* lo que le permite prevenir las representaciones bidireccionales, al condicionar simultáneamente el contexto izquierdo y derecho en todas las capas (como se explicó en el punto 2.4). El modelo BERT pre-entrenado se puede ajustar con solo una capa de salida adicional para crear modelos de última generación para una amplia gama de tareas, como la respuesta a preguntas y la inferencia del lenguaje, sin modificaciones sustanciales de la arquitectura específica de la tarea.

Existen dos estrategias existentes para aplicar representaciones de lenguaje pre-entrenadas a tareas posteriores: *feature-based* (basadas en características) y *fine-tuning* (ajuste fino). El enfoque *feature-based*, empleado por ELMo utiliza arquitecturas específicas de tareas que incluyen las representaciones pre-entrenadas como *features* adicionales. El enfoque *fine-tuning*, aplicado por *Generative Pre-trained Transformer* (OpenAI GPT) Radford (2018), introduce parámetros mínimos específicos de la tarea, y se entrena en las tareas posteriores simplemente ajustando todos los parámetros previamente entrenados. Los dos enfoques comparten la misma función objetivo durante el pre-entrenamiento, donde usan modelos de lenguaje unidireccionales para aprender representaciones generales del lenguaje.

La principal limitación de los modelos de lenguaje estándar (como ELMo y OpenAI GPT) es que son unidireccionales, y esto limita la elección de arquitecturas que se pueden usar durante el pre-entrenamiento. Esto quiere decir que los modelos observan la secuencia de entrada de izquierda a derecha o de derecha a izquierda, independientemente de que empleen redes LSTM (ELMo) o *transformers* unidireccionales (OpenAI GPT). Estás restricciones son subóptimas para las tareas a nivel de oración y podrían ser muy dañinas cuando se aplican enfoques basados en *fine-tuning* a tareas a nivel de token, como la respuesta a preguntas, donde es crucial incorporar el contexto en ambas direcciones.

BERT alivia la restricción de unidireccionalidad, mencionada anteriormente, mediante el uso de un objetivo de pre-entrenamiento de "modelo de lenguaje enmascarado" (MLM). El modelo de lenguaje enmascarado enmascara al azar algunos de los tokens de la entrada (en BERT se enmascara el 15 % de las palabras) con un token [MASK]. El objetivo es intentar predecir el valor original de las palabras enmascaradas, en función del contexto proporcionado por las otras palabras no enmascaradas en la secuencia de entrada dada. A diferencia del pre-entrenamiento de modelos del lenguaje bidireccionales (como ELMo), MLM permite entrenar un *transformer* profundo que fusiona los contextos izquierdo y derecho, lo que se conoce como *bidirectional transformer*.

Una particularidad de BERT es la representación de entrada/salida que realiza. Por un lado, la representación de entrada es capaz de representar inequívocamente una sola oración y un par de oraciones en una secuencia de tokens. Una "secuencia" se refiere a la secuencia de token de entrada a BERT, que puede ser una sola oración o dos oraciones juntas. Para ayudar al modelo a distinguir entre las dos oraciones en el entrenamiento, la entrada se procesa de la siguiente manera antes de ingresar al modelo:

- Se inserta un token [CLS] al comienzo de la primera oración y se inserta un token [SEP] al final de cada oración.
- Cada token se representa mediante tokenización WordPiece (*token embedding*), que consta de un vocabulario de 30000 tokens. Por ejemplo, la frase "*multilingual model*" se representaría como "*multi ##lingual model*".
- Se agrega una incrustación de oración (*sentence embedding*) que indica si un token pertenece a la oración *A* o la oración *B*.
- Se agrega una incrustación posicional a cada token (*positional embedding*) para indicar su posición en la secuencia.

La representación de entrada se construye sumando las tres representaciones de token, sentencia y posición comentadas. En la figura 2.2 se puede ver esta representación.

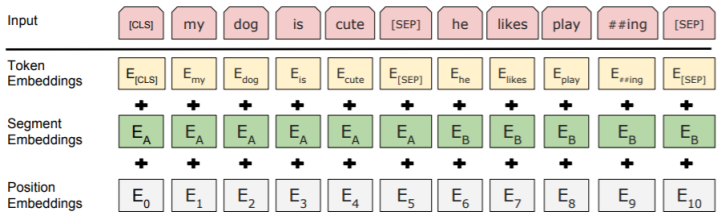

Figura 2.2: *Fuente: Devlin et al. (2018). Representación de la entrada de BERT. Las incrustaciones de entrada son la suma de las incrustaciones de token, las incrustaciones de oración y las incrustaciones de posición.*

BERT consta de dos tareas, como ya se ha mencionado, para poder aplicarse: pre-entrenamiento y *fine-tuning* (ajuste fino).

Durante el pre-entrenamiento el modelo se entrena con datos no etiquetados. Como se ha descrito, BERT no emplea la técnica de bidireccionalidad (como usaba ELMo) dado que no está basado en redes recurrentes, si no en *transformers*. Por el contrario, emplea dos tareas no supervisadas para completar la representación bidireccional profunda. Primero se aplica "LM (*lenguage model*) enmascarado" (*masked LM*). Esta consiste en enmascarar un porcentaje de los tokens de entrada al azar, y luego predecir esos tokens enmascarados. Este procedimiento denominado "*masked LM*" (MLM) también se conoce como tarea *Cloze* en la literatura. En este caso, los vectores ocultos finales, correspondientes a los tokens de máscara, se alimentan a una capa *softmax* de salida sobre el vocabulario, como en un LM estándar. La segunda tarea es la predicción de la siguiente oración (NSP). Muchas tareas de PLN como la "respuesta a preguntas" (QA) y la inferencia del lenguaje natural (NLI) se basan en la comprensión de la relación entre dos oraciones, que no se capta directamente mediante el modelado del lenguaje. Para entrenar un modelo que entienda las relaciones de las oraciones, pre-entrenamos para una tarea de predicción de la siguiente oración binarizada que puede generarse trivialmente a partir de cualquier corpus monolingüe. El procedimiento de pre-entrenamiento sigue en gran medida la literatura existente sobre pre-entrenamiento de modelos de lenguaje existentes. Aunque es posible adaptar BERT a nuestras necesidades realizando nosotros mismos el pre-entrenamiento, los desarrolladores dispusieron un modelo previamente pre-entrenado que se adapta bien a la mayoría de problemas de PLN. Como corpus de pre-entrenamiento emplearon BooksCorpus (800 millones de palabras) y Wikipedia en inglés (2.500 millones de palabras). Resulto fundamenta emplear un corpus a nivel de documento en lugar de un corpus aleatorio a nivel de oración, para poder extraer secuencias contiguas largas.

El proceso de *fine-tuning* consiste en aplicar BERT a una tarea específica. Para cada tarea, simplemente conectamos las entradas y salidas específicas al modelo BERT y ajustamos todos los parámetros (*fine-tune*) de "extremo a extremo" del modelo. En la salida las representaciones de token se introducen en una capa de salida para tareas a "nivel de token", como el etiquetado de oraciones o la respuesta a preguntas, y la representación [CLS] se alimenta en una capa de salida para problemas de clasificación, como la vinculación o el análisis de sentimientos.

En la figura 2.3 se puede ver un esquema que recoge la parte de pre-entrenado y ajuste fino para una tarea de preguntas y respuestas.

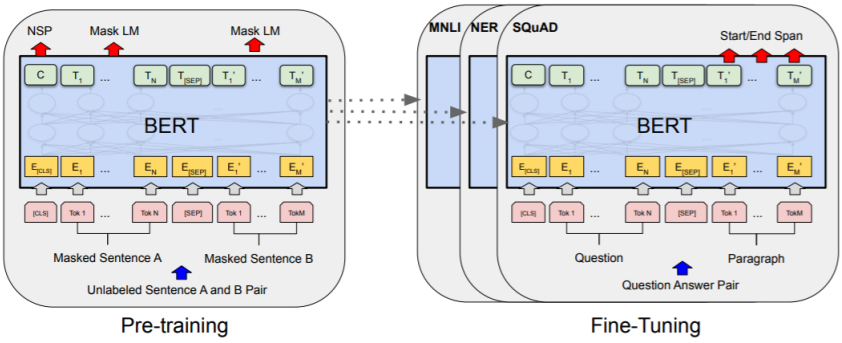

Figura 2.3: *Fuente: Devlin et al. (2018). Procedimientos de pre-entrenamiento y ajuste fino para BERT. Además de las capas de salida, se utilizan las mismas arquitecturas tanto en el pre-entrenamiento como en el fine-tuning. Los mismos parámetros de modelo previamente entrenados se utilizan para inicializar modelos para diferentes tareas posteriores. Durante el ajuste fino, todos los parámetros están ajustados. [CLS] es un símbolo especial agregado delante de cada ejemplo de entrada, y [SEP] es un token separador especial (por ejemplo, preguntas/respuestas de separación).*

En la actualidad existen varias versiones de BERT disponibles (aunque cada poco tiempo salen varias nuevas). Las variaciones de cada versión dependen del número de capas, denotado por L y el tamaño oculto (*hidden size*) H. Las dos versiones más destacables son BERTBASE (L = 12 , H = 768, Parámetros totales = 110M) y BERTLARGE (L = 24, H = 1024, Parámetros totales = 340M), aunque existen otras disponibles BER (2019). Otra de las diferencias es los idiomas soportados por cada versión. En el momento de realizar este documento existen versiones para inglés, chino y una versión multi-lenguaje que soporta 102 lenguas distintas.

