Trabajo Fin de Máster: Desarrollo de un sistema para la detección del machismo en redes sociales

Trabajo Fin de Máster

Francisco Miguel Rodríguez Sánchez

Trabajo de investigación para el

Máster en Lenguajes y Sistemas Informáticos

Universidad Nacional de Educación a Distancia

Dirigido por

Dr. Jorge Amando Carrillo de Albornoz

Dr. Laura Plaza Morales

Junio 2019

Agradecimientos

Agradecimientos si procede.

Resumen

En el presente trabajo se propone una arquitectura para la detección del machismo en la red de microblogging Twitter. El objetivo es comprender los mecanismos y señales textuales que conlleven actitudes o lenguaje machista en castellano. Para ello, se compone un corpus que contiene texto con contenido y expresiones machistas utilizando como fuente de datos la red social Twitter. Para la creación del corpus, se desarrollará una herramienta que recopilará los mensajes creados en Twitter que contengan distintos términos que pueden conllevar comportamientos machistas. Las expresiones o términos que se buscan son aquellas que, de un modo u otro, minusvaloran el papel de las mujeres en nuestra sociedad, incentiven el abuso o acoso hacia las mujeres, o no les permita expresarse libremente. Existen gran cantidad de expresiones y términos que se utilizan a diario de modo consciente o no que minimizan el papel de la mujer en la sociedad. Expresiones como "feminazi", "niñata" o "a fregar" han sido utilizadas para recopilar los tweets. Asimismo, utilizando el corpus generado, se desarrolla un sistema de clasificación que permite detectar lenguaje machista de un modo automatizado. Se emplean distintos algoritmo de clasificación y se establecen dos líneas base que sirven como punto de partida para la evaluación del sistema.

Finalmente, se ha realizado una evaluación exhaustiva del sistema de clasificación sobre el corpus creado para determinar su rendimiento. Los resultados obtenidos muestran que el procedimiento empleado es capaz de detectar el machismo en el corpus utilizado con un buen acierto. Sin embargo, se han detectado distintas limitaciones y problemas debido a la aproximación empleada.

Abstract

The present work defines a new architecture to detect sexism at Twitter. The main goal is to understand the mechanism and textual signals which imply sexist attitudes or language of this kind in spanish. We create a new corpus which contains text with expressions and sexist attitudes using the social network Twitter as main datasource. To create the corpus, a specific tool is developed to collect all the messages created at Twitter containing certains terms related to sexist attitudes. We search for expressions which underestimate the role of women in our society, encourage the harassment towards them or limit their freedom of speech. There are many expressions used in a daily basis that underestimate the role of women in society. Terms such as "feminazi", "niñata" o "a fregar" have been used to collect tweets in our corpus. We also develop a classification system which allows us to identify sexist language in an automatic way. The system employs several machine learning algorithms and establish two baselines used as reference points for the system's evaluation.

Finally, an extensive evaluation is perfomed on the corpus to determine the perform of our approach in the classification task. The results obtained shows this procedure is able to identify sexism properly in the corpus used. However, some limitations and problems have been detected due to the approach used.

Índice general

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

El presente capítulo tiene como objetivo presentar al lector la motivación y los objetivos del presente trabajo. Además, se indica la estructura y los principales puntos abordados en el trabajo fin de máster.

1.1. Motivación

Con el rápido crecimiento de las redes sociales, la comunicación entre personas de diferentes culturas en todo el mundo se produce de forma mucho más directa y sencilla. Esto provoca un gran aumento de "ciber" conflictos entre las personas que utilizan con frecuencia este tipo de plataformas. Con millones de contribuciones generadas diariamente por los usuarios de este tipo de herramientas, resulta impracticable y poco escalable implementar una política manual para detectar prácticas como el machismo o la xenofobia. Pese a esto, empresas como Facebook han anunciado planes para contratar varios miles de empleados encargados de moderar el contenido de la plataforma (Quartz, 2017). Pese a dedicar muchos esfuerzos y recursos, las grandes compañías como Twitter encuentran gran cantidad de dificultades para afrontar el problema (Atlantic, 2016) debido a la gran cantidad de posts que no pueden ser mediados por sus moderadores. Además, han impulsado fuertes iniciativas para responder a las críticas recibidas por no atajar el problema con la suficiente contundencia. Twitter, por ejemplo, ha aplicado políticas para prohibir el uso de sus plataformas a quienes ataquen a personas o grupos sociales (Twitter, 2018b). La importancia de este problema junto con la gran cantidad de información generada por los usuarios hace necesaria la creación de sistemas y herramientas que puedan automáticamente

detectar el contenido inapropiado en redes sociales.

Un nuevo estudio realizado en EEUU (Duggan, 2017) sostiene que el 41 % de personas encuestadas había sufrido personalmente algún tipo de discriminación o acoso online, de las cuales el 18 % había sufrido algún tipo de acoso grave, por ejemplo debido a su género (8 %). De igual modo, las mujeres tienen más de el doble de probabilidades de sufrir acoso debido a su género (Duggan, 2017).

El problema es aún más grave si se tiene en cuenta que un abuso verbal en redes sociales puede acarrear un acto de violencia física. No solo es importante por el propio mensaje, este tipo de plataformas permiten propagar un mensaje machista que incita al odio, al menosprecio o a la desigualdad. De hecho, no es inusual que contenidos machistas hacia las mujeres sean trasladados a acciones violentas. Por ejemplo, algunos estudios sociales como (Fulper y Rowe, 2014) demuestra la existencia de una correlación entre el número de violaciones y el número de tweets machistas por estado en USA. Esto sugiere que las redes sociales pueden ser utilizadas como detector de violencia machista e incluso pueden ayudar a anticiparla o prevenirla.

Amnistía internacional publicó recientemente un estudio donde denuncia este hecho (International, 2017). En el reporte, se explica cómo para muchas mujeres Twitter es una plataforma donde la violencia y al abuso contra ellas florece, en la mayoría de los casos, sin ninguna consecuencia. Según este informe, Twitter está fallando como empresa a la hora de respetar los derechos de la mujer en línea. En lugar de reforzar las voces de las mujeres, la violencia y el abuso que experimentan en la plataforma hace que las mujeres se autocensuren a la hora de postear, limiten sus interacciones e incluso les hace abandonar Twitter por completo. De este modo, la violencia y el abuso que muchas mujeres experimentan en Twitter tiene un efecto perjudicial en su derecho a expresarse en igualdad, libremente y sin miedo.

Todo lo expuesto justifica, sin duda, la realización de este trabajo, en el que se propone una arquitectura para la detección automática del machismo. Todos los estudios listados justifican la necesidad de detectar y filtrar de un modo automatizado el contenido que incita o promueve el machismo. En concreto, el lenguaje machista o sexista, ocupa gran parte de este discurso en sitios webs como Twitter. Mientras que en la mayoría de las plataformas el uso de este tipo de lenguaje está prohibido, el tamaño de estas redes hace imposible controlar todo el contenido que generan. Resulta adecuado, por

tanto, afirmar que las posibilidades y ventajas que proporcionaría un sistema capaz de detectar este tipo de actitudes en en texto de manera automática supondrían un beneficio sustancial para los usuarios y consumidores de redes sociales e internet en general.

1.2. Propuesta y objetivos

El trabajo realizado en este proyecto presenta un sistema de clasificación automático para la detección del machismo en redes sociales. El abuso online se ha convertido en un gran problema, especialmente por el anonimato y la interactividad de la web que facilita el incremento y permanencia de este tipo de abusos. Se trata de un campo en el que ha aumentado la producción científica enormemente durante este mismo año y donde se han desarrollado competiciones con gran participación por parte de la comunidad científica (E. Fersini y Anzovino, 2018b).

Los atributos utilizados para la tarea de clasificación se agrupan en 3 tipos: variables categóricas, numéricas y texto. Para cada atributo, se aplican diferentes métodos como la tokenización, el escalado o la sustitución de emoticonos. Tras esto, se unifican los distintos tipos de atributos procesados en un conjunto de datos común que será la entrada de la última fase. En la última etapa, se emplean algoritmos de clasificación supervisada con la intención de obtener un modelo predictivo capaz de detectar las señales textuales que expresan lenguaje machista. A lo largo del trabajo, se presenta el ciclo completo para la recolección de datos, preprocesamiento y construcción del sistema de clasificación para el lenguaje.

Para el correcto funcionamiento de estos sistemas de clasificación supervisados se necesitan ejemplos previamente etiquetados con los que entrenar el algoritmo. En este caso, se ha desarrollado un corpus mediante la búsqueda de 29 términos o expresiones utilizando como fuente de datos la red social Twitter. Este corpus está compuesto por 3600 mensajes y ha sido etiquetado en 3 categorías: MACHISTA, NO_MACHISTA Y DUDOSO. De este modo, este trabajo realiza una aportación importante mediante este corpus etiquetado compuesto por un gran número de expresiones y actitudes machistas.

1.3. Estructura del documento

En este capítulo se estructuran los capítulos que componen el presente trabajo fin de máster.

Capítulo 1. Introducción. Este capítulo introduce los principales motivos que han llevado a la realización de este trabajo, así como la problemática y el estado actual de la disciplina. Por último, se presentan las diferentes contribuciones del trabajo realizado.

Capítulo 2. Estado del arte. Este capítulo describe en mayor detalle la disciplina que nos ocupa, presentando su origen y su historia hasta el presente. Se muestran las técnicas actuales más utilizadas para resolver las tareas más relevantes del tema abordado, así como sus debilidades.

Capítulo 3. Herramientas utilizadas. En este capítulo se describen todas las herramientas y tecnologías que han sido necesarias para elaborar el proyecto. Se realiza una definición de cada una de ellas y se describe el papel que desempeñan dentro del trabajo desarrollado.

Capítulo 4. MeTwo dataset (Machismo and Sexism Twitter Identification dataset) En este capítulo se describe en profundidad la metodología seguida para componer el corpus con texto y expresiones machistas.

Capítulo 5. Sistema. En este capítulo se describe en profundidad el sistema de clasificación propuesto.

Capítulo 6. Evaluación y discusión. Este capítulo describe la metodología utilizada para evaluar la propuesta realizada, a la vez que presenta los resultados obtenidos al evaluar el método propuesto en diferentes tareas y sobre colecciones de evaluación de distintos dominios. Además, se analiza y discute en profundidad los resultados obtenidos en la evaluación presentada anteriormente.

Capítulo 7. Conclusiones y trabajo futuro. Este capítulo recopila las diferentes conclusiones extraídas del trabajo realizado, y propone algunas líneas de trabajo futuro.

Capítulo 2

Estado del arte

El presente capítulo tiene como objetivo presentar al lector la detección del lenguaje machista en redes sociales. Para ello, se realizará una revisión de los trabajos más relevantes en la tarea de detección de lenguaje abusivo y machista, en los que se analizarán los orígenes de esta tarea, las soluciones técnicas y las aportaciones más relevantes.

2.1. Clasificación de textos

El procesamiento del lenguaje natural (*"Natural Language Processing"*, NLP) tiene como objetivo fundamental el desarrollo de métodos que permitan a los computadores realizar tareas relacionadas con el lenguaje humano, como la comunicación o el procesamiento de textos. La principal diferencia del NLP con el resto de líneas de investigación relacionadas con el análisis de datos o la inteligencia artificial es la necesidad de un conocimiento del lenguaje en todas sus aplicaciones. Elementos clave del lenguaje como la fonética, la fonología, la morfología, la sintaxis, la semántica, la pragmática y la discursiva son esenciales en cualquier técnica de procesamiento del lenguaje (Jurafsky, 2009, Capítulo 1).

Una de las áreas más importantes de investigación relacionadas con el NLP es la clasificación de textos o documentos. De un modo general, se conoce como clasificación automática (Jurafsky, 2009, Capítulo 4) a la tarea de asignar una o varias categorías predefinidas sobre una colección de instancias a clasificar. Del mismo modo, la clasificación de textos (Jurafsky, 2009, Capítulo 4) se puede entender como aquella tarea en la que un documento o texto es etiquetado como perteneciente a un determinado conjunto. Este

tipo de técnicas se utilizan para un gran número de aplicaciones, tales como:

- Indexación para sistemas de recuperación de información (Jurafsky, 2009, Chapter 17)
- Detección de *spam* (Wang, 2010; Jurafsky, 2009)
- Identificación del lenguaje (Jurafsky, 2009, Chapter 3)
- Análisis de sentimientos (Jurafsky, 2009, Chapter 4)
- Organización de documentos (De Mauro, 2016)
- Desambigüación del sentido de las palabras (Russell y Norvig, 2002)
- Filtrado de textos (Mark Hepple,)

Formalmente, el problema se define como un texto o documento d que puede pertenecer a un conjunto fijo de clases $C = \{c_1, c_2, ..., c_i\}$. La salida del sistema es la predicción la clase $c \in C$.