Dada la versatilidad de BERT, su capacidad para soportar varios idiomas y los resultados que ha ofrecido en varios problemas de PLN clásicos, han hecho que nos decantemos por emplear este algoritmo de *contextual word embeddings*.

Capítulo 3

Diseño experimental

Este capítulo describe el *dataset* empleado para hacer los experimentos, así como la metodología utilizada para evaluar el sistema propuesto y las distintas variantes empleadas en los experimentos.

3.1. Dataset RepLab 2013

Para evaluar el método propuesto, empleamos la colección RepLab 2013 Amigó et al. (2013) que contiene datos de Twitter en inglés y español. RepLab 2013 se diseñó para la tarea de monitorear la reputación de las entidades (empresas, organizaciones, celebridades, etc.) en Twitter. La tarea de monitoreo para analistas consiste en buscar en el flujo de tweets posibles menciones a la entidad, filtrar aquellos que sí se refieren a la entidad, detectar temas (es decir, agrupar tweets por tema) y clasificarlos en función del grado en que señalan alertas de reputación (es decir, problemas que pueden tener un impacto sustancial en la reputación de la entidad). Hasta el momento RepLab 2013 es una de las colecciones más grande de Twitter para el monitoreo de la reputación.

El *dataset* RepLab 2013 consta de un conjunto de datos etiquetados con la polaridad reputacional y otro conjunto de datos no etiquetado. Para los experimentos realizados solo se han empleado los datos etiquetados, que están divididos en un conjunto de dato de entrenamiento (34095 tweets), recolectados tres meses antes en relación con el conjunto de datos de prueba (75470 tweets). El resto de los tweets rastreados (1,038,060 tweets) representan el conjunto de datos de fondo y se refieren a los tweets publicados entre los conjuntos de entrenamiento y prueba. Cada conjunto de datos está dividido en 61 entidades diferentes que pertenecen a cuatro dominios distintos: automotriz, banca, universidades y música.

En las figuras de 3.1 a 3.3 se pueden ver diferentes representaciones de los *datasets* de *train* y *test* en función de la lengua el dominio, la entidad y la polaridad.

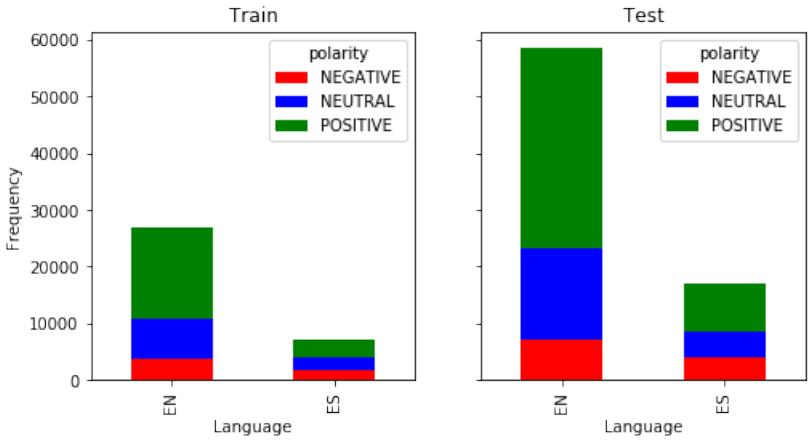

Figura 3.1: *Tamaño del dataset train y test en función de la lengua y la polaridad*

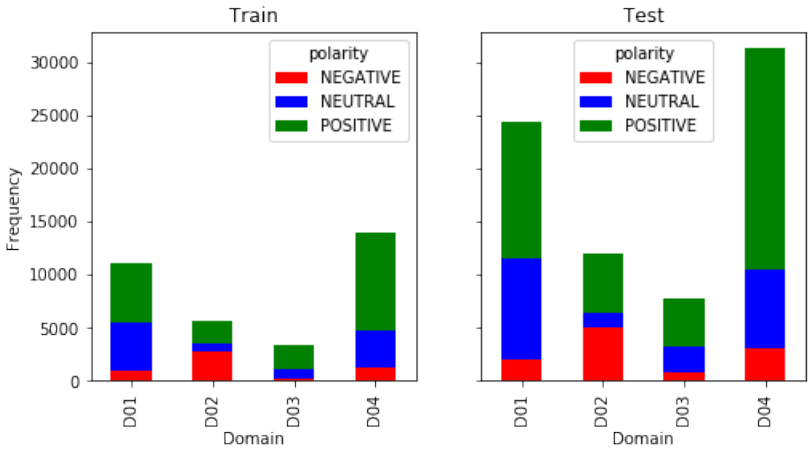

Figura 3.2: *Tamaño del dataset train y test en función del dominio y la polaridad*

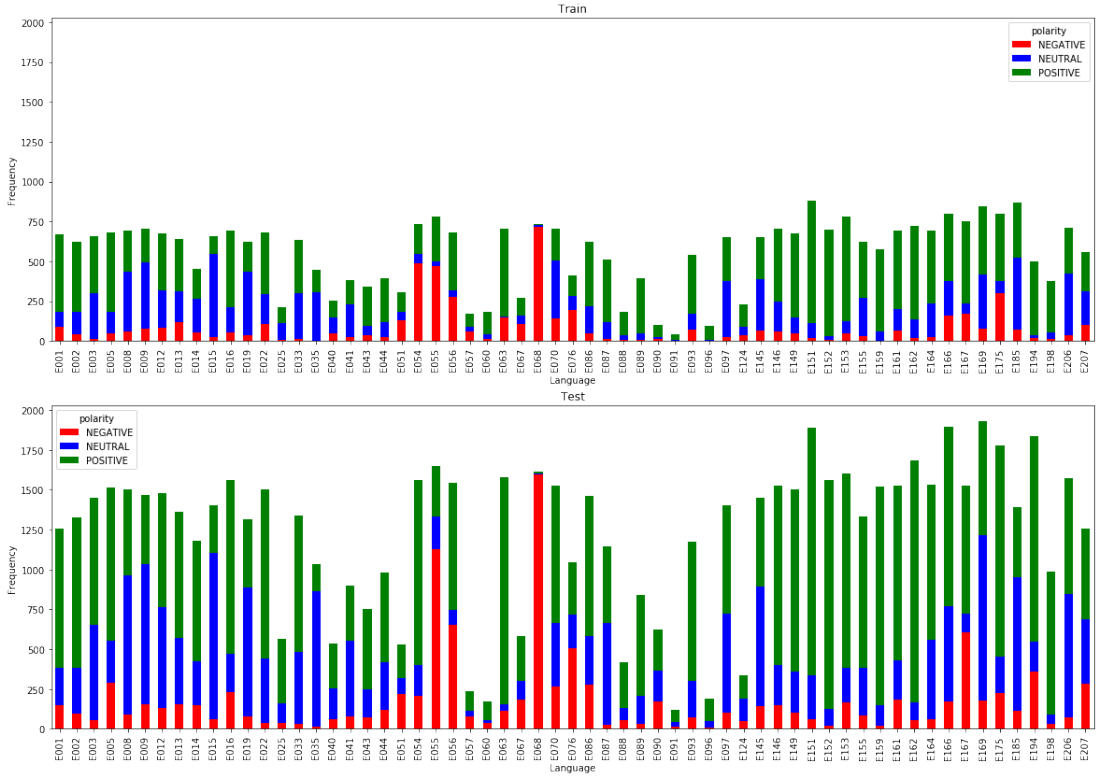

Figura 3.3: *Tamaño del dataset train y test en función de la entidad y la polaridad*

3.2. Método propuesto

El método propuesto pretende clasificar tweets en función de su polaridad reputacional empleando el modelo del lenguaje BERT. El problema se plantea como un clasificador de textos, es decir, el modelo se entrena sobre un conjunto de tweets de entrenamiento etiquetados de forma manual en base a la polaridad, de este modo el modelo aprende las relaciones que permiten segmentar los tweets por polaridad. Posteriormente el modelo se emplea para intenta predecir la polaridad de un conjunto de tweets de prueba.

Para poder emplear el modelo BERT como un clasificador de texto se debe presentar a la entrada del modelo un texto en la forma descrita en el punto 2.5, con un token [CLS] al inicio y [SEP] al final. Por último, se debe conectar una capa completamente conectada (perceptrón multicapa) sobre el estado oculto final, correspondiente con al token de entrada [CLS] del modelo BERT. El token de clasificación especial [CLS] codifica la información de toda la secuencia de texto de entrada, lo que al conectarla a un pequeño perceptrón multicapa,

que consta de capas completamente conectadas (densas), permite generar la distribución de todos los valores de etiqueta discretos. En la figura 3.4 se puede ver la arquitectura descrita.

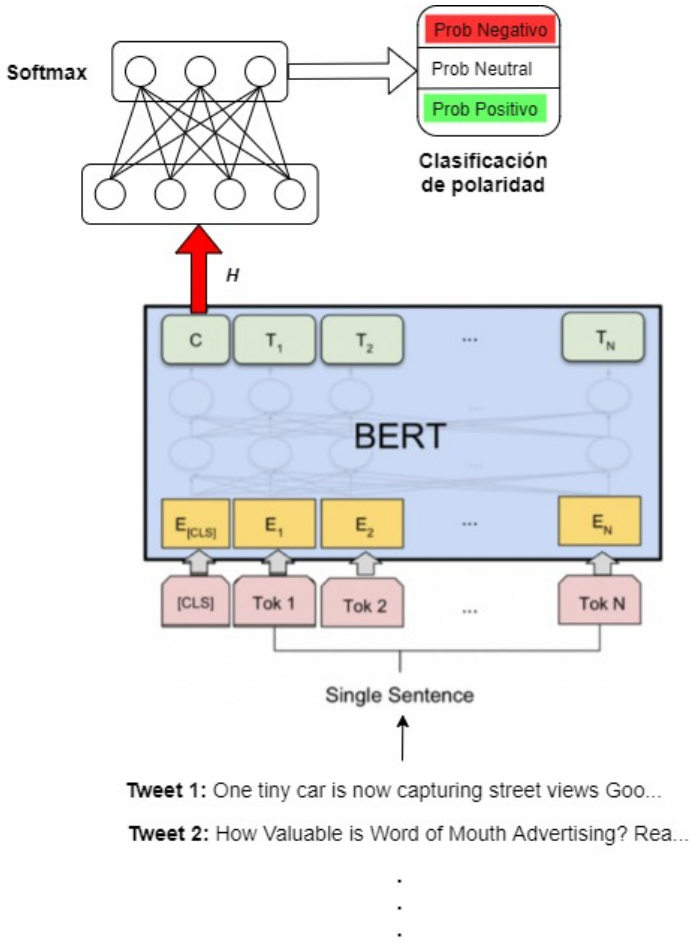

Figura 3.4: *Arquitectura propuesta para la clasificación de tweets en función de la polaridad reputacional*

En la arquitectura mostrada en la figura 3.4, N representa la longitud de la secuencia máxima, que es un valor que se establece de antemano. Si el número de tokens de un tweet supera este umbral, este se trunca para poder adaptarlo a la longitud establecida. Por su parte H representa el tamaño oculto (*hidden size*) que depende de la versión del modelo BERT (descrito en el punto 2.4). En la salida del modelo se ha empleado una capa *softmax* de tamaño 3 (una por cada tipo de polaridad) para poder obtener la probabilidad de cada una de las 3 clases y poder comprender mejor los experimentos. Esta capa se conecta a la

salida del modelo BERT empleando una capa *dropout* que elimina el 20 % de las conexiones. Esta se emplea para reducir el sobreajuste del modelo.

Las características empleadas en el proceso de entrenamiento del modelo se recogen a continuación:

- Optimizador: *Adam* con *learning rate* $= 3 * 10^{-5}$
- *Epochs*: 2
- *Loss*: *sparse categorical crossentropy*.

Otro parámetro importante es *maximum length of a sequence* que indica la longitud máxima de la secuencia de tokens de entrada después del proceso de tokenización. La longitud máxima en cualquier modelo BERT es de 512, pero se decide usar 128 dado que ningún tweet supera dicho umbral tras ser tokenizado, además de permitir que el entrenamiento tarde menos.

El proceso de uso del método propuesto consta de dos etapas, que se aplicarán en cada uno de los experimentos realizados. Primero se realiza el entrenamiento o *fine-tuning* del modelo empleando el conjunto de datos de entrenamiento. Una vez se tiene el modelo pre-entrenado se procede a predecir los datos de test. Los conjuntos de datos de entrenamiento y test se describen en el apartado 3.1.

En el proceso de predicción el método calcula, para cada tweet, tres probabilidades referentes a cada una de las posibles polaridades. La polaridad que se asigna en cada caso, corresponderá con la mayor probabilidad calculada.

3.3. Métricas de evaluación

Para evaluar las diferentes combinaciones del método propuesto se emplean varias métricas que miden la precisión con la que el modelo es capaz de clasificar la polaridad reputacional. En concreto se emplean *precision*, *reca*ll y *F1-score* *Zhang et al. (2015)* que serán obtenidos por clase y por cómputo global, donde estos valores se darán en la versión "macro" (calcular la métrica por separado para cada clase y luego tomar el valor promedio).

3.4. Variantes del sistema

Se han realizado varias pruebas variando el idioma de los tweets, el dominio y la versión del modelo BERT empleada. En el cuadro 3.1. se resumen todas las pruebas realizadas.

En todos los casos se ha empleado como conjunto de datos de entrenamiento y test los *datasets* ofrecidos para tal objetivo por RepLab 2013, en los cuales se han filtrado el idioma y el dominio en función del experimento realizado. Es decir, el modelo se entrena solo con los tweets correspondientes al conjunto de datos de entrenamiento que cumplen el idioma y dominio indicado, y se

testea solo con los tweets del conjunto de datos de test que cumplen las mismas condiciones mencionadas.

Idioma	Dominio	Modelo
Inglés	Todos	*bert-base-cased*
Español	Todos	*bert-base-multilingual-cased*
Inglés y español	Todos	*bert-base-multilingual-cased*
Inglés y español	D01	*bert-base-multilingual-cased*
Inglés y español	D02	*bert-base-multilingual-cased*
Inglés y español	D03	*bert-base-multilingual-cased*
Inglés y español	D04	*bert-base-multilingual-cased*

Cuadro 3.1: *Tabla resumen de pruebas realizadas*

En todos los casos se ha empleado la versión "cased" del modelo, esta versión soporta las mayúsculas y en la actualidad es la versión más recomendada en comparación con la versión "*uncased*" BER (2019). Las versiones del modelo empleadas tienen como características *12-layer*, *768-hidden*, *12-heads*, *110M parámetros*, han sido entrenados para textos en inglés para la versión *bert-base-cased* y en 104 idiomas distintos (entre los que está en inglés y el español) para la versión *bert-base-multilingual-cased*.

Capítulo 4

Resultados experimentales

Este capítulo mostrará los resultados obtenidos en cada uno de los experimentos realizados. En cada uno de los subcapítulos se describirá el experimento realizado y se mostrarán los resultados de forma cualitativa.

4.1. Tweets en inglés, todos los dominios, modelo *bert-base-cased*

Características del experimento:

- Lengua de los tweets: inglés
- Dominio: todos
- Modelo: *bert-base-cased*

En la tabla 4.1 se recoge el tamaño de las muestras y en la tabla 4.2 los resultados obtenidos.

Conjunto de datos	**Nº de tweets**
Train	26869
Train Negativos	4446
Train Neutrales	8228
Train Positivos	16882
Test	58568

Cuadro 4.1: *Tamaño de las muestras empleadas para el experimento: Tweets en inglés, todos los dominios, modelo bert-base-cased*

Polaridad	**Precisión**	**Recall**	**F1-score**
Negativa	0,490	0,6	0,54
Neutral	0,64	0,42	0,5
Positiva	0,76	0,85	0,81
Macro avg	0,63	0,62	0,62

Cuadro 4.2: *Resultados obtenidos para el experimento: Tweets en inglés, todos los dominios, modelo bert-base-cased*

4.2. Tweets en español, todos los dominios, modelo *bert-base-multilingual-cased*

Características del experimento:

- Lengua de los tweets: español
- Dominio: todos
- Modelo: *bert-base-multilingual-cased*

En la tabla 4.3 se recoge el tamaño de las muestras y en la tabla 4.4 los resultados obtenidos.