Para resolver el problema de la clasificación de textos existen dos enfoques principales: uno basado en reglas y otro mediante algoritmos de clasificación supervisado.

Los sistemas basados en reglas utilizan patrones predefinidos por un experto para crear un conjunto de pautas mediante la combinación de palabras u otros atributos (Liddy, 2001). En este tipo de arquitecturas, la precisión puede ser alta siempre que estas reglas estén cuidadosamente seleccionadas por un experto. Sin embargo, dichos sistemas resultan muy costosos de construir y mantener. Además, se trata de sistemas muy específicos para un dominio o problema concreto, y difícilmente trasladables a un dominio distinto.

El aprendizaje supervisado se construye sobre un conocimiento a priori. Se debe disponer de un conjunto de documentos de ejemplo para cada una de las categorías consideradas. Después de una etapa de entrenamiento, el sistema queda ajustado de modo que, ante nuevos ejemplos, el algoritmo es capaz de clasificarlos en alguna de las clases existentes. Para este tipo de sistemas se utilizan distintos modelos de clasificadores: *Naive Bayes*, *Regresión logística*, *SVM*, *redes neuronales*, etc (Aurangzeb Khan, 2010).

Para construir cualquier clasificador de textos o documentos es necesario seguir los siguientes pasos:

- Extraer los atributos o *features* necesarias para realizar una representación fiel del texto y que permita la utilización de un algoritmo de

clasificación

- Desarrollar procedimientos por los cuales los documentos puedan ser clasificados automáticamente dentro de categorías.
- Evaluar la calidad de la clasificación en relación a algún criterio.

2.1.1. Representación textual

Modelo de espacio vectorial

La representación del texto es un paso fundamental para el procesamiento automático de textos. Una representación fiel al contenido del documento, que incluya la información necesaria para extraer conocimiento útil, será clave para el desarrollo de una arquitectura con un rendimiento adecuado. En este proceso, se han de tener en cuenta las especificaciones de los algoritmos que se empleen a continuación.

En esta fase, se definen todos los atributos utilizados en el paso posterior por el algoritmo de clasificación. Los atributos seleccionados o generados a partir de los originales serán los que marquen el éxito de la arquitectura completa. La elección del algoritmo de clasificación para los pasos posteriores influirá de un modo mucho menos significativo. Por ejemplo, en (Nina-Alcocer, 2018) y (Endang Wahyu Pamungkas y Patti, 2018) se utiliza el mismo algoritmo de clasificación pero los resultados son muy diferentes debido a los atributos utilizados.

Un modelo de representación muy utilizado se conoce como modelo de representación vectorial (Fresno, 2006). Mediante esta representación, los documentos se modelan como vectores dentro de un espacio euclídeo. De este modo, se pueden aplicar operaciones de distancia entre vectores, como indicador de su cercanía según el contenido textual. En la siguiente imagen se muestra un ejemplo en dos dimensiones:

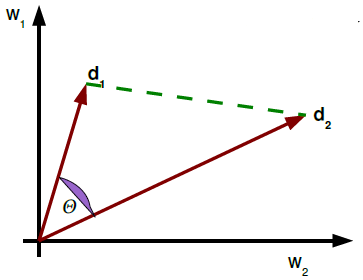

Figura 2.1: Representación vector de documentos

En este caso, se tendría un vocabulario con únicamente dos rasgos w_1 y w_2 que conforman el espacio en el que se encuentran los documentos o textos d_1 y d_2. De este modo, se pueden emplear medidas de distancia, como la distancia euclídea o la distancia coseno, para comparar ambos documentos.

Utilizando este modelo, un texto quedará representado como una combinación lineal de vectores, donde cada coeficiente representa la relevancia de cada rasgo en el contenido del texto, calculado con una función de pesado. Para un texto d, un vocabulario de tamaño n: $\vec{d} = t_1 j\vec{t_1} + ... + t_n j\vec{t_n}$. Para el cálculo de la relevancia de cada rasgo $t_n j$, se utilizará una función de pesado. Una de las más utilizadas se conoce como TF-IDF (frecuencia del termino x frecuencia inversa del documento) y se calcularía del siguiente modo:

$$TF - IDF(\vec{t_i}, \vec{d_j}) = f_{ij} \log(\frac{N}{d_f(\vec{t_i})})$$

donde N es la dimensión del corpus (en este caso número de tweets), f_{ij} la frecuencia del término en el documento y $d_f(\vec{t_i})$ el número de documentos (en este caso el tweet) en los que aparece el término.

Representación semántica mediante modelo de espacio vectorial

En el apartado anterior, se ha descrito un procedimiento para la representación textual que permite convertir la información contenida en el texto como una combinación lineal de vectores, donde cada coeficiente representa la existencia o no de los términos contenidos en el vocabulario del corpus. A pesar de ser uno de los métodos más simples y utilizados para el problema de clasificación de textos, los documentos y textos que se quieren representar tienen un contenido semántico que no queda representado con este tipo de métodos. Es decir, este método no es capaz de capturar información acerca del significado que tienen los términos que contienen los documentos. Por ejemplo, pensemos en los 3 documentos de ejemplo:

d_1 =“*Pizza margarita*”

d_2 =“*Pizza calzone*”

d_3 =“*Macarrones*”

Si se realiza una representación de los documentos según el modelo descrito anteriormente (con el número de términos como función de pesado) se

Términos	**Pizza**	**margarita**	**calzone**	**Macarrones**
d_1	1	1	0	0
d_2	1	0	1	0
d_3	0	0	0	1

Tabla 2.1: Matriz documento - término

obtendría la matriz de la tabla 2.1. Como se puede observar, los documentos d_1 y d_2 tienen en común un atributo mientras que el documento d_3 no tendría ningún rasgo común con el resto de documentos. Sin embargo, si se tiene en cuenta el significado o la representación semántica de los documentos, se concluiría que todos los documentos indican comida italiana y, por tanto, todos tienen en común ese rasgo semántico. Esto indica que si un documento se trata únicamente como una sucesión de términos, se perderán los conceptos a los que hace referencia.

Los métodos para la generación automática de atributos semánticos fueron desarrollados alrededor del año 1990 con el objetivo de capturar la información semántica presente en los documentos. Uno de los algoritmos más populares dentro de esta familia se conoce como LSA (*Latent Semantic Analysis*) (Deerwester, 1990) que describe a cada palabra o conjunto de palabras en un espacio vectorial de dimensión reducida en el que las palabras semánticamente similares se encontrarán más próximas. Para ello, este método toma como entrada una matriz término-documento a la que se le aplica una descomposición de valores singulares obteniendo así una representación vectorial tanto de los términos como de los documentos. De este modo, se puede cuantificar la similitud semántica de dos palabras mediante el uso de medidas de distancias entre vectores. Esta técnica fue rápidamente aplicada a distintos dominios como modelo cognitivo (Landauer y Dumais, 1997), como modelo de lenguaje (Jurafsky y Coccaro, 1998) o como calificador automático (Rehder, 1998).

Word Embedding

In the previous sections we saw how to represent a word as a sparse, long vector with dimensions corresponding to the words in the vocabulary, and whose values were tfidf or PPMI functions of the count of the word co-occurring with each neighboring word. In this section we turn to an al-

ternative method for representing a word: the use of vectors that are short (of length perhaps 50-500) and dense (most values are non-zero).

It turns out that dense vectors work better in every NLP task than sparse vectors. While we don?t complete understand all the reasons for this, we have some intuitions. First, dense vectors may be more successfully included as features in machine learning systems; for example if we use 100-dimensional word embeddings as features, a classifier can just learn 100 weights to represent a function of word meaning; if we instead put in a 50,000 dimensional vector, a classifier would have to learn tens of thousands of weights for each of the sparse dimensions. Second, because they contain fewer parameters than sparse vectors of explicit counts, dense vectors may generalize better and help avoid overfitting. Finally, dense vectors may do a better job of capturing synonymy than sparse vectors. For example, car and automobile are synonyms; but in a typical sparse vector representation, the car dimension and the automobile dimension are distinct dimensions. Because the relationship between these two dimensions is not modeled, sparse vectors may fail to capture the similarity between a word with car as a neighbor and a word with automobile as a neighbor.

Unsupervisedly learned word embeddings have seen tremendous success in numerous NLP tasks in recent years. So much so that in many NLP architectures, they are close to fully replacing more traditional distributional representations such as LSA features.

The term word embeddings was originally coined by Bengio et al. in 2003 who trained them in a neural language model together with the model?s parameters. However, Collobert and Weston were arguably the first to demonstrate the power of pre-trained word embeddings in their 2008 paper A unified architecture for natural language processing, in which they establish word embeddings as a highly effective tool when used in downstream tasks, while also announcing a neural network architecture that many of today?s approaches were built upon. It was Mikolov et al. (2013), however, who really brought word embedding to the fore through the creation of word2vec, a toolkit enabling the training and use of pre-trained embeddings. A year later, Pennington et al. introduced us to GloVe, a competitive set of pre-trained embeddings, suggesting that word embeddings was suddenly among the mainstream.

-word embedding vs lsa (http://ruder.io/secret-word2vec/):

Word embedding models such as word2vec and GloVe gained such popularity as they appeared to regularly and substantially outperform traditional Distributional Semantic Models (DSMs). Many attributed this to the neural architecture of word2vec, or the fact that it predicts words, which seemed to have a natural edge over solely relying on co-occurrence counts.

DSMs can be seen as count models as they ?count? co-occurrences among words by operating on co-occurrence matrices. Neural word embedding models, in contrast, can be viewed as predict models, as they try to predict surrounding words.

In 2014, Baroni et al. [11] demonstrated that, in nearly all tasks, predict models consistently outperform count models, and therefore provided us with a comprehensive verification for the supposed superiority of word embedding models.

2.1.2. Clasificación

Como ya se introdujo en apartados anteriores, la clasificación automática de documentos se puede entender como aquella tarea en la que un documento, o una parte del mismo, es etiquetado como perteneciente a un determinado conjunto, grupo o categoría predeterminada.

Los métodos de clasificación supervisados utilizan un conjunto de documentos de ejemplo para cada una de las categorías que presenta la variable objetivo (a clasificar). Estos algoritmos, realizan una etapa de entrenamiento donde se presentan los patrones de ejemplo de modo que ante futuros patrones, el algoritmo será capaz de clasificar en alguna de las clases contenidas en el conjunto de ejemplo. Dentro de este proceso, existen muchas variables que influirán en los resultados del sistema como el tamaño del conjunto de ejemplo, la elección del algoritmo de clasificación o los parámetros de inicialización del mismo.

Existen numerosos tipos de algoritmos de clasificación, a continuación se indican los más importantes para clasificación textual:

- Naive Bayes (Jurafsky, 2009, Capitulo 3): Está basado en la teoría de la decisión de Bayes: la teoría de las probabilidades condicionadas. Por tanto, el problema de la clasificación se reduce al cálculo de las probabilidades a posteriori de una clase dado un documento.
- Arboles de decisión (Breiman, 2001): Se trata de un método que a

través de un proceso recursivo de las los atributos de entrada, realiza una representación para clasificar el conjunto de datos presentado.

- Máquinas de vectores de soporte ("Support Vector Machine", SVM) (Cortes, 1995): Estos algoritmos pretenden encontrar una hipersuperficie de separación entre clases dentro del espacio de representación.
- Redes Neuronales (Goodfellow, Bengio, y Courville, 2016): Son un modelo computacional compuesto por elementos ("neuronas") interconectados entre sí que aplican una transformación a los datos para producir una salida. Es posible entrenar una red neuronal para que dada una entrada determinada (un vector de representación) produzca una salida deseada (la categoría a la que corresponde ese documento). Dentro de este tipo de algoritmos, destacan las "redes neuronales profundas" (*deep learning*) que proveen de una herramienta muy poderosa añadiendo más capas de neuronas que permiten representar funciones de mayor complejidad. Este tipo de técnicas alcanza los mejores resultados hasta la fecha en tareas como el procesado de imagen (Tan, 2019) o el procesado de textos (Devlin, 2019).
- KNN (K-Nearest Neighbour) (Guo, 2003): Este algoritmo se basa en la aplicación de una métrica que establezca la similitud entre un documento que se quiere clasificar y cada uno de los documentos de entrenamiento. La clase o categoría que se asigna al documento sería la categoría del documento más cercano según la métrica establecida.

2.2. Detección de lenguaje o discurso del odio (*hate speech detection*)

La detección del lenguaje machista o sexista está muy relacionada con la detección del lenguaje o discurso del odio en redes sociales. Existen numerosos trabajos donde se intenta detectar distintos tipos de lenguaje del odio, entre ellos el sexismo (Watanabe, 2018; Waseem, 2016b; Georgios K. y Langseth, 2018; Pinkesh Badjatiya, 2017; Zimmerman, 2018; Park y Fung, 2017; Waseem, 2016a). El lenguaje del odio se refiere al uso de lenguaje agresivo, violento u ofensivo hacia un grupo específico de personas que comparten una propiedad en común, sea esta propiedad su género, su raza, sus creencias o su religión (Thomas Davidson, 2017). Atendiendo a esta definición, se puede

considerar la detección del machismo como un caso particular del discurso del odio. Por ello, es muy interesante realizar una evaluación de los trabajos realizados en esta línea de investigación.