Conjunto de datos	**Nº de tweets**
Train	7227
Train Negativos	1996
Train Neutrales	2433
Train Positivos	3521
Test	16902

Cuadro 4.3: *Tamaño de las muestras empleadas para el experimento: Tweets en español, todos los dominios, modelo bert-base-multilingual-cased*

Polaridad	**Precisión**	**Recall**	**F1-score**
Negativa	0,790	0,54	0,64
Neutral	0,49	0,47	0,48
Positiva	0,67	0,78	0,72
Macro avg	0,65	0,6	0,62

Cuadro 4.4: *Resultados obtenidos para el experimento: Tweets en español, todos los dominios, modelo bert-base-multilingual-cased*

4.3. Tweets en inglés y español, todos los dominios, modelo *bert-base-multilingual-cased*

Características del experimento:

- Lengua de los tweets: inglés y español
- Dominio: todos
- Modelo: *bert-base-multilingual-cased*

En la tabla 4.5 se recoge el tamaño de las muestras y en la tabla 4.6 los resultados obtenidos.

Conjunto de datos	**Nº de tweets**
Train	34096
Train Negativos	6441
Train Neutrales	10660
Train Positivos	20405
Test	75470

Cuadro 4.5: *Tamaño de las muestras empleadas para el experimento: Tweets en inglés y español, todos los dominios, modelo bert-base-multilingual-cased*

Polaridad	**Precisión**	**Recall**	**F1-score**
Negativa	0,77	0,4	0,53
Neutral	0,61	0,47	0,53
Positiva	0,72	0,88	0,79
Macro avg	0,7	0,59	0,62

Cuadro 4.6: *Resultados obtenidos para el experimento: Tweets en inglés y español, todos los dominios, modelo bert-base-multilingual-cased*

4.4. Tweets en inglés y español, dominio D01, modelo *bert-base-multilingual-cased*

Características del experimento:

- Lengua de los tweets: inglés y español
- Dominio: D01
- Modelo: *bert-base-multilingual-cased*

En la tabla 4.7 se recoge el tamaño de las muestras y en la tabla 4.8 los resultados obtenidos.

Conjunto de datos	**Nº de tweets**
Train	11124
Train Negativos	1345
Train Neutrales	4873
Train Positivos	6019
Test	24415

Cuadro 4.7: *Tamaño de las muestras empleadas para el experimento: Tweets en inglés y español, dominio D01, modelo bert-base-multilingual-cased*

Polaridad	**Precisión**	**Recall**	**F1-score**
Negativa	0,64	0,04	0,07
Neutral	0,63	0,7	0,66
Positiva	0,71	0,77	0,74
Macro avg	0,66	0,5	0,49

Cuadro 4.8: *Resultados obtenidos para el experimento: Tweets en inglés y español, dominio D01, modelo bert-base-multilingual-cased*

4.5. Tweets en inglés y español, dominio D02, modelo *bert-base-multilingual-cased*

Características del experimento:

- Lengua de los tweets: inglés y español
- Dominio: D02
- Modelo: *bert-base-multilingual-cased*

En la tabla 4.9 se recoge el tamaño de las muestras y en la tabla 4.10 los resultados obtenidos.

Conjunto de datos	**Nº de tweets**
Train	5683
Train Negativos	2.949
Train Neutrales	942
Train Positivos	2361
Test	12053

Cuadro 4.9: *Tamaño de las muestras empleadas para el experimento: Tweets en inglés y español, dominio D02, modelo bert-base-multilingual-cased*

Polaridad	Precisión	Recall	F1-score
Negativa	0,68	0,93	0,79
Neutral	0,23	0,4	0,29
Positiva	0,94	0,46	0,62
Macro avg	0,62	0,6	0,57

Cuadro 4.10: *Resultados obtenidos para el experimento: Tweets en inglés y español, dominio D02, modelo bert-base-multilingual-cased*

4.6. Tweets en inglés y español, dominio D03, modelo *bert-base-multilingual-cased*

Características del experimento:

- Lengua de los tweets: inglés y español
- Dominio: D03
- Modelo: *bert-base-multilingual-cased*

En la tabla 4.11 se recoge el tamaño de las muestras y en la tabla 4.12 los resultados obtenidos.

Conjunto de datos	Nº de tweets
Train	3373
Train Negativos	341
Train Neutrales	994
Train Positivos	2376
Test	7715

Cuadro 4.11: *Tamaño de las muestras empleadas para el experimento: Tweets en inglés y español, dominio D03, modelo bert-base-multilingual-cased*

Polaridad	Precisión	Recall	F1-score
Negativa	0,330	0,06	0,1
Neutral	0,45	0,77	0,57
Positiva	0,8	0,6	0,69
Macro avg	0,52	0,48	0,45

Cuadro 4.12: *Resultados obtenidos para el experimento: Tweets en inglés y español, dominio D03, modelo bert-base-multilingual-cased*

4.7. Tweets en inglés y español, dominio D04, modelo *bert-base-multilingual-cased*

Características del experimento:

- Lengua de los tweets: inglés y español
- Dominio: D04
- Modelo: *bert-base-multilingual-cased*

En la tabla 4.1 se recoge el tamaño de las muestras.

Conjunto de datos	**Nº de tweets**
Train	13916
Train Negativos	1807
Train Neutrales	3851
Train Positivos	9650
Test	31287

Cuadro 4.13: *Tamaño de las muestras empleadas para el experimento: Tweets en inglés y español, dominio D04, modelo bert-base-multilingual-cased*

Polaridad	**Precisión**	**Recall**	**F1-score**
Negativa	0,4	0,2	0,26
Neutral	0,57	0,49	0,53
Positiva	0,78	0,88	0,83
Macro avg	0,58	0,52	0,54

Cuadro 4.14: *Resultados obtenidos para el experimento: Tweets en inglés y español, dominio D04, modelo bert-base-multilingual-cased*

Capítulo 5

Discusión

Este capítulo analiza y discute en profundidad los resultados obtenidos en la evaluación presentada en el capítulo anterior. Además, se compararán los resultados con estudios anteriores.

5.1. Análisis de resultados

En la tabla 5.1 se pueden ver el resumen de resultados obtenidos seleccionando como métrica *F1-score.*

Idioma	Dominio	Modelo	**F1-score**
Inglés	Todos	*bert-base-cased*	0,62
Español	Todos	*bert-base-multilingual-cased*	0,62
Inglés y español	Todos	*bert-base-multilingual-cased*	0.62
Inglés y español	D01	*bert-base-multilingual-cased*	0,49
Inglés y español	D02	*bert-base-multilingual-cased*	0,57
Inglés y español	D03	*bert-base-multilingual-cased*	0,45
Inglés y español	D04	*bert-base-multilingual-cased*	0,54

Cuadro 5.1: Tabla resumen de resultados

Como se puede apreciar, el valor *F1-score* en los tres primeros experimentos en los que no se filtra por dominio es la misma, mientras que en los casos en los que se filtra por dominio este valor baja. Observando los resultados de *F1-score* obtenidos por polaridad para cada uno de los experimentos, se puede apreciar como por norma general, los valores son mejores cuanto mayor es la proporción de datos, es decir, si la proporción de datos clasificados con polaridad positiva es mayor que el resto, tiende a clasificar mejor esta polaridad. Esto tiene sentido dado que el modelo es capaz de extraer más información de la clase mayoritaria. En la figura 5.1 se observan los resultados obtenidos de *F1-score* para cada uno de los experimentos en los que se ha separado por dominio. Si se compara con

la figura 4.2, se puede observar cómo existe bastante relación entre el valor de *F1-score* obtenido y el tamaño de cada muestra.

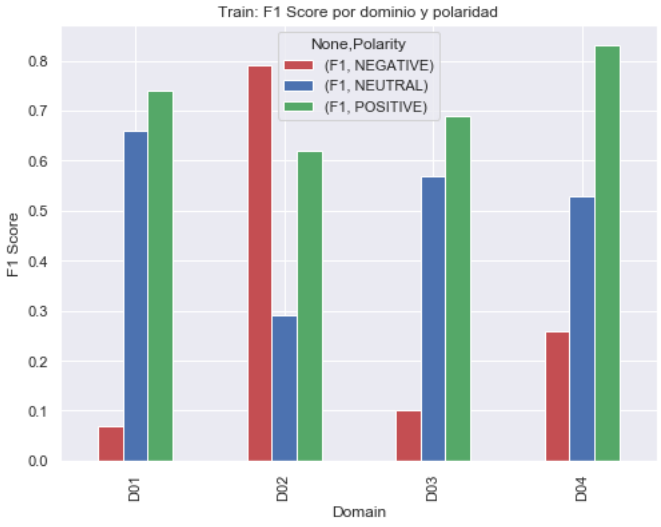

Figura 5.1: *F1-score obtenido por dominio y polaridad*

5.1.1. Estudio de correlación entre el número de tweets por dominio/polaridad o dominio/entidad/polaridad y *F1-score* obtenida

Para estudiar que efecto tiene el tamaño de las muestras sobre los resultados obtenidos calcularemos la correlación que existe entre el número de tweets y el valor de *F1-score* por cada dominio/polaridad y el *F1-score* por cada dominio/entidad/polaridad. Se han empleado los datos del experimento para los tweets en inglés y español y todos los dominios.

En la figura 5.2 se muestra los valores de *F1-score* obtenidos por dominio/polaridad. Estos datos se calculan filtrando los datos de test predichos por dominio y calculando el valor *F1-score* para cada polaridad.

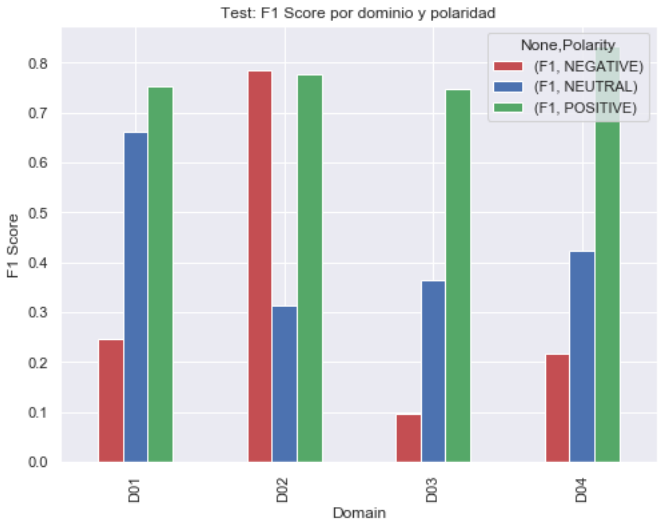

Figura 5.2: *F1-score obtenido por dominio y polaridad para los datos predichos*

Para decidir si aplicar un método de correlación paramétrico o no paramétrico, es necesario comprobar la normalidad de los datos. Para ello, en primer lugar, se muestra el histograma y el gráfico Q-Q para el número de tweets y la puntuación *F1-score* en la figura 5.3. El gráfico Q-Q compara de forma lineal una distribución normal perfecta y los datos deseados, lo que permite ver de forma gráfica la normalidad de los datos.

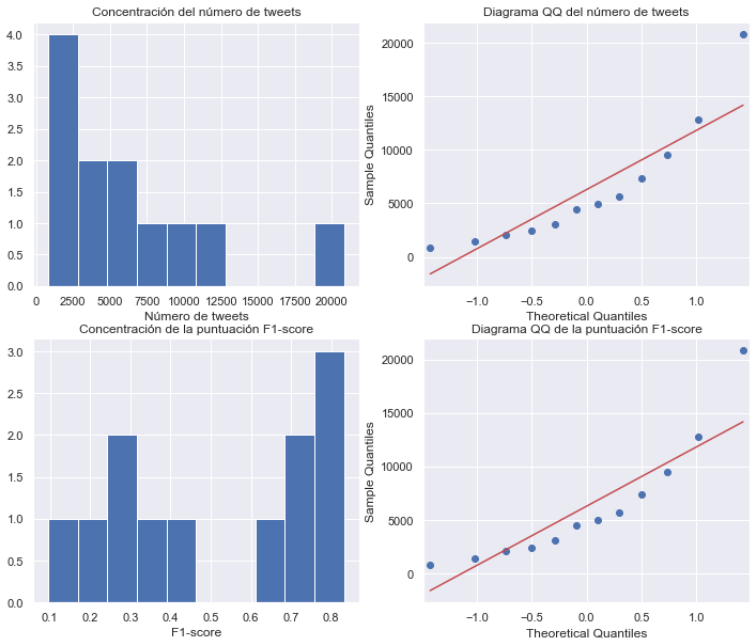

Figura 5.3: *Histograma y diagrama Q-Q para el número de tweets y F1-score*

Observando las gráficas se puede apreciar que el número de muestras es pequeño, lo que impide determinar de forma visual si los datos describen una distribución normal. Para poder determinar de forma más precisa la normalidad de los datos se aplica el test *shapiro*, estableciendo un nivel de significación de 0.05.

- Datos: número de tweets por dominio/polaridad:
  - $Statistics = 0.836$, $p\_value = 0.025$
- Datos: *F1-score* por dominio/polaridad:
  - $Statistics = 0.878$, $p\_value = 0.083$

El test de *shapiro* establece como hipótesis nula (H0) que los datos describen una distribución normal, si el *p-valor* obtenido es mayor o igual que el nivel de significación, entonces no se puede rechazar H0 y por lo tanto es más probable que los datos describan una relación normal. Observando los resultados se comprueba que los datos de número de tweets no siguen una distribución normal, mientras que los datos de F1 sí.

A la vista del test de normalidad y teniendo en cuenta la escasa cantidad de datos, se decide aplicar los test de correlación de *Pearson*, que sería adecuado para datos con distribución normal, y el test de *Spearman*, que sería adecuado para datos con distribución no normal.

- *Pearson Correlation: Corr = 0.6802, p_value = 0.014935*
- *Spearman Correlation: Corr = 0.7902, p_value = 0.002223*

Ambos test sugieren que existe correlación positiva entre ambas variables. En la figura 5.4 se gráfica la recta que mejor se ajusta a las dos variables.

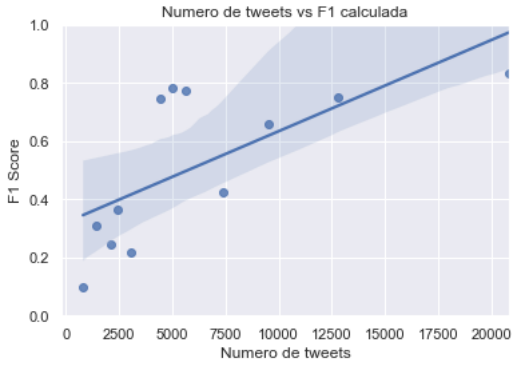

Figura 5.4: *Correlación entre el número de tweets y F1-score*

Para corroborar los resultados obtenidos en el punto anterior, se decide calcular la correlación entre el número de tweets y *F1-score* obtenida por entidad y dominio, de este modo disponemos de muchas más muestras. En la figura 5.5 se observa el *F1-score* obtenido por entidad y polaridad para cada uno de los dominios. En algunos casos se puede observar como el valor de *F1-score* es nulo, esto se produce en algunos casos en los que alguna polaridad tiene un valor muy bajo de tweets para alguna de las entidades (ver figura 4.3).

Figura 5.5: *F1-score obtenido por entidad y polaridad para cada dominio de los datos predichos*

Procediendo del mismo modo que en el caso anterior, obtenemos las gráficas para comprobar la normalidad de forma gráfica, figura 5.6.