La detección del lenguaje del odio es una linea de investigación muy actual, datando el primer estudio evaluado en el año 2012 (Guang Xiang, 2012). En este articulo se emplea un modelo de detección de temas o categorías (*topic modelling*) que explota la concurrencia de palabras para la creación de atributos o *features* que alimentarán un algoritmo de clasificación de aprendizaje de máquina o *machine learning*. En la mayoría de trabajos previos se empleaban soluciones basadas en patrones para la clasificación de tweets. Utilizando estos métodos, el uso de expresiones coloquiales y soeces en redes sociales hace más complicado establecer las fronteras entre el uso de lenguaje ofensivo que no tiene como objetivo despreciar a ningún grupo de personas y el lenguaje del odio (Thomas Davidson, 2017). De este modo, este artículo supone un paso muy importante hacia la automatización y a los sistemas basados en algoritmos de *machine learning*.

Durante los últimos tres años, se han sucedido diferentes artículos en la temática aumentando considerablemente la producción científica en este campo. En (Waseem, 2016b) se aporta el primer corpus de referencia anotado que se utilizará posteriormente en (Waseem, 2016a; Georgios K. y Langseth, 2018; Pinkesh Badjatiya, 2017; Zimmerman, 2018; Park y Fung, 2017). Está compuesto por 16.000 *tweets* etiquetados en mensajes sexistas, racistas o sin contenido ofensivo. En este primer trabajo, se sientan las bases de las soluciones aplicadas en el resto de artículos, se utilizan atributos como los *unigramas, bigramas, trigramas* y *cuatri-gramas* y un algoritmo de regresión logística para la clasificación.

En el artículo desarrollado por el mismo autor (Waseem, 2016a) se propone una solución similar pero se amplía el corpus en 4033 *tweets* y se utiliza una plataforma de *crowdsourcing* para anotar los mensajes, lo que introduce más diversidad en los criterios del etiquetado. Según los autores, el empeoramiento de los resultados puede deberse al posible sesgo que se produce en (Waseem, 2016b), ya que los *tweets* fueron etiquetados por los autores únicamente.

En el resto de artículos que evalúan su propuesta utilizando el corpus desarrollado por (Waseem, 2016a), se utilizan redes neuronales en la etapa de clasificación y, en algunos, en la etapa de preprocesamiento. En la

solución propuesta por (Zimmerman, 2018) se aplican redes neuronales convolucionales (*CNN, Convolutional Neural Network*) para codificar el texto y extraer los atributos que se utilizarán para el clasificador final, basado también en CNNs. Esta técnica permite tener en cuenta la posición de la palabra (su contexto) para extraer los atributos de cada *tweet*. Esta misma idea junto con el uso de redes neuronales recurrentes (*RNN, Recurrent Neural Network*) se utiliza en (Pinkesh Badjatiya, 2017) para obtener los atributos en la etapa de procesamiento. En ambos artículos se consiguen mejorar los resultados alcanzados por (Waseem, 2016a) lo que afianza el uso de técnicas basadas en redes neuronales en el procesado del lenguaje.

Una idea interesante es el uso de atributos como la tendencia al racismo o al sexismo sirviéndose del historial de los usuarios. En (Georgios K. y Langseth, 2018) se demuestra como el uso de este tipo de atributos mejora notablemente los resultados. Esta misma idea se utiliza en (Despoina Chatzakouy, 2017) donde se detectan cuentas agresivas estudiando al usuario y su red de seguidores.

En todos los artículos revisados anteriormente, se trata el problema como una clasificación múltiple donde el texto se puede clasificar según las etiquetas racismo, sexismo o ninguno. Sin embargo, se podría resolver el problema con un doble clasificador, el primero detecta si el texto contiene lenguaje abusivo o no y el segundo realizaría la tarea de clasificar en contenido sexista o racista (Park y Fung, 2017).

Un desafío importante en la detección del lenguaje del odio en redes sociales es la separación entre el lenguaje ofensivo y el lenguaje que incita o promueve el odio. Davidson (Thomas Davidson, 2017) aporta un corpus etiquetado de 25.000 *tweets* para diferenciar entre estos 2 tipos de lenguaje. En su trabajo, se propone un modelo similar a (Waseem, 2016a) donde se ponen de manifiesto las dificultades de esta solución para considerar el contexto de las palabras. De este modo, si se utilizan palabras que pueden expresar odio (por ejemplo, "*gay*") en un contexto positivo, hay muchas probabilidades de que el sistema detecte odio en el texto. Los resultados serán mejorados posteriormente en (Watanabe, 2018) donde se ampliará el número de *features* y se utilizará un algoritmo basado en árboles de decisión para la tarea de clasificación.

2.3. Detección de la misoginia

La misoginia se define según la RAE como *"Aversión a las mujeres"* (RAE, b). El machismo, sin embargo, se define como "Actitud de prepotencia de los varones respecto de las mujeres" o "forma de sexismo caracterizada por la prevalencia del varón" (RAE, a). Si bien estos dos términos tienen matices distintos, tienen como denominador común la discriminación de las mujeres debido a su sexo. De hecho, existen trabajos donde se expone que la misoginia se manifiesta lingüísticamente mediante la exclusión, discriminación, hostilidad, trato de violencia objetificación o cosificación sexual (Maria Anzovino y Rosso, 2018; E. Fersini y Anzovino, 2018a). Muchas de estas señales textuales de misoginia serían aplicables del mismo modo al machismo (Aranbarri, 2014; Giraldo, 1972).

Durante este último año, se ha llevado a cabo la competición IberEval 2018 donde una de las tareas era la detección automática de la misoginia (https://amiibereval2018.wordpress.com/, 2018) (AMI, *"Automatic Misogyny Identification"*). En esta tarea se propone la labor de identificar la misoginia en *tweets* en español e inglés. En total, participaron once equipos de cinco países distintos para la detección en inglés, mientras que para la detección en castellano participaron un total de ocho equipos (E. Fersini y Anzovino, 2018b). Los artículos publicados para esta tarea en castellano resultan de gran interés, pues guarda una relación importante con el presente trabajo.

Para la tarea de clasificación, la mayoría de los equipos utilizaron Máquinas de Vectores de Soporte (SVM, *Suppor Vector Machines*) y métodos combinados de aprendizaje (EoC, *Ensemble of Classifiers*). Las técnicas basadas en SVMs fueron utilizadas por (Canós, 2018; Endang Wahyu Pamungkas y Patti, 2018; Nina-Alcocer, 2018) mientras que los equipos (Resham Ahluwalia y Cock, 2018; Shushkevich y Cardiff, 2018; Simona Frenda y y Gomez, 2018; Han Liu y Cocea, 2018) aplicaron técnicas EoC.

Las soluciones aportadas por (Canós, 2018; Endang Wahyu Pamungkas y Patti, 2018) obtuvieron la mejor tasa de acierto para la detección de la misoginia en castellano. El modelo propuesto por (Canós, 2018) utiliza *features* basadas en la vectorización de cada tweet, utilizando la medida tf-idf (*term frequency - Inverse document frequency*). Posteriormente, se emplea un modelo SVM con núcleo lineal para la etapa de clasificación. Esta solución tan sencilla alcanza los mejores resultados para *tweets* en castellano,

pero empeora considerablemente para *tweets* en inglés.

Una idea interesante, explorada en (Endang Wahyu Pamungkas y Patti, 2018), es el uso de un léxico auxiliar que contenga palabras que se encuentren con frecuencia en textos sexistas. Este léxico fue desarrollado en un trabajo italiano (De Mauro, 2016). En dicho estudio, se utiliza como clasificador un modelo basado en SVM con núcleo lineal para el castellano y núcleo radial para el inglés. En este caso, se alcanza la máxima tasa de acierto en inglés y en español.

(Goenaga y Perez, 2018) fue uno de los pocos trabajos donde se exploraron soluciones basadas en redes neuronales. En este trabajo se utilizan redes neuronales recurrentes (*Recurrent Neural Network, RNN*) como sistema de clasificación realizando previamente un preprocesado basado en *word embeddings* que permite codificar las palabras o términos mediante números reales. Pese a que este tipo de técnicas han mostrado su eficacia en tareas relacionadas con el procesado de texto (Devlin, 2019), esta solución queda lejos de los mejores resultados alcanzados en la competición.

2.3.1. Corpus disponibles

A continuación se citan algunos corpus que pueden ser utilizados para la detección de lenguaje del odio en textos:

- IberEval 2018 Automatic Misogyny Identification (E. Fersini y Anzovino, 2018b): Se trata de un corpus etiquetado que contiene campos que denotan si el texto contenido en un tweet tiene un componente sexista. Fue recogido entre el 20-07-2018 y 30-11-2017 donde se recogieron 83 millones de tweets en inglés y 72 millones en castellano. Para el proceso de etiquetado se utilizadon dos pasos: en el primero dos anotadores etiquetaban el conjunto y en el segundo se utilizó una plataforma de crowdsourcing. Finalmente, se etiquetaron 3521 tweets en inglés y 3307 en español para la fase de entrenamiento. En cuanto al conjunto de test, se compartieron 831 tweets en español y 726 en inglés.
- Corpus etiquetado (Waseem, 2016b): Está compuesto por *tweets* etiquetados para mensajes sexistas, racistas o sin contenido ofensivo. Se trata de un conjunto de datos recolectado durante dos meses y compuesto por 136.052 de los cuales se etiquetaron 16.614. De los tweets

etiquetados, 3.383 contienen mensajes machistas, 1972 contienen expresiones racistas y 11.559 mensajes libres de contenido ofensivo.

Capítulo 3

Herramientas utilizadas

En este capítulo se describen en profundidad las distintas herramientas evaluadas para la creación del sistema propuesto. Además, se exponen los motivos por los que se han elegido frente a otras alternativas disponibles.

3.1. Crawler

3.1.1. Amazon Web Services

AWS es una creciente unidad dentro la compañía Amazon.com que ofrece una importante variedad de soluciones de Cloud Computing tanto PYMES como a grandes empresas a través de su infraestructura interna, siendo la marca más utilizada actualmente en el mercado de la nube con casi un 40 % de cuota de mercado (Research, 2018). Amazon ofrece unos servicios en la nube pública mediante una tarificación de precios en función del tiempo de uso, anchos de banda consumidos, etc. Por lo tanto, su gran ventaja competitiva es ofrecer unos recursos de infraestructura y plataforma poco asumibles a la mayoría de empresas para el periodo que se requiera.

Los clientes de AWS tan sólo deben pagar lo que usen del servicio, de esta manera, obtener unos potentes servidores con una plataforma determinada, un espacio de almacenamiento o una gran base de datos supone la adquisición de un hardware que no se aproveche todo el tiempo, que tan sólo interese para un periodo determinado y satisfacer una necesidad puntual, prescindiendo de importantes inversiones en infraestructura. Orientado a empresas, se adapta con total flexibilidad y escalabilidad a las necesidades de cloud que tenga el cliente, mediante un acuerdo de nivel de servicio,

se especifica el nivel de compromiso del servicio, disponibilidad y ofrece un punto de confianza que otros proveedores de nube pública no proporcionan, dato que le da ventaja frente a sus competidores.