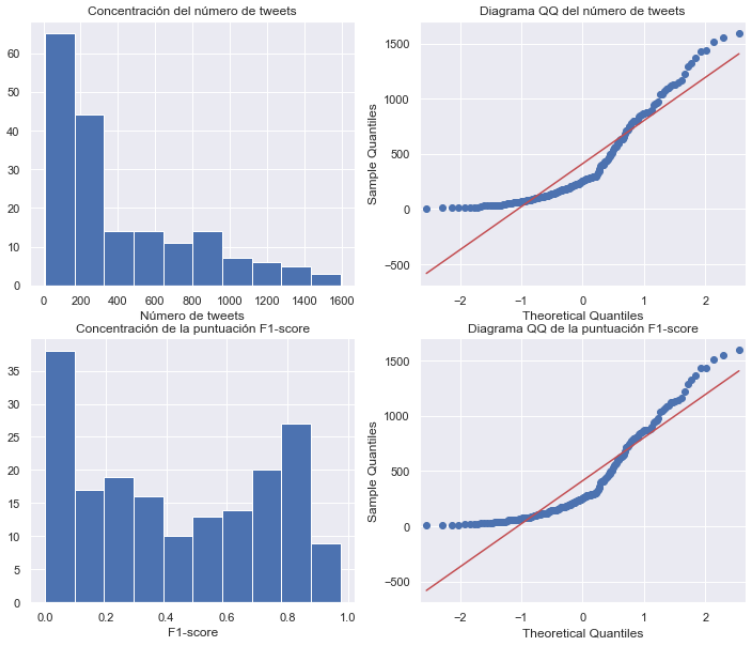

Figura 5.6: *Histograma y diagrama Q-Q para el número de tweets y F1-score por entidad*

- Datos: número de tweets por entidad/polaridad:
  - $Statistics = 0.855$, $p\_value \simeq 0$
- Datos: *F1-score* por entidad/polaridad:
  - $Statistics = 0.92$, $p\_value \simeq 0$

Se comprueba que ninguna de las dos muestras de datos cumple la condición de normalidad, por lo que se aplica el test de *Spearman*.

- *Spearman Correlation: Corr = 0.8130,* $p\_value \simeq 0$

En la figura 5.7 se muestra la correlación entre el número de tweets y *F1-score* por entidad.

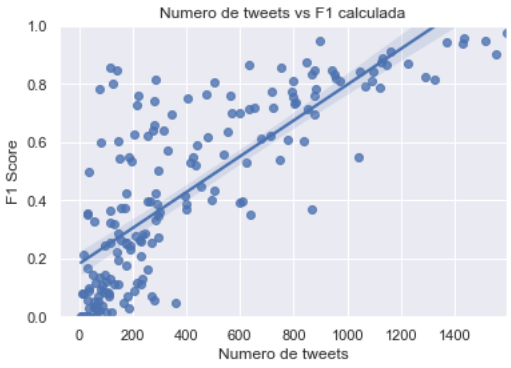

Figura 5.7: *Correlación entre el número de tweets y F1-score por entidad*

Se puede corrobora que existe correlación positiva entre el número de muestras y los valores de *F1-score* obtenidos.

A la vista de los resultados se puede apreciar como existe bastante correlación entre el número de tweets y el valor de *F1-score* obtenida. Esto parece indicar que el modelo BERT es sensible a la cantidad de muestras de las que se dispone.

5.1.2. Análisis de probabilidades obtenidas en los datos mal clasificados

A la hora de emplear el método presentado para intentar predecir la polaridad de un tweet, este le asigna tres probabilidades, una por cada polaridad. La suma de las probabilidades es igual a 1, dado que se ha empleado como función de salida de la red neuronal una función *sigmoid*. La polaridad de un tweet se decide por la probabilidad mayor entre las tres que asigna el modelo. La figura 5.8 muestra la cantidad de tweets clasificados de forma errónea y correcta por dominio y polaridad para el experimento de tweets en inglés y español y todos los dominios.

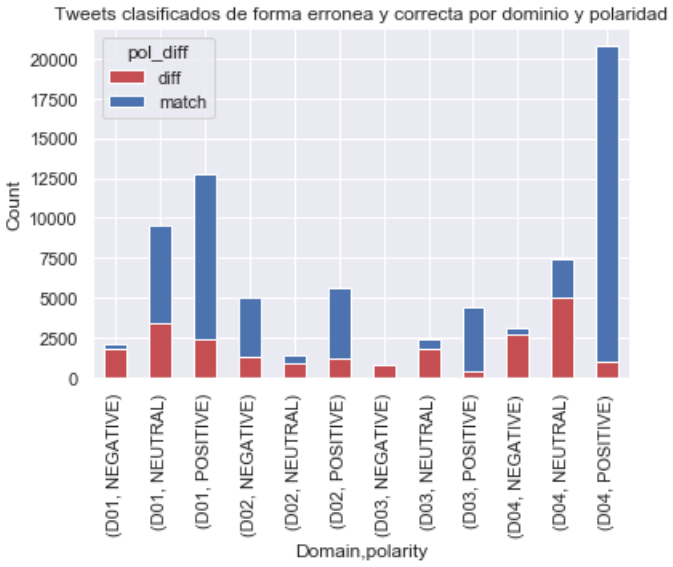

Figura 5.8: *Tweets clasificados de forma errónea y correcta por dominio y polaridad*

Comparando la gráfica 5.8 con la 4.2, que representa el número de tweets por dominio y polaridad, se puede observar cómo los mayores errores se suelen dar en los casos en los que el tamaño de la muestra es más pequeño. En relación con lo observado en las gráficas anteriores, puede resultar interesante estudiar hacía que polaridades se van los tweets mal clasificados. En la figura 5.9 se muestra la probabilidad media de los tweets mal clasificados en función del dominio.

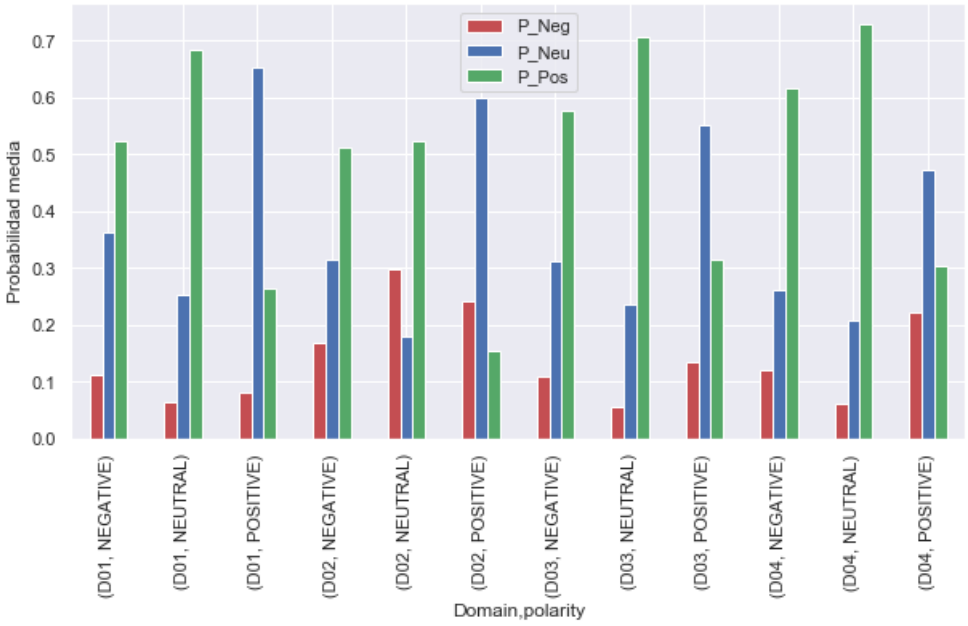

Figura 5.9: *Probabilidad media de los tweets mal clasificados en función del dominio y la polaridad*

Observamos como, por ejemplo, para el dominio D01, se tiene que los tweets negativos clasificados de forma errónea tienden a clasificarse en la mayor parte de los casos como positivos, dado que la probabilidad media en este caso es mayor. Cabe destacar que los tweets mal clasificados tienden hacía la clase más numerosa (positiva). Incluso los tweets negativos tienden hacía la clase positiva, lo que puede indicar una cierta ambigüedad a la hora de anotar los tweets, o puede indicar que el modelo no detecta fácilmente el sarcasmo, es decir, expresar sentimiento negativo empleando palabras cuyo sentimiento es positivo o neutro. Por ejemplo el tweet:

"*text 191 @FatalMoves Step 2 suggestion: compel all BMW drivers to go for driving lessons.*"

expresa sarcasmo y fue clasificado como positivo cuando está etiquetado como negativo.

5.1.3. Extracción del sentimiento de los tweets

En el punto anterior se comprobó como los tweets con polaridad negativa clasificados de forma errónea, tienden normalmente hacía la clase positiva (mayoritaria). Esto puede indicar que el modelo no detecta bien el sarcasmo (frases con sentimiento positivo pero que expresan negatividad sobre algo) que suele ser clasificado con polaridad negativa si hace referencia a la entidad en

cuestión. Para intentar comprender mejor la influencia que tiene el sentimiento que expresan los tweets y poder detectar el sarcasmo, se extraerá la puntuación de sentimiento de cada tweet. Para ello se comprobará cuantas palabras positivas y negativas tiene cada tweet. Los pasos por seguir para calcular el sentimiento de cada tweet son:

1. Extraer el lema de cada palabra.
2. Contar el número de lemas por tweet.
3. Compara cada palabra con una lista de palabras positivas y negativas.
4. Si el número de palabras positivas es superior al de negativas se establece el tweet como positivo, si el número de palabras negativas es superior al de positivas, se establece el tweet como negativo y en caso de tener el mismo número se considera neutral.

Posteriormente se comparará si el sentimiento extraído de las palabras concuerda con la polaridad reputacional del tweet. En la figura 5.10 se puede ver la tasa de acierto entre el sentimiento y la polaridad (gráfica izquierda), y la tasa de acierto obtenida por el modelo propuesto (gráfica derecha).

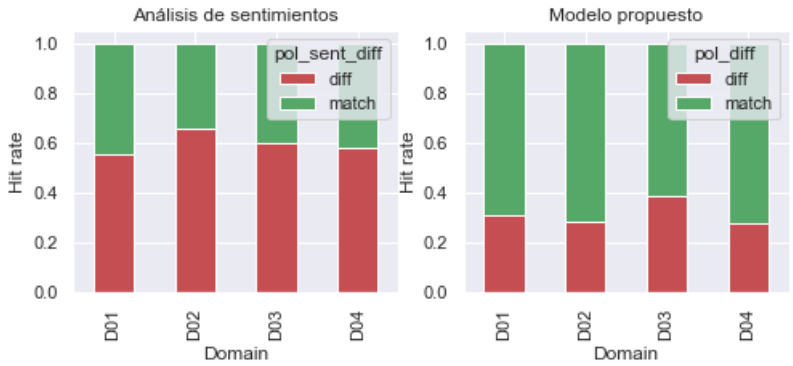

Figura 5.10: *Tasa de acierto obtenida con el análisis de sentimiento y el modelo propuesto*

Comparando las dos gráficas de la figura 5.10, se observa cómo solo analizando el sentimiento de las palabras no es suficiente para obtener buenos resultados a la hora de clasificar la polaridad reputacional. Por su parte, el modelo propuesto ofrece resultados significativamente mejores, lo que indica que aplicar modelos más complejos que tienen en cuenta la relación entre las palabras (como el modelo propuesto) se detecta mejor la polaridad reputacional.

Para poder analizar el sentimiento que tienen los tweets clasificados de forma correcta e incorrecta por el modelo propuesto, calcularemos el número medio de palabras positivas y negativas para cada caso como se muestra en la figura 5.11. Cada gráfica se hace por polaridad anotada. Cada barra equivale al número

medio de palabras positivas (verde) y negativas (rojo). Se pone un par de barras por dominio y acierto (*match*) o desacierto (*diff*).

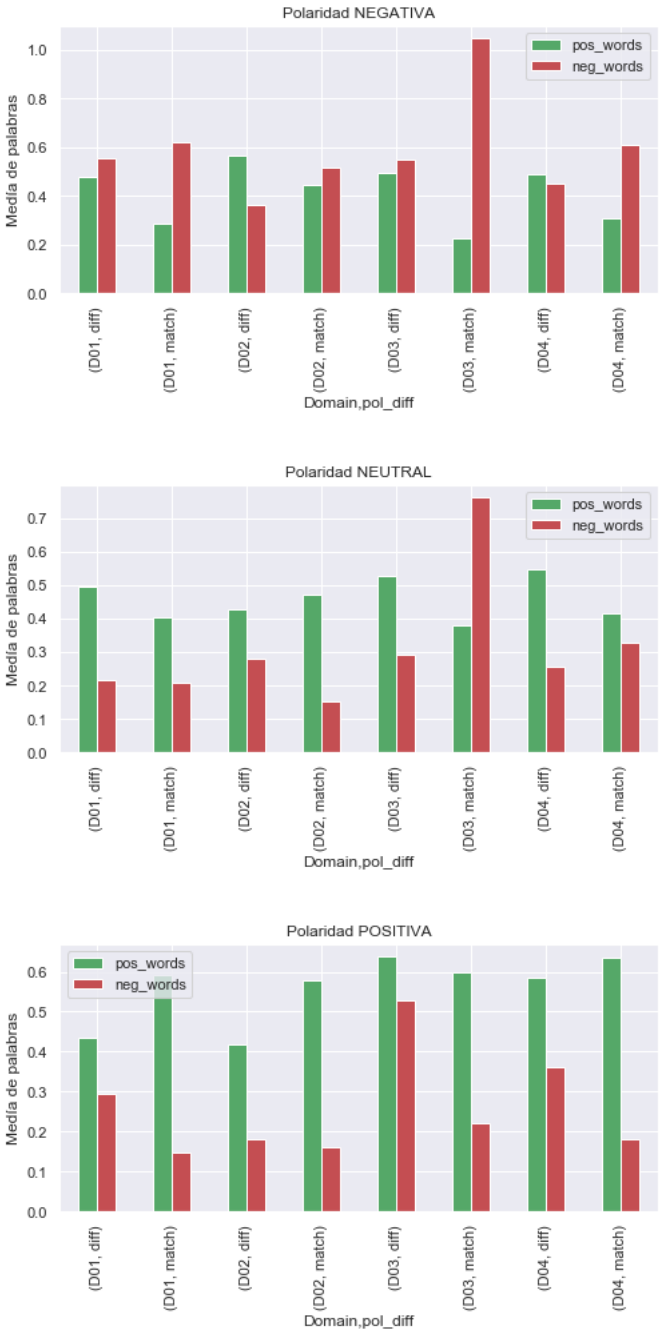

Figura 5.11: *Número medio de palabras positivas y negativas por polaridad, dominio y acierto o no*

De las gráficas mostradas en la figura 5.11 se puede extraer que en los casos en los que el modelo falla, el sentimiento medio está balanceado hacía el lado contrario de la polaridad asignada. Es decir, los tweets con polaridad positiva clasificados de forma errónea tienden a ser más negativos de medía, que los clasificados correctamente. Para el caso de los tweets con polaridad negativa el desbalanceo se da con mayor énfasis. Esto puede ser debido por el "sarcasmo", dado que este suele consistir en frases con sentimiento positivo que expresan negatividad. En el caso de los tweets neutrales, la diferencia de sentimiento es más difusa, teniendo en algunos casos un sentimiento más negativo los tweets clasificados de forma correcta. De todo lo comentado se puede deducir que los resultados del modelo propuesto dependen en buena parte del sentimiento de las palabras. No obstante, el modelo propuesto tiene en cuenta otros factores como la relación y orden de las palabras que permiten realizar una mejor clasificación.

5.2. Comparación de los resultados obtenidos con estudios anteriores

El cuadro 5.2 compara los mejores resultados de *F-score* publicados hasta ahora para la tarea de clasificar la polaridad reputacional en el conjunto de datos RepLab 2013.

Método	*F-score*
Peetz et al. 2016 (Mejor resultado)	0.553
Débilmente supervisado - Propagación (similitud y frecuencia por pares)	0.526
Supervisado - PMI y Entidad Dependiente	0.586
Modelo propuesto (BERT)	**0.62**

Cuadro 5.2: *Comparación de resultados con el estado del arte*

Los resultados expuestos en el cuadro 5.2 corresponden a Peetz et al. (2016), que plantea un sistema basado en SVM entrenado en características de mensajes y recepción y en un escenario dependiente de la entidad; y a Giachanou et al. (2019), que platea dos enfoques, supervisado basado en PMI (*Pointwise Mutual Information*) empleando entidades dependientes y débilmente supervisado basado en propagación de sentimiento en tweets similares. Con el enfoque propuesto en el presente trabajo basado en el modelo del lenguaje BERT, superamos la mejor nota hasta el momento ofrecida por el modelo basado en PMI, con una mejora relativa de 5.8 % en términos de *F1-score* (0.62 vs 0.586).

A la vista de los resultados obtenidos, se puede decir que el empleo de BERT como clasificador en la tarea de polaridad reputacional mejora los resultados. Por otro lado, se ha empleado el modelo BERT "tal cual", es decir, sin ningún tipo de adaptación, como podría ser un pre-entrenamiento sobre tweets en lugar de texto de Wikipedia y BooksCorpus (el modelo original se entrenó con texto de Wikipedia y BooksCorpus), o refinamiento para el problema de la polaridad

reputacional, como podría ser un detector de "sarcasmo". Por el contrario, los sistemas con los que hemos comparado sí que emplean sistemas para adaptarse al problema de polaridad reputacional. Esto otorga al sistema propuesto, por un lado, la posibilidad de mejora y por otro una complejidad menor que otros sistemas.