Dado que ha sido pionero en el sector y posee una gran cantidad de desarrolladores que trabajan para mejorar el servicio, desde su publicación en 2006, ha sido líder en el sector por delante de Google App Engine, Azure de Microsoft, Alibaba, etc (Research, 2018). Siempre ha ido un paso por delante y le ha permitido innovar en el sector y ofrecer unos precios muy competitivos, soluciones para todos los gustos e importantes acuerdos con Microsoft, IBM y HP como estrategias de marketing para ofrecer software y plataformas propietarias (además de software libre que fue lo primero que se ofrecía con plataformas Linux) en sus imágenes de máquinas virtuales. En la siguiente figura se puede ver un resumen de los servicios de AWS:

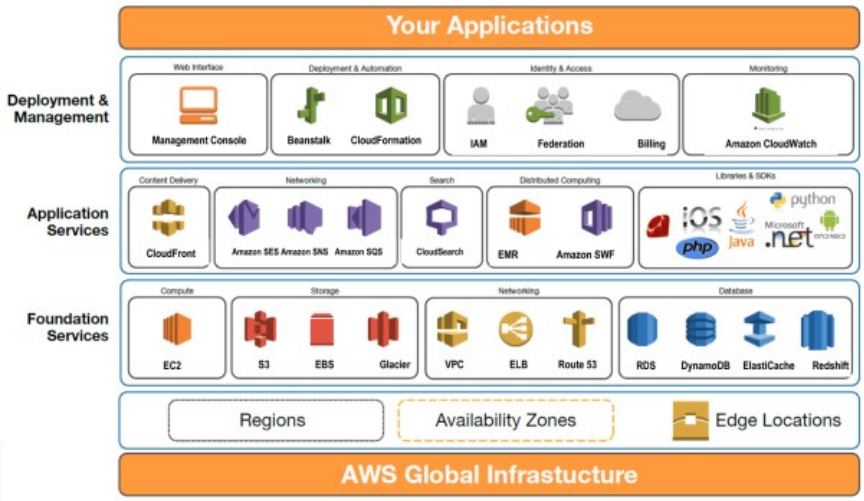

Figura 3.1: Resumen de servicios AWS

Los diferentes servicios de AWS se incrementan con el paso del tiempo, siendo EC2, S3 y Lambda los que más peso tienen en el presente proyecto:

- Amazon Elastic Compute Cloud (EC2): proporciona servidores virtuales escalables. Proporciona las capacidades de Cloud Computing a sus clientes de manera que permite una configuración y administración de las capacidades de máquinas virtuales que se solicitan a la nube, pudiendo pagar tan sólo el tiempo de computación. Actualmente existen numerosos tipos de instancias con características hardware distintas según los requisitos del usuario.

- Amazon Simple Storage Service (S3): proporciona un Web Service basado en el almacenamiento online para aplicaciones. Este almacenamiento en Internet proporciona una simple interfaz web, como su nombre indica, que puede ser usada para almacenar grandes cantidades de datos en cualquier momento desde cualquier sitio, dando acceso confiable y seguro con SLA, altamente escalable, rápido y barato en la infraestructura de Amazon. Físicamente, los datos están distribuidos por los Data Center de Amazon, pero es algo que permanece ajeno al cliente y de lo que no debe preocuparse (escalabilidad). Su integración con EC2 es esencial para que las imágenes de máquinas virtuales puedan trabajar con datos y objetos almacenados en S3 y tener un espacio donde los desarrolladores puedan trabajar cómodamente incluso poder solicitar más espacio temporal para las máquinas o disponer de varios ?buckets? donde compartir datos entre instancias.
- AWS Lambda: se trata de un servicio de computación sin servidor. Este servicio permite ejecutar código sin aprovisionar ni administrar servidores, pagando únicamente por el tiempo de cómputo que se consuma. De este modo, este servicio permite que se AWS quien se encargue de la administración de las máquinas y el usuario únicamente trabaje en el código que se ejecuta.

La segunda plataforma de cloud computing más importante a nivel mundial es Azure, propiedad de la empresa Microsoft. En este caso, no se ha elegido Microsoft Azure porque ya se contaba con un conocimiento previo en el uso de los servicios de AWS. Además, AWS cuenta con servicios de computación serverless, como AWS Lambda, muy útiles para la realización del crawler.

3.1.2. Twitter API y rtweet

Twitter proporciona múltiples APIs para facilitar el acceso a los datos de su plataforma. De todas ellas, la necesaria para crear el corpus objetivo sería el API REST de Twitter (Twitter, 2018a). En concreto, es necesario utilizar la funcionalidad Tweet Search que permite realizar búsquedas de los tweets generados en la plataforma según distintos parámetros de búsqueda.

Dentro del API existen 3 tipos de cuenta según la cantidad de información disponible para consulta: Standard Search, Premium Search y Enter-

prise Search. De todas ellas, solamente la primera es gratuita por lo que será la utilizada durante el proceso de generación del corpus. Es importante señalar que este tipo de búsqueda presenta algunas limitaciones. Las dos más importantes serían la existencia de una ventana temporal de consulta limitada a 7 días anteriores y, por otra parte, la limitación de descarga de tweets a 18.000 cada 15 minutos.

Para recopilar la información de Twitter, se ha utilizado la herramienta rtweet (Kearney, 2018). Se trata de un cliente del lenguaje de programación R para acceder al API de Twitter. Este paquete facilita mucho las tareas habituales como la búsqueda de tweets.

Existen varias alternativas a rtweet como tweetpy (Roesslein, 2009) para el lenguaje de programación Python o twitteR. Se ha optado por rtweet porque ambas alternativas están más desactualizadas y son proyectos mucho más inactivos.

3.2. Preprocesado y tokenización

En la etapa inicial para la clasificación de textos, se aplican distintas técnicas que permitan Extraer los atributos o *features* necesarias para realizar una representación fiel del texto y que permitan la utilización de un algoritmo de clasificación. Existen multitud de procedimientos aplicables en esta etapa como la tokenización, el reconocimiento de entidades nombradas, el etiquetado sintáctico y morfológico:

- Tokenización: Permite separar cada palabra o símbolo del corpus en unidades independientes (como palabras) que pueden ser almacenadas para su posterior procesado.
- Reconocimiento de entidades nombradas: Tarea que permite clasificar fragmentos de texto en categorías predefinidas, como personas, organizaciones, lugares, expresiones de tiempo y cantidades.
- Etiquetado sintáctico: Proceso en el que se busca sobre el espacio de todas las posibles combinaciones de las reglas gramaticales definidas para encontrar la estructura de una oración.
- Etiquetado morfológico: En este proceso se le asigna a cada palabra su función dentro del corpus utilizado. Normalmente, se utilizan 8 etiquetas distintas en la mayoría de los idiomas utilizados en Europa: nom-

bre, verbo, pronombre, preposición, adverbio, conjunción, partícula y articulo.

Para aplicar este tipo de técnicas, existen gran cantidad de proyectos o librerías de computación disponibles. Algunas de las más utilizadas son las siguientes:

- Freeling (Carreras et al., 2004): Es una librería que soporta el lenguaje español y se utiliza en (Simona Frenda y y Gomez, 2018). Pese a que tiene mucha de las características que se necesitan, tiene una menor comunidad y está menos extendido que algunas del resto de las herramientas.
- Stanford Parser (Group, 2015): Se trata de una librería desarrollada por el grupo de trabajo de NLP de la universidad de Stanford.
- TweetNLP (Ovoputi, 2013): Librería desarrollada específicamente para le procesado de tweets. Su uso no está muy extendido.
- Spacy (Honnibal y Montani, 2017): Se utiliza en (Waseem, 2016a) y permite aplicar las técnicas de procesado de un modo eficiente.
- NLTK (NLTK, 2018): Se trata de la librería más extendida para el preprocesamiento, se utiliza en (Zimmerman, 2018; Thomas Davidson, 2017; Simona Frenda y y Gomez, 2018).

De todas las herramientas listadas, se ha optado por la librería NLTK. Se trata de una librería muy extendida que cuenta con una gran comunidad y permite un desarrollo muy ágil. Algunas librerías como Freeling o Stanford Parser requieren varias dependencias para poder ser utilizadas.

La mejor alternativa a NLTK considerada sería Spacy. Su uso está aumentando y su funcionamiento es muy similar ya que ambas están desarrolladas en Python. Se ha optado por NLTK porque a día de hoy sigue siendo más utilizada.

3.2.1. NLTK: Natural Language Toolkit

NLTK (NLTK, 2018) es una librería que define una infraestructura en la que crear programas para el procesado del lenguaje natural (NLP, "*Natural language processing*") en "*Python*". Provee la estructura básica para representar datos relevantes para el procesado del lenguaje natural, interfaces

para realizar tareas como el etiquetado del discurso (POS, "part-of-speech tagging"), etiquetado sintáctico y clasificación de texto.

Esta librería fue desarrollada originalmente en el año 2001 como parte de un curso de lingüística computacional en la Universidad de Pennsylvania. Desde entonces, ha sido desarrollada y mejorada por distintos contribuidores al tratarse de un proyecto libre. Actualmente, NLTK es utilizado en gran cantidad de investigaciones y supone un estándar muy importante para realizar tareas relacionadas con NLP. Está compuesto por una cantidad importante de módulos que pueden ser invocados desde un programa escrito en Python. En la siguiente figura se recogen los más importantes (Steven Bird y Loper, 2009):

Language processing task	NLTK modules	Functionality
Accessing corpora	nltk.corpus	standardized interfaces to corpora and lexicons
String processing	nltk.tokenize, nltk.stem	tokenizers, sentence tokenizers, stemmers
Collocation discovery	nltk.collocations	t-test, chi-squared, point-wise mutual information
Part-of-speech tagging	nltk.tag	n-gram, backoff, Brill, HMM, TnT
Classification	nltk.classify, nltk.cluster	decision tree, maximum entropy, naïve Bayes, EM, k-means
Chunking regular	nltk.chunk	expression, n-gram, namedentity
Parsing	nltk.parse	chart, feature-based, unification, probabilistic, dependency
Semantic interpretation	nltk.sem, nltk.inference	lambda calculus, first-order logic, model checking
Evaluation metrics	nltk.metrics	precision, recall, agreement coefficients
Probability and estimation	nltk.probability	frequency distributions, smoothed probability distributions
Applications	nltk.app, nltk.chat	graphical concordancer, parsers, WordNet browser, chatbots
Linguistic fieldwork	nltk.toolbox	manipulate data in SIL Toolbox format

Figura 3.2: Módulos NLTK

3.3. Scikit-learn

Scikit-learn (F. Pedregosa, 2011) es un proyecto que provee una librería de aprendizaje de máquina para el entorno de programación "Python". El objetivo principal de esta librería es establecer un conjunto de herramientas dentro de un entorno de programación que sea accesible a usuarios no ex-

pertos. Esta librería incluye algoritmos clásicos de aprendizaje de máquina, herramientas para la selección, evaluación de modelos y preprocesado. Todos los objetos dentro de la librería comparten una API básica compuesta por 3 interfaces complementarias: "estimators" que permiten construir y ajustar modelos, "predictors", para realizar predicciones, y "transformers", que permiten realizar conversiones a los datos.

La mayor parte de modelos de aprendizaje supervisados o funciones auxiliares relacionadas con el procesado de datos utilizados en el presente trabajo están implementados o han sido desarrollados con ayuda de funciones disponibles en la librería scikit-learn.

3.3.1. "Estimators"

La interfaz "estimator" define objetos y provee de un método "fit" para ajustar un modelo a los datos de entrenamiento. Todos los algoritmos supervisados y no supervisados implementados en la librería son tratados como objetos implementando esta interfaz. Otro tipo de tareas relacionadas con el aprendizaje de máquina como la selección de atributos o métodos para la reducción de la dimensionalidad también utilizan el interfaz "estimator".

La inicialización de un "estimator" y el ajuste de un modelo a los datos de entrenamiento están diferenciados en la librería. Un "estimator" se puede inicializar un con conjunto de parámetros de entrada (por ejemplo, el parámetro C para SVM) y, posteriormente, se utiliza el método "fit" para realizar el proceso de ajuste a los datos de entrenamiento. En el siguiente código se ilustra esta funcionalidad:

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=250)
rf.fit(X_train, y_train)
```

En el código anterior, primero se inicializa un "estimator" estableciendo el argumento "n_estimators". Tras esto, se realiza una llamada al método "fit" para realizar el ajuste utilizando los datos de entrenamiento.

3.3.2. "Predictors"

La interfaz "predictor" extiende la funcionalidad del "estimator" añadiendo el método "predict". Este método devuelve un vector de predicciones

tomando como entrada una matriz con los datos de testeo. Ampliando el ejemplo anterior:

```
y_pred = rf.predict(X_test)
```

3.3.3. "Transformers"

Antes de aplicar un método de clasificación supervisada, suele ser habitual realizar filtrados o modificaciones en los datos, para ello, "scikit-learn" implementa la interfaz "transformer".

Esta interfaz define el método "transform" que toma como entrada una matriz de datos y devuelve como salida una versión transformada de estos datos. Algunas de las transformaciones más comunes pueden ser la selección de atributos, preprocesado o métodos de reducción de dimensionalidad. Un ejemplo de preprocesado podría ser el estandarizado de un conjunto de datos:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
```

3.3.4. "Pipelines y selección de modelos"

"Scikit-learn" permite componer nuevos "estimators" utilizando otros, lo que permite crear flujos de trabajo completos en un único objeto. Este tipo de tarea se puede realizar de dos modos: mediante "Pipeline" utilizando un modelo secuencial o mediante "FeatureUnion".

Los objetos "Pipeline" encadenan "estimators" en un único objeto. Esto permite crear flujos de trabajo siguiendo un número fijo de pasos, por ejemplo: extracción de atributos, reducción de dimensionalidad, ajuste de un modelo y realización de de predicciones.

Los objetos "FeatureUnion" combinan multiples "transformers" en uno único y concatena los resultados. De este modo, este tipo de objeto es capaz de realizar transformaciones distintas sobre el mismo conjunto de datos o sobre una parte del mismo.