Capítulo 6

Conclusiones y trabajo futuro

En este documento presentamos un método basado en *contextual word embeddings* para estimar la polaridad reputacional de tweets. En concreto se emplea el modelo BERT, uno de los modelos que mejores resultados ha ofrecido en cuanto a estado del arte de modelos del lenguaje se refiere. Para poder emplear el modelo BERT como clasificador ha sido necesario añadir a la capa de salida de BERT una capa extra *softmax*, conectada directamente a la salida correspondiente al token de entrada [CLS]. Posteriormente el modelo resultante es entrenado (*fine-tuning*) con el conjunto de datos de entrenamiento y testeado con el conjunto de datos de test, ambos conjuntos facilitados por la colección de tweets RepLab 2013, que ofrece una de las mejores colecciones de tweets para el análisis de la polaridad reputacional. Se han realizado varios experimentos filtrando los conjuntos de datos por idioma y por dominio.

El método propuesto ha demostrado ser efectivo, mejorando los resultados obtenidos en estudios anteriores sin necesidad de emplear un modelo BERT "refinado" para el objetivo que nos atañe, como podría ser uno pre-entrenado con tweets, o añadiendo complejidad a la red neuronal resultante, como podría ser un detector de sarcasmo. Los mejores resultados obtenidos correspondientes a los experimentos realizados con todos los dominios y variando el idioma de los tweets. Por su parte las versiones de BERT han sido *bert-base-cased y bert-base-multilingual-cased*. El valor de *F1-score* obtenido ha sido de 0.62, que corresponde con una mejora relativa del 5.8 %, respecto a los mejores resultados obtenidos hasta el momento, que corresponden a Peetz et al. (2016).

Para futuros trabajos sería interesante, en primer lugar, emplear un modelo BERT más específico para la tarea en cuestión, como podría ser uno pre-entrenado con tweets, incluso podría incluir emoticonos que se emplean para expresar sentimiento, esto ayudaría a que el modelo se adaptase mejor a las expresiones y formas de escribir típicas de Twitter. Por otro lado, se ha descrito el impacto negativo que tiene el sarcasmo sobre la tarea de clasificar tweets en base a la polaridad reputacional, por lo que sería interesante probar algún mecanismo para detectar sarcasmo y añadirlo al método planteado. La flexibilidad de los modelos basados en redes neuronales permite agregar múltiples entradas a un

mismo modelo, así como múltiples salidas.

Además de las métricas que se han empleado para mostrar los resultados se podrían emplear otras métricas mejoradas, como *F(R,S)*, que es una combinación de las métricas de *Reliability* y *Sensitivity* que permite considerar la relación entre las distintas clases. Otra posible métrica es CEM (*Closeness Evaluation Measure*) Amigó et al. (2020), que está especialmente diseñada para problemas de "clasificación ordinal" y ha demostrado ser una de las mejores alternativas por robustez y por capturar mejor que otras métricas aspectos de calidad.

Bibliografía

(2019). BERT Git repo: https://github.com/google-research/bert.

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pages 30–38, Portland, Oregon. Association for Computational Linguistics.

Amigó, E., Carrillo de Albornoz, J., Chugur, I., Corujo, A., Gonzalo, J., Martín, T., Meij, E., de Rijke, M., and Spina, D. (2013). Overview of RepLab 2013: Evaluating Online Reputation Monitoring Systems. In *Proceedings of the Fourth International Conference of the CLEF initiative*, pages 333–352.

Amigó, E., Corujo, A., Gonzalo, J., Meij, E., and Rijke, M. (2012). Overview of replab 2012: Evaluating online reputation management systems. *CEUR Workshop Proceedings*, 1178.

Amigó, E., Gonzalo, J., Mizzaro, S., and de Albornoz, J. C. (2020). An effectiveness metric for ordinal classification: Formal properties and experimental results.

Ankit and Saleena, N. (2018). An ensemble classification system for twitter sentiment analysis. *Procedia Computer Science*, 132:937 – 946. International Conference on Computational Intelligence and Data Science.

Bollen, J., Mao, H., and Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1).

Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C., and Mercer, R. L. (1992). Class-based n-gram models of natural language. *Computational Linguistics*.

Cossu, J.-V., Bigot, B., Bonnefoy, L., Morchid, M., Bost, X., Senay, G., Dufour, R., Bouvier, V., Torres-Moreno, J.-M., and El-Bèze, M. (2013). Lia@ replab 2013. In *Lia@ replab 2013*.

Davidov, D., Tsur, O., and Rappoport, A. (2010). Enhanced sentiment learning using twitter hashtags and smileys. In *Coling 2010: Posters*, pages 241–249, Beijing, China. Coling 2010 Organizing Committee.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Farzindar, A. and Inkpen (2013). Natural Language Processing for Social Media. In *Natural Language Processing for Social Media (Synthesis Lectures on Human Language Technologies)*, pages 33–41.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Bradford Books.

Filgueiras, J. and Amir, S. (2013). Popstar at replab 2013: Polarity for reputation classification. In *CLEF*.

Giachanou, A., Gonzalo, J., and Crestani, F. (2019). Propagating sentiment signals for estimating reputation polarity. *Information Processing and Management*, 56(6):102079.

Giachanou, A., Gonzalo, J., Mele, I., and Crestani, F. (2017). Sentiment propagation for predicting reputation polarity.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. *Processing*, 150.

Hangya, V. and Farkas, R. (2013). Filtering and polarity detection for reputation management on tweets. In *Filtering and Polarity Detection for Reputation Management on Tweets*.

Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 174–181, Madrid, Spain. Association for Computational Linguistics.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD04, New York, NY, USA. Association for Computing Machinery.

JOSHI, P. (2020). A step-by-step nlp guide to learn elmo for extracting features from text.

Kaptein, R. (2012). Learning to analyze relevancy and polarity of tweets. In *CLEF*.

Karlgren, J., Sahlgren, M., Olsson, F., Espinoza, F., and Hamfors, O. (2012). Profiling reputation of corporate entities in semantic space : Notebook for replab at clef 2012. In *CLEF 2012 Evaluation Labs and Workshop Online Working Notes*. QC 20130204.

Kim, S.-M. and Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING04, USA. Association for Computational Linguistics.

Kiritchenko, S., Zhu, X., and Mohammad, S. (2014). Sentiment analysis of short informal text. *The Journal of Artificial Intelligence Research (JAIR)*, 50.

López, A. M., Fernández, J., Gómez, J. M., Martínez-Barco, P., and Moreda, P. (2013). Dlsi-volvam at replab 2013: Polarity classification on twitter data. In *Polarity Classification on Twitter Data*.

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

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Class-based n-gram models of natural language.

Ng, H. T., Goh, W. B., and Low, K. L. (1997). Feature selection, perceptron learning, and a usability case study for text categorization. *SIGIR Forum*, 31(SI).

Nielsen, F. A. (2011). A new ANEW: evaluation of a word list for sentiment analysis in microblogs. *CoRR*, abs/1103.2903.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86.

Peetz, M.-H., [de Rijke], M., and Kaptein, R. (2016). Estimating reputation polarity on microblog posts. *Information Processing and Management*, 52(2):193 – 216.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *CoRR*, abs/1802.05365.

Radford, A. (2018). Improving language understanding by generative pre-training. In *Improving Language Understanding by Generative Pre-Training*.

Ruiz, M. E. and Srinivasan, P. (1999). Hierarchical neural networks for text categorization. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR99, New York USA. Association for Computing Machinery.

Shamanth Kumar, Fred Morstatter, H. L. (2014). Twitter Data Analytics. In *Twitter Data Analytics*.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565, Baltimore, Maryland. Association for Computational Linguistics.

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A. (2010). Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12).

Turney, P. D. (2002). Thumbs up or thumbs down semantic orientation applied to unsupervised classification of reviews. *CoRR*, cs.LG/0212032.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *ArXiv*, abs/1706.03762.

Villena-Román, J., Lana-Serrano, S., Moreno, C., García-Morera, J., and González, J. C. (2012). Daedalus at replab 2012: Polarity classification and filtering on twitter data. In *CLEF*.

Xu, J., Chen, D., Qiu, X., and Huang, X. (2016). Cached long short-term memory neural networks for document-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1660–1669, Austin, Texas. Association for Computational Linguistics.

Yang, C., Bhattacharya, S., and Srinivasan, P. (2012). Lexical and machine learning approaches toward online reputation management. In *CLEF*.

Zhang, D., Wang, J., and Zhao, X. (2015). Estimating the uncertainty of average f1 scores.