Ambos objetos pueden ser combinados para crear flujos de trabajo más complejos. Por ejemplo, en el siguiente código se combinan dos "Pipeline" utilizando "FeatureUnion" y se añade un último paso "clf" que añade un clasificador.

```
Pipeline([('feature-union',
        FeatureUnion([('text-features',
        text_pipeline),
        ('other-features', preprocess_pipeline)])),
        ('clf', LogisticRegression(penalty = ``L2'')
        ])
```

En "scikit-learn" es posible realizar selección de modelos mediante el meta-estimador "GridSearchCV". Este método toma como entrada un "estimator" cuyos parámetros de entrada deben de ser optimizados. Para ello, se definen todos los valores que se deben de tener en cuenta en el proceso para cada parámetro de entrada.

Capítulo 4

MeTwo dataset (Machismo and Sexism Twitter Identification dataset)

Este capítulo describe la metodología utilizada para la creación de un corpus en castellano que permita entrenar un sistema de detección del lenguaje machista. De este modo, se ha generado un corpus utilizando como semillas un conjunto de términos en castellano que, por lo general, pueden llevar a actitudes machistas. Idealmente, el conjunto generado contendrá suficientes ejemplos de tweets machistas que serán utilizados como entrada del sistema de clasificación.

Al comienzo de esta sección, se analiza el problema y se estudian las expresiones y términos más empleados en textos con comportamiento machista. Se realiza una búsqueda preliminar para observar si con los términos estudiados se encuentran ejemplos de textos machistas y, finalmente, se define el método para extraer la información de Twitter y evaluar las características principales del corpus obtenido.

4.1. Machismo en Twitter

En este apartado, se van a definir todas las características que deben de presentar los textos seleccionados para considerarse machistas y la lista de términos o expresiones seleccionados para generar el corpus. Como primer paso, se han estudiado diversas referencias para recopilar aquellas expre-

siones más comunes que pueden conllevar a comportamientos machistas o sexistas. Posteriormente, se han identificado las expresiones más representativas y se han realizado diversas búsquedas para evaluar la cantidad de contenido machista que contienen dichas expresiones en Twitter. Finalmente, se han obtenido de la red social Twitter los mensajes que contenían estas expresiones utilizando su API. Durante este estudio solo se han considerado expresiones en castellano.

Las expresiones o términos que se buscan son aquellas que, de un modo u otro, minusvaloran el papel de las mujeres en nuestra sociedad, incentivan el abuso o acoso hacia las mujeres o no les permitan expresarse libremente. Existen gran cantidad de expresiones y términos que se utilizan a diario de modo consciente o no, que minimizan el papel de la mujer en la sociedad. En (Twi, 2017) la periodista Ana Isabel Bernal-Triviño propone, a través de un hilo en su cuenta de Twitter, recopilar frases de violencia machista que las mujeres hayan escuchado o recibido en algún momento. Estas expresiones tienen mucho valor para el análisis pues un gran número de mujeres aporta su experiencia personal frente al machismo. En muchos tweets se repiten términos como "niñata" o "a fregar" que se utilizan para referirse a las mujeres de forma despectiva.

Existen refranes, dichos populares y tópicos que se utilizan diariamente y que refuerzan la idea de que las mujeres son inferiores al hombre en una tarea determinada. Por ejemplo, la expresión "Mujer al volante, peligro constante" o "¡Mujer tenía que ser!" minusvalora la habilidad de las mujeres para realizar una tarea específica solo por ser mujeres. Otras muchas expresiones machistas se recopilaron en un trabajo realizado en un instituto de Albacete (Bernal, 2017).

Pese a que, como se ha presentado, existen gran cantidad de expresiones que conllevan actitudes machistas, ha sido necesario recoger también ciertos términos más generales que permitan establecer una buena relación entre tweets con contenido machista y aquellos que no presentan este tipo de lenguaje. Esto será muy importante a la hora de entrenar el sistema automático de clasificación. Así, partiendo de los trabajos anteriores, se han seleccionado las siguientes expresiones: "feminazi", '"a la cocina", '"a fregar", "marimacho", "ninata", '"mujer tenias que ser", '"las feministas","en tus dias", "zorra",'"como una mujer", '"como una nina", '"pareces una fulana", '"pareces una puta",'"no ha probado un hombre",'"loca del", '"obsesiona-

da con el machismo", '"para ser mujer",'"para ser chica",'"hombre que te aguante",'"acabaras sola", "mojigata", '"mucho feminismo pero",'"mujer al volante", '"las mujeres no deberian", '"A las mujeres hay que", '"odio a las mujeres",'"las mujeres de hoy en dia", "nenaza", "lagartona". En total, 29 expresiones que se utilizarán para recopilar *tweets* como punto de partida para la generación del dataset. La tabla 1 recopila dichos términos.

El término **"feminazi"** es una forma de relacionar, de forma claramente despectiva, al feminismo con el nazismo. De este modo, es una palabra utilizada ampliamente en redes sociales y foros de toda España que conllevan actitudes machistas. En Twitter se encuentran mensajes como:

"*Uy habló de inventos la feminazi*"

"*Pero que violento!!! De cinco billetes, solo en uno aparece una mujer, esto es inaceptable!!! #feminazi*"

Sin embargo, existen otros ejemplos que se utiliza la expresión sin conllevar actitudes machistas:

"*Si quieres pensar que el origen del hashtag deviene de la búsqueda del rigor histórico, a tope. Estás en tu derecho, igual que yo en un texto de opinión de valorarlo totalmente al contrario. ¿No te convencen mis argumentos? Intentemos debatir. ¿Usas "feminazi¿ Te quedas solo*"

Los términos **"A la cocina"** y **"A fregar"** se utilizan de modo despectivo para denigrar a la mujer o restar importancia a sus argumentos. Como en el caso del término anterior, se ha comprobado que existe una relación aceptable entre mensajes machistas y aquellos que no conllevan estas actitudes. Algunos ejemplos de contenido machista:

"*Hay que anunciar por megafonia que te marches a tu casa a fregar que estas haciendo el ridículo .tu y las borregos*"

"*De los relatos corazon... anda a la cocina a hacee algo productivo antes de seguir diciendo pavadas*"

Ejemplos que no expresan actitudes machistas:

"*Me dio sed, pero me da miedo ir a la cocina*"

"*Me molesta que a esta hora me dé hambre porque me da miedo ir a la cocina por algo para comer*"

Las palabras **"Marimacho"** y **"Niñata"**, dependiendo del contexto, se utilizan como calificativos despectivos a la mujer. Ejemplos despectivos serían los tweets encontrados:

"*@IrantzuVarela hace unos sketches poniéndose unas barbas postizas des-*

Número de término	**Texto**
1	“feminazi”
2	“a la cocina”
3	“a fregar”
4	“marimacho”
5	“ninata”
6	“mujer tenias que ser”
7	“las feministas”
8	“en tus dias”
9	“zorra”
10	“como una mujer”
11	“como una nina”
12	“pareces una fulana”
13	“pareces una puta”
14	“no ha probado un hombre”
15	“loca del”
16	“obsesionada con el machismo”
17	“para ser mujer”
18	“para ser chica”
19	“hombre que te aguante”
20	“acabaras sola”
21	“mojigata”
22	“mucho feminismo pero”
23	“mujer al volante”
24	“las mujeres no deberian”
25	“A las mujeres hay que”
26	“odio a las mujeres”
27	“las mujeres de hoy en dia”
28	“nenaza”
29	“lagartona”

Tabla 4.1: Términos machistas elegidos para la creación del corpus

peinadas. Es lo que antes se llamaba un marimacho. El igualitarismo ha hecho mucho daño. Uno tiene que mandar y otro obedecer acríticamente."

"*@carmenro_ Mal follada y retrasada. Das pena niñata. Tienes un concepto muy equivocado de los andaluces y Andalucía. Gente como tú son las qué no queremos en nuestra comunidad. Aquí hay gente muy trabajadora y humilde. Envidia es lo que tienes.*"

Es importante aclarar, que en el proceso de etiquetado, el término **"Marimacho"** y **"niñata"** no se ha considerado machista si no se encontraba en un contexto muy claro. De forma general, **"marimacho"** se utiliza para indicar que una mujer presenta rasgos masculinos algo que, aún pudiendo ser un tipo de abuso, no se ha considerado machista. Por otra parte, **"niñata"** se utiliza para indicar que una mujer es inmadura, algo que de forma general no sería machista.

Las expresiones **"Mujer tenías que ser"** y **"Las Feministas"** se utilizan para descalificar a las mujeres por el hecho de pertenecer a un colectivo. Ejemplos encontrados en Twitter son los siguientes:

"*Mujer tenias que ser para dar una respuesta tan pobre*"

"*@NiicoMilan Coincido eso es el verdadero machismo que tiene un respeto absoluto por la mujer más no lo que quieren hacer las feministas pasar por machismo.Nadie puede lastimar a una mujer*"

La expresión **"en tus días"** se utiliza como descalificativo para referirse a las mujeres por tener el periodo:

"*@PriscilaTrs_ Bb se delicia de culo no te dejaría descansar ni cuando estés en tus días mamasita rica me pasaría dándole placer mañana tarde y noche a ese delicioso culo hasta hacerte gritar de placer*"

"*Seguro era porque andabas en tus días. xd*"

El término **"zorra"** se utiliza continuamente en Twitter con actitudes machistas:

"*zorra se puede saber pork no me sigues?*"

"*Dedicado a las estúpidas feminazis que dicen que a las mujeres hay que creerles todo solo por ser mujeres.SIN PRUEBAS NO HAY DELITO.https://t.co/XNx3fk3m5K*"

Otra expresión que minusvalora a las mujeres considerada es **"como una mujer"** para comparar una acción que se realiza peor únicamente por ser mujer:

"*Un hombre sin barba es como una mujer sin nalgas*"

"*#UnMadrazoPara Cesar Gaviria, que habla como una mujer en proceso de parto.*"

Otra expresión similar a la anterior sería **"como una niña"**:

"*No te vas a ir de aquí sin decirme qué sitio es ese. Y más te vale responder y no comportarte como una niña pequeña, niña pequeña.*"

El término **"nenaza"** se utiliza también como descalificativo en Twitter:

"*A mi me ha pasado igual. Los lobos vestidos de corderos que lo mismo quieren matar fachas que lloran como una nenaza no puedo con ellos*"

"*@marianorajoy ha perdido la Moncloa como una nenaza porque no ha sabido defender como hombre y estadista a la Nación. Se va como los cobardes. Por la puerta de atrás. El peor presidente de España* "

La expresión **"pareces una fulana"**'se utiliza como descalificativo por la apariencia física de las mujeres:

"*Te acabo de ver en el Telenoticiasde TV3. Maquillate mejor, pareces una fulana.*"

"*¡Lávate esa cara que pareces una fulana!*"

Otra expresión muy similar a la anterior sería **"pareces una puta"**:

"*Deja de gritar hermano, pareces una puta*"

"*@emuyshondt @alcaldia_ss @SSesTuyo podes administrar o no, porque te vibis quejan maje, mas bien pareces una puta*"

Otra expresión machista utilizada en Twitter es **"no ha probado un hombre"**. En este caso, se sexualiza a la mujer y se sugiere que tiene un algún problema o defecto por no tener relaciones con un hombre.

"*que se pueda esperar de una mamerta @ktikariza que no lee no estudia va a la provincia tomar cerveza una millenian que no ha probado un hombre, el hambre o la guerra, estas juventud esta empecinada a decir y tener too regalado a costa de si* "

"*pobre mujer, no ha probado un hombre en su vida.*"

La expresión **"loca del"** se utiliza de forma despectiva hacia las mujeres:

"*@Madrekoraje @PhilAMellows @CasosAislados Todo los hombres tenemos que tener miedo a cruzarnos con una loca del coño que se invente lo que le vengan en gana y por el mero hecho de tener papo haya que creerla.*"

"*Odio a esta clase de feministas, loca del coño.*"

La frase **"obsesionada con el machismo"** se utiliza para restar importancia a la denuncia del machismo que realizan las mujeres:

"*Típica obsesionada con el machismo, el patriarcado y franco. De ahí no*

la sacas. No evoluciona"

"*Feliz día de la gente tontaca obsesionada con el machismo*"

Otra expresión que minusvalora las habilidades de la mujer sería **"para ser mujer"**, pues implícitamente da por hecho que una mujer realiza cierta acción peor que un hombre:

"*@damita2808 @berege7 @Mariagtriana Y los ojos? Uff demasiado dureza en la mirada para ser chica...no?*"

"*Para ser mujer habla bastante decente ...*"

Otra expresión similar a la anterior sería **"para ser chica"**:

"*@GreatBlastG5 Mi madre mide bien para ser chica, y mi padre es de estatura media, lo de los yogures te lo decía a ti porque se que me mientes prro*"

"*Sos un poco masculina como para ser chica, ¿sabías?*"

Otra expresión machista utilizada en Twitter es **"hombre que te aguante"**:

"*Chingón el hombre que te aguante hasta en tus días. "*

"*Damaris, cásate con tu "marido (Ana)"porque no vas a encontrar un hombre que te aguante.*"

Otra expresión machista utilizada en Twitter es **"acabaras sola"**. En este caso, expresa que una mujer sin un hombre a su lado es una mujer ïncompleta":

"*#SomosLaAudiencia21F SOFIA: ¡ eres patética, no, lo siguiente ¡ Deja a ALEJANDRO en paz de una vez. ERES TAN PREPOTENTE que quieres ver a todos los hombres arrastrándose por ti. ACABARÁS SOLA..... TIEMPO AL TIEMPO....*"

"*Nunca encontrarás un hombre que te sorpote. Acabarás sola, vieja, con gatos y escuchando techno*"

Otro insulto muy recurrente es **"mojigata"**:

"*#EnLaPedaNuncaFalta la mojigata persinada que con 3 tequilas encima es más fácil que la tabla del 1. "*

"*@Mony_loreH Y CON ESA CARITA ME MOJIGATA QUE TIENE.....SON LAS PEORES.....*"

La expresión **"mucho feminismo pero"** se utiliza para minusvalorar el papel del feminismo en la sociedad:

"*Pues menuda hija de mierda si tienes a tu padre cargando con TODAS las tareas del hogar. Mucho feminismo, pero de compartir las labores no te*

enseñó nada."

"*Mucho feminismo pero Christian Gray se las hace mojar qué tipo hdp este pibe jajaja*"

La expresión **"mujer al volante"** se utiliza habitualmente para minusvalorar la capacidad de las mujeres para realizar una tarea, en este caso, conducir:

"*por la prudencia con la que tomo la bajada creo que se trata de una mujer al volante* "

"*No hay nada más peligroso que una mujer al volante con prisa . Nada, ni MALO De presidente es tan peligroso*"

La expresión **"las mujeres no deberían"** sugiere que las mujeres no deben de realizar alguna tarea o acción por el hecho de ser mujer:

"*Las mujeres no deberían de tener ni voz ni voto, no están preparadas para mantener una familia menos un gobierno , son unas mantenidas*"

"*Por eso es que las mujeres no deberían tener derecho al voto*"

La frase **"A las mujeres hay que"** se utiliza habitualmente para discriminarlas o tratarlas de un modo distinto por su género:

"*Como dice mi papá: "a las mujeres hay que quererlas, no entenderlas""*"

"*Hoy en día a las mujeres hay que tenerle un cuidado cabron. Acuérdate que estamos en la era del chismin, la era mierdosa. Te da un abrazo y después dice que le rozaste la teta. Wao! Pero si nos abrazamos! Obvio que las tetas feas esas van a rozar cabrona!*"

Otra expresión que suele conllevar actitudes machistas el **"odio a las mujeres"**:

"*Odio a las mujeres pero me atraen sexualmente*"

"*A lo que hemos llegado, lo unico que van a conseguir las feministas es el odio a las mujeres*"

La expresión **"las mujeres de hoy en dia"** suele expresar un sentimiento negativo hacia las mujeres solo por el hecho de ser mujer:

"*Las mujeres de hoy en día son tan doble moral; no puedes decirle a una chica que trae el botón desabrochado de la blusa, por que se emperra y te tacha de pervertido, ah pero si no le dices te va peor. Tampoco es que sea un santo pero que no mamen. #LaHubieraDejadoEnseñarTeta* "

"*Se enojan las mujeres de hoy en día que uno les diga que parecen hombres, más insensibles que una piedra y más inútiles en las actividades de a diario, puta no sean tan maletas prepárense, por lo menos yo hice mi obra*

hoy ya se defiende en un montón de áreas de nada..."

Por último, el insulto **"lagartona"** se utiliza coloquialmente para sugerir que una mujer tiene relaciones sexuales a cambio de dinero:

"*@pescadosgori ¿Solo le quieres por el dinero? ¡Ay ay ay lagartona! https://t.co/7HE8PrXdDk*"

"*No quiero depender de nadie. No soy una lagartona ni la zorrita de nadie, si no un hombre*"

4.2. Generación del corpus, "*Crawler*"

En este apartado, se especifica el proceso utilizado para extraer la información deseada de Twitter. En la fase de creación del corpus se ha utilizado el API de Twitter a través del paquete *rtweet* disponible para el lenguaje de programación R.

Las especificaciones iniciales para el desarrollo del *Crawler* implican la recolección diaria de 100 tweets para cada término contenido en la tabla 4.1, lo que, idealmente, haría un total de 2900 tweets diarios. De este modo, no se superaría el límite diario del API de Twitter y se recopilaría bastante información cada día. Al llegar a 15000 tweets para un término, se considera que hay suficiente información y se dejará de buscar los tweets que lo contengan.

Al término del proceso de *crawling*, se eligirán aleatoriamente 150 tweets para cada término. Por tanto, será un requisito obtener, al menos, 150 tweets para cada término para poder ser considerado.

Para esta tarea, el proceso de *crawling* se ha extendido a lo largo de las fechas 1/07/2018-31/12/2018, haciendo un total de 6 meses de información. Durante este periodo, el *Crawler* recolectó un total de 181792 tweets para todos los términos. Para cada *tweet*, el API de Twitter permite acceder a 42 atributos distintos:

- status_id: identificador único del tweet.
- created_at: fecha de creación del tweet.
- user_id: identificador único de usuario.
- screen_name: alias que el usuario utiliza para identificarse.
- text: mensaje del tweet.
- source: dispositivo/cliente utilizado para publicar el mensaje.

- reply_to_status_id: si es una respuesta, indica el id del tweet original.
- reply_to_user_id: si es una respuesta, indica el id del usuario original.
- reply_to_screen_name: si es una respuesta, indica el alias del usuario original.
- is_quote: indica si el tweet es citado.
- is_retweet: indica si el tweet es un retweet.
- favorite_count: conteo aproximado del número de favoritos.
- retweet_count: conteo aproximado del número de tweets.
- hashtags: hastags utilizados en el tweet.
- symbols: símbolos contenidos en el mensaje.
- urls_url: URLs contenidas en el texto del tweet.
- urls_t.co: URLs en formato acortado.
- urls_expanded_url: URLs en formato expandido.
- media_url: URLs de los elementos subidos con el tweet.
- media_t.co: URLs acortadas de los elementos subidos con el tweet.
- media_expanded_url: URLs expandidas de los elementos subidos con el tweet.
- media_type: tipo de elemento subido junto al tweet (por ejemplo, foto).
- ext_media_url: URLs del elemento externo adjunto al tweet.
- ext_media_t.co: URLs acortada del elemento externo adjunto al tweet.
- ext_media_expanded_url: URLs expandidas del elemento externo adjunto al tweet.
- ext_media_type: tipo del elemento externo adjunto al tweet.
- mentions_user_id: identificadores de otros usuarios nombrados en el tweet.
- mentions_screen_name: alias de otros usuarios nombrados en el tweet.
- lang: idioma del tweet.
- quoted_status_id: identificador del tweet citado.
- quoted_text: texto del tweet citado.

- retweet_status_id: si es un retweet, indica el identificador del tweet original.
- retweet_text: texto del retweet original.
- place_url: URL que representa la localidad y aporta información adicional.
- place_name: nombre de la localidad.
- place_full_name: nombre de la localidad con información añadida (por ejemplo, provincia).
- place_type: tipo de localidad (por ejemplo, ciudad).
- country: país.
- country_code: abreviatura del país.
- geo_coords: longitud y latitud del tweet.
- coords_coords: longitud y latitud del tweet en distinto formato.
- bbox_coords: un cuadro delimitador de coordenadas que encierra este lugar
- latitud: latitud del tweet.
- longitud: longitud del tweet.

4.2.1. Resultados de la creación del corpus

A continuación, se presentan las características y estructura principal del corpus generado.

La tabla 4.2 muestra el número de tweets recopilado por término. Como se puede observar, se ha recopilado información para todos los términos pero existen 4 de ellos para los que se han recopilado menos de 150 mensajes y que serán descartados por no disponer de suficiente información, se trata de los términos "acabarás sola", "hombre que te aguante", "obsesionada con el machismo", "pareces una fulana" y "no ha probado un hombre". Además, es importante remarcar que la cantidad de información varía notablemente según los términos, a partir del término 11 la cantidad de información por término se reduce en más de la mitad.

Como se ha comentado, el *crawler* recolectó información desde el 01/07/2018 hasta 31/12/2018. En la figura 4.1 se puede observar el número de tweets

Término	**N**
como una mujer	15094
feminazi	15093
a la cocina	15087
zorra	15086
loca del	15084
como una nina	15080
las feministas	15076
ninata	15032
en tus dias	14190
a fregar	14013
mojigata	6008
marimacho	5770
para ser mujer	4693
nenaza	4358
odio a las mujeres	2749
lagartona	2006
A las mujeres hay que	1845
las mujeres no deberian	1285
las mujeres de hoy en dia	991
mujer al volante	962
mucho feminismo pero	852
mujer tenias que ser	683
pareces una puta	474
para ser chica	180
acabaras sola	50
hombre que te aguante	37
obsesionada con el machismo	8
pareces una fulana	5
no ha probado un hombre	1

Tabla 4.2: Número de tweets por término encontrados

recopilados por día. Como se puede observar, el número de tweets recopilados diariamente dista mucho de los 2400 que podrían generarse si se encontraran los 100 tweets objetivo para cada término. Además, a partir del día 29/11/2018 se reducen los tweets recopilados al encontrar los 15000 tweets objetivo para alguno de los términos. De todos estos tweets, se puede observar en la figura 4.2 como España es el país que más genera.

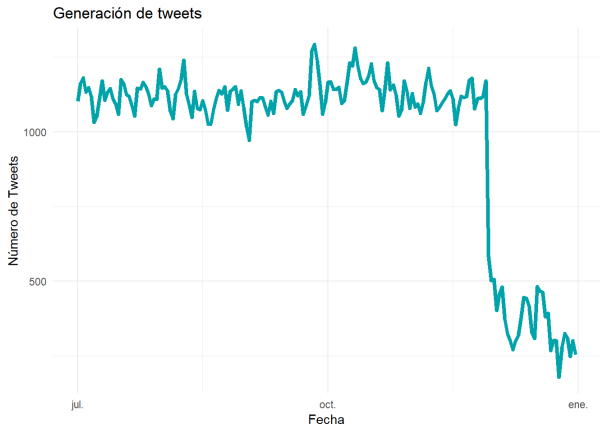

Figura 4.1: Tweets recopilados diariamente

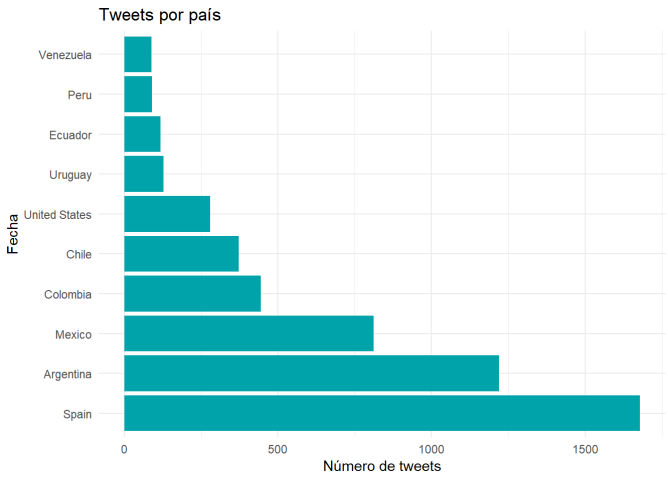

Figura 4.2: Número de tweets por país

Otra característica interesante es el uso de *hastag* en los tweets, en total, se han recopilado 15218 hastags. En la figura 4.3 se pueden observar los términos más utilizados. El *hastags* más utilizado coincide con uno de los términos utilizados para la recopilación de los datos por el *crawler*. El resto parecen estar relacionados con el programa televisivo "Gran Hermano". En

la figura 4.4 se puede observar cómo España es el país que más *hastags* utiliza.

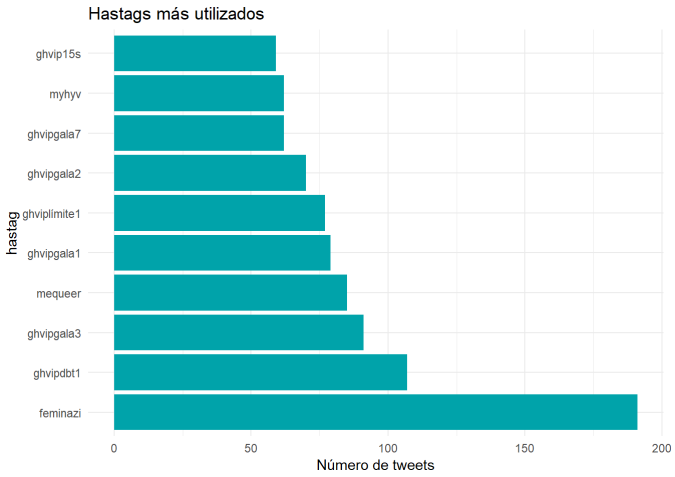

Figura 4.3: Número de hastags

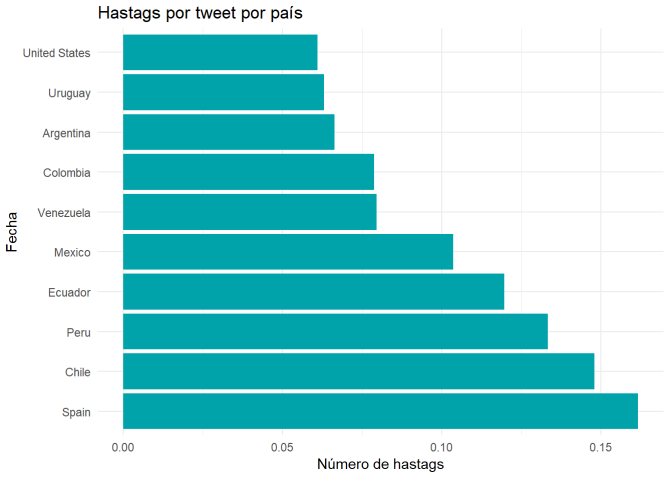

Figura 4.4: Número de hastags por país

Para identificar si los tweets utilizan una gramática adecuada, se ha observado las palabras OOV (*out of vocabulary*) utilizando un diccionario español. En la figura 4.5 se muestra como los Estados Unidos es el país con más OOV por tweet, llegando a 4 palabras oov por tweet, lo que indica que los mensajes producidos contienen un gran número de palabras que no se encuentran en ningún diccionario. Esto obligará a realizar un procesamiento para intentar normalizar este tipo de términos y permiten el correcto funcionamiento del sistema de clasificación.

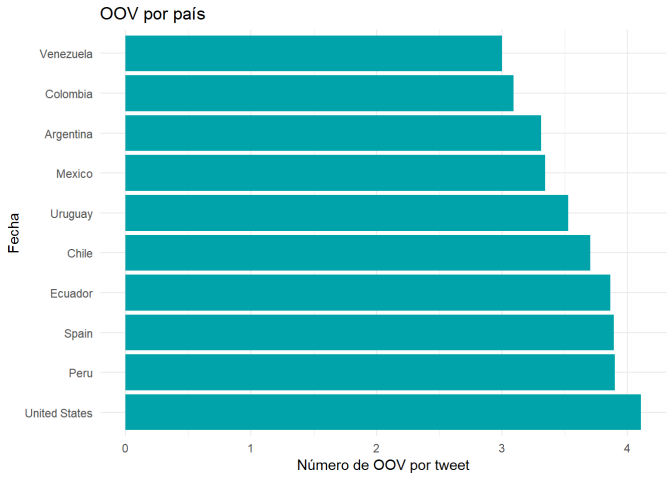

Figura 4.5: OOV por país

Por último, se han estudiado los enlaces por tweets. En total, se han recopilado 25539 enlaces para todos los mensajes. En la figura 4.6 se observa cómo España es el país que más enlaces por tweet utiliza.

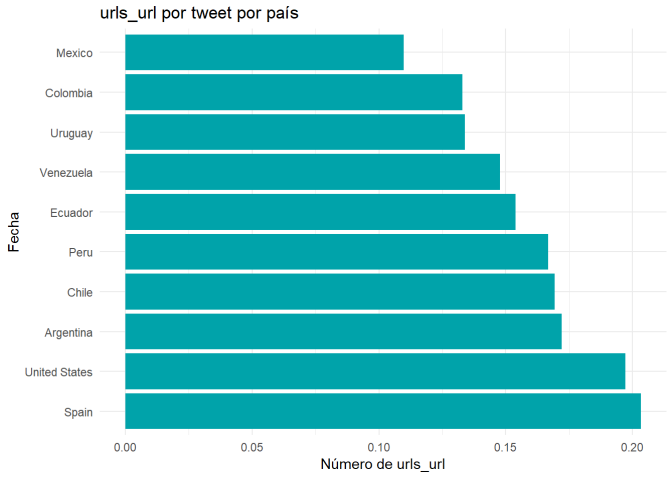

Figura 4.6: Número medio de URLs utilizadas por país

4.3. Etiquetado del corpus

Tras la ejecución del *crawler*, es necesario componer el corpus a etiquetar para poder entrenar un algoritmo de aprendizaje supervisado.

Para realizar el corpus, se han utilizado las 24 expresiones con más de 150 tweets que se pueden observar en la tabla 4.2. Para cada expresión o

término, se realiza un muestreo aleatorio de 150 tweets haciendo un total de 3600 tweets para ser etiquetados. Para todos los términos, se ha comprobado que la separación temporal entre sus tweets es más de un día, de este modo, se evitan mensajes que puedan ser conversaciones dentro del mismo día.

Teniendo en cuenta el objetivo principal del presente trabajo, se ha propuesto la clasificación del contenido del corpus generado en tres categorías distintas: machista, no machista y dudoso. A continuación, se definen las etiquetas propuestas:

- MACHISTA: tweets que contienen connotaciones machistas/ofensivas hacia las mujeres. Los tweets que pertenecen a esta categoría deben de contener actitudes discriminatorias hacia las mujeres por su sexo. Ejemplo: "*@EmanuelGPA Lo irónico es que lo dice una mujer, que "naturalmente" debería callarse y dedicarse a la cocina, limpiar y criar hijos*" En este ejemplo, se infravalora a la mujer de un modo expreso.
- NO_MACHISTA: tweets que no contienen connotaciones machistas. Ejemplo: "*¿Por qué cuando ven a una chica con pelo corto piensan que es bi, torta o marimacho? ...Capaz es solo un look o que el pelo largo le da mucho calor... Y si es...¿Cual es el problema?*" Dentro de esta categoría se clasifican también mensajes xenófobos y ofensivos en general pero que no discriminan a las mujeres por su sexo. Ejemplo: "*@kenia773 @LuisCarlos POR CIERTO, EN TU FOTO DE PERFIL SE PUEDE OBSERVAR QUE ERES BASTANTE VARONIL, ASÍ QUE SI NO ERES MARIMACHO, EMPIEZA A SERLO*"
- DUDOSO: tweets que, dependiendo del contexto, no presente en el tweet, podrían ser machistas (si el fragmento ofensivo se refiriese a mujeres). Ejemplo: "*@hazteoir @PSOE Más vale que se marche a fregar!*" Para poder considerar el tweet machista es necesario una referencia expresa a las mujeres, pese a que en el texto se pueda realizar una comparación con una mujer mediante una expresión machista.

El proceso de etiquetado ha sido llevado a cabo por 3 anotadores que han analizado los 3600 tweets que conforman el corpus para asignar una de las 3 etiquetas propuestas. Para ello, se ha desarrollado una guía de etiquetado en la que se explican las etiquetas, se proponen ejemplos de cada una de ellas y se establecen las normas de entrega (ver apéndice). El etiquetado se llevó a cabo entre los meses de enero y marzo de 2019 mediante la herramienta

"Google Sheets", un entorno de trabajo colaborativo online que permite la creación de hojas de cálculo.

Durante el anotado del corpus, cada uno de los 3 etiquetadores propone una clase de las 3 disponibles para cada tweets. Al término del etiquetado, se decidirá por votación de los 3 anotadores la etiqueta final. En caso de desacuerdo total entre los 3, un cuarto etiquetador decidirá la clase final para el tweet.

4.3.1. Dificultades encontradas en el etiquetado del corpus

Uno de los problemas más importantes del procesado del lenguaje natural es la ambigüedad, este fenómeno provoca que una misma expresión actúe con un significado distinto dependiendo del contexto en el que se encuentre. Además, la complejidad inherente al uso del lenguaje, el uso incorrecto de la gramática del idioma y fenómenos como la ironía o el sarcasmo dificultan la tarea de detección de actitudes machistas.

En la tarea de etiquetado, una de las dificultades más importantes ha sido detectar señales textuales que indiquen comportamientos machistas debido a los efectos descritos. Para ilustrar este efecto, se pueden considerar los siguientes ejemplos:

"*@tonifreixa La zorra guardando las gallinas. ¡¡ Que se encargue Rosell ¡¡ Bueno..., cuando salga de la cárcel. Cinismo en grado máximo.*"

"*@marijopellicer @radchiaru @_lxuli Jajajaja si no sabes cuanto odio a las mujeres tengo por favor no veas enemigos donde no los hay.*"

"*Cuando subes a tu amiga la lagartona al Uber porque ya andaba mala-copeando https://t.co/DcnK5ZGuL4*"

"*Mucho feminismo pero a la primera de cambio..... https://t.co/y2McecsgcT*"

En los ejemplos anteriores, los efectos del uso de la lengua dificultan la clasificación en alguna de las etiquetas propuestas. En el primer ejemplo, el término "zorra" no es evidente si se utiliza como referencia a un animal o como un sinónimo de prostituta.

El segundo ejemplo, se utiliza un tono irónico que dificulta su clasificación. El usuario utiliza la expresión "cuanto odio a las mujeres" con sarcasmo por lo que no es posible afirmar que conlleve una actitud machista. En el tercer ejemplo, se podría entender que el término "lagartona" se refiere a un animal, de hecho, el tweet enlaza una imagen de un lagarto en el interior de un coche, sin embargo, en este caso el término hace referencia a prostituta.

Por último, se presenta un tweet en el que se nombra al feminismo sin posicionarse en contra o a favor, en este caso sin poder detectar una actitud machista expresa.

Otra de las dificultades encontradas durante el proceso de etiquetado ha sido la clasificación de tweets en los que el usuario que escribe el mensaje cita un contenido machista, en ocasiones con el que está en desacuerdo. Este hecho se puede observar en los siguientes ejemplos:

"*Pareces una puta con ese pantalón. - Mi hermano de 13 cuando me vio con un pantalón de cuero*"

"*Cada vez más a menudo (todos lo días) mi padre me dice que las mujeres no deberían recibir premios, trabajar en puestos superiores, que son putas, y que deben quedarse en casa y servir al hombre y criar hijos.*"

"*@localgothgirI Me pasa todo el tiempo, incluso, cuando me va mal, siempre recibo insultos y comentarios como tenía que ser mujer las mujeres no deberían jugar este juego". A veces se me quitan las ganas de seguir jugando*"

"*@mariarodd17 Tía ella diciendo cosas tipo: es que las mujeres de hoy en día son muy sueltas y son unas acosadoras, estás ahora van preparadas con to y van a los hombres diciéndoles vamos a pasar un rato que ni te cobro y ellos los pobres si tienen pareja tienen que ir preparados para no caer*"

En los anteriores, hay contexto suficiente para comprender que se cita un contenido machista como ejemplo para apoyar su desacuerdo con ese tipo de lenguaje. Sin embargo, se ha decidido incluir tanto citas, chistes o incluso críticas al machismo como mensajes machistas. El objetivo principal de este trabajo ha sido identificar todo tipo de expresión machista para, de ese modo, determinar eso la cantidad de machismo existente en las redes sociales.

4.3.2. Resultados del etiquetado del corpus

Evaluación del acuerdo

Para valorar el acuerdo alcanzado por los etiquetadores en su labor se ha optado por el cálculo del coeficiente *kappa de Cohen* (Cohen, 1960). Esta medida permite realizar una estimación de la concordancia entre los etiquetadores de modo que se evalúen los resultados derivados del etiquetado manual. Una concordancia muy pobre podría indicar varios problemas en el proceso. Por una parte, podría ser que la naturaleza del problema dificulte la tarea de clasificación incluso para un humano, por lo tanto, se debería de

kappa	Tipo de acuerdo
0.00 - 0.20	Acuerdo muy débil
0.21 - 0.40	Acuerdo débil
0.41 - 0.60	Acuerdo medio
0.61 - 0.80	Acuerdo satisfactorio
0.81 - 1.00	Acuerdo excelente

Tabla 4.3: Umbrales de kappa

replantear el problema. Por otro lado, los etiquetadores podrían no haber entendido la tarea y sus criterios para realizar la tarea no fueran homogéneos. La expresión se calcula del siguiente modo:

$$k = \frac{p_0 - p_c}{1 - p_c}$$

donde p_0 representa el acuerdo observado relativo entre los etiquetadores o proporción de acierto, y p_c es la probabilidad hipotética de acuerdo por azar. Considerando p clases distintas y N el número de elementos de la matriz de confusión:

$$p_0 = \frac{1}{n} \sum_{i=1}^{p} n_{ii}$$

$$p_c = \frac{1}{n^2} \sum_{i=1}^{p} n_{i\cdot} n_{\cdot i}$$

Un valor de $k = 0$ indica que la proporción de acuerdo es igual a la probabilidad de acuerdo por azar. Para evaluar el nivel de acuerdo según el valor de este medida se suele utilizar como convención la tabla 4.3.

Acuerdo etiquetadores

Se realizó una evaluación de la medida kappa en 2 puntos: Al completar el 20 % del etiquetado de todo el corpus y al etiquetar el corpus completo. En la primera evaluación, se obtuvieron los resultados de la tabla 4.4. En este caso, se valoró que para el número de clases distintas este valor del coeficiente kappa era insuficiente y se revisaron los patrones de etiquetas propuestos por cada uno de los etiquetadores. De este modo, se detectaron los siguientes problemas:

	Kappa
Etiquetador 1-2	0,5
Etiquetador 1-3	0,44
Etiquetador 2-3	0,58
Media Etiquetadores	**0,51**

Tabla 4.4: Kappa obtenido con el 20 % del etiquetado

	Kappa
Etiquetador 1-2	0,76
Etiquetador 1-3	0,78
Etiquetador 2-3	0,74
Media Etiquetadores	**0,76**

Tabla 4.5: Kappa obtenido con el 20 % del etiquetado tras la corrección

- El etiquetador 3 utilizó más que el resto la etiqueta MACHISTA. Se concluyó que confundía tweets por actitudes xenofobas por tweets con actitudes machistas.
- El etiquetador 1 utilizó más que el resto la etiqueta DUDOSO. Se revisaron sus criterios de valoración de acuerdo con la guía de anotación.

Tras revisar los criterios de cada uno de los dos etiquetadores se obtuvieron los resultados de la tabla 4.4. En este caso, el coeficiente kappa alcanzado supera el valor 0.6 indicado como un resultado adecuado para este tipo de tarea.

Por último, se volvió a valorar la métrica kappa al término del etiquetado, alcanzando los resultados de la tabla 4.5. La distribución de etiquetas elegida por cada anotador se observa en la figura 4.7.

	Kappa
Etiquetador 1-2	0,68
Etiquetador 1-3	0,68
Etiquetador 2-3	0,88
Media Etiquetadores	**0,75**

Tabla 4.6: Kappa obtenido con el 100 % del etiquetado

Etiqueta	Veces asignada
NO_MACHISTA	2181 (60.58 %)
MACHISTA	1152 (32 %)
DUDOSO	267 (7.42 %)

Tabla 4.7: Distribución de la clase para el corpus final

% ETIQUETADO	¿CUANTOS MACHISTAS?	¿CUANTOS NO_MACHISTAS?	¿CUANTOS DUDOSOS?
Etiquetador 1	30,0	62,6	7,4
Etiquetador 2	31,4	60,9	7,7
Etiquetador 3	32,6	59,0	8,4

Figura 4.7: Porcentaje de etiquetas elegidor por etiquetador

Exploración de la clase

En el presente apartado se presentan los resultados derivados del proceso de etiquetado. En concreto, se va a estudiar la relación de la clase asignada por los etiquetadores con dos de los tipos de atributos disponibles en el corpus: atributos numéricos y categóricos.

Para el corpus final, se han asignado los valores de clase reflejados en la tabla 4.7. Como se puede observar, debido a la naturaleza del problema, los valores de la clase están muy desbalanceados para el valor "NO_MACHISTA", contando este con más del 60 % de los tweets del corpus.

Utilizando los valores de la clase etiquetada, se han representado las variables numéricas de las que se disponen en el corpus (figura 4.8). Dos de las más destacadas se pueden observar en la figura 4.9, donde se representan el tamaño de los tweets en número de caracteres y de *retweets*. Un aspecto a destacar es que los tweets DUDOSOS son, en media, mucho más cortos que el resto de valores de la clase; sin embargo, los tweets MACHISTAS y NO_MACHISTAS siguen una distribución muy similar en cuanto al número de caracteres.

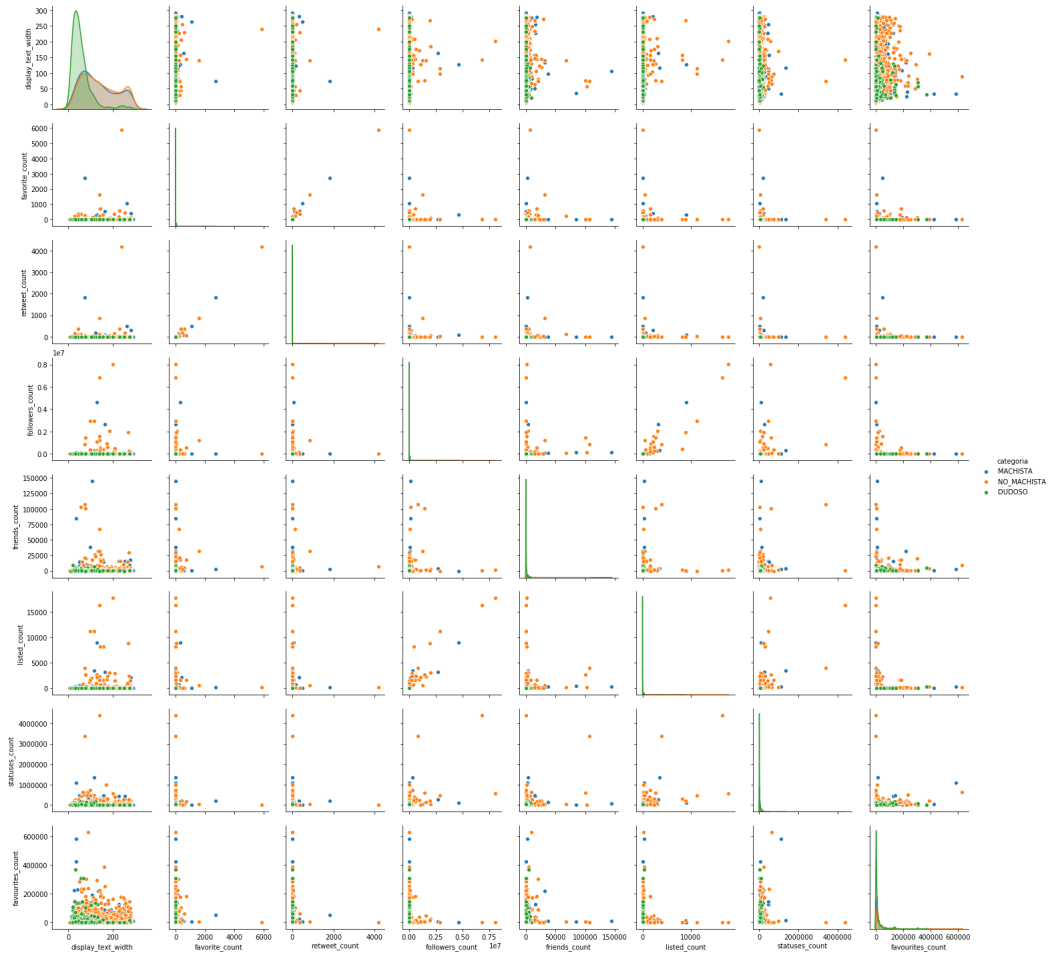

Figura 4.8: Representación de valores numéricos de los tweets en función de la clase

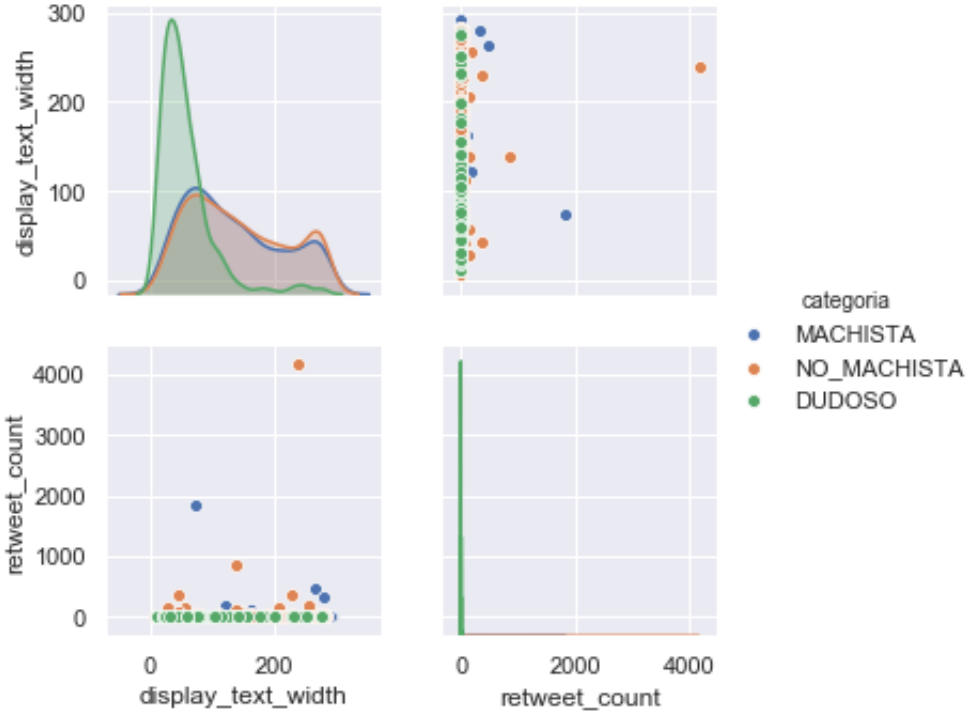

Figura 4.9: Representación de valores numéricos relevantes

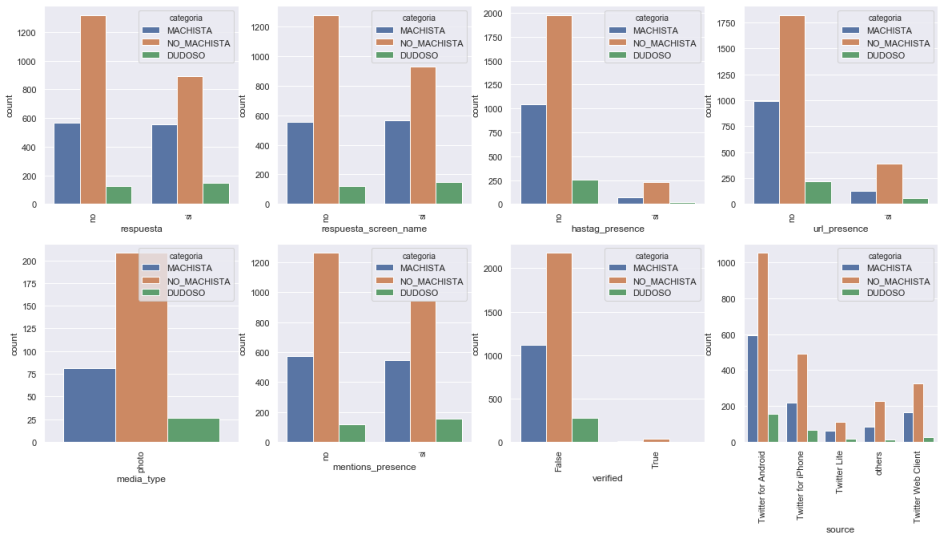

Figura 4.10: Representación de variables categóricas

En cuanto a las variables categóricas, en la figura 4.10 se representan los valores para cada atributo respecto de los valores de la clase. En este caso, la distribución de todas las variables categóricas es muy similar y no

parece que la separabilidad de la clase sea tan elevada como en el caso de la figura 4.9. Un aspecto a destacar se puede observar en la variables "respuesta" y "respuesta_screen_name" donde parece que la probabilidad de tweet machistas es más elevada en el caso de que estos atributos sean afirmativos.

Capítulo 5

Sistema propuesto

En el presente capítulo se describe el sistema automático de detección del machismo en redes sociales desarrollado en el presente trabajo fin de máster. El sistema está basado en aprendizaje supervisado y emplea distintos atributos unificados para, posteriormente, hacer uso de un algoritmo de aprendizaje de máquina que clasificará los registros de entrada en 3 categorías: MACHISTA, NO_MACHISTA Y DUDOSO. De este modo, los mensajes de texto o tweets de entrada se clasificarán según el grado de machismo que presenten.

El sistema ha sido desarrollado teniendo en cuenta la naturaleza del problema tratado. En este caso, como ya se introdujo, se trata de texto en español por lo que método de clasificación estará preparado para trabajar con este idioma. No obstante, la arquitectura del sistema y las técnicas aplicadas pueden ser adaptadas perfectamente para otros idiomas.

El sistema propuesto se divide en tres fases principales, tal y como se puede observar en la figura 5.1.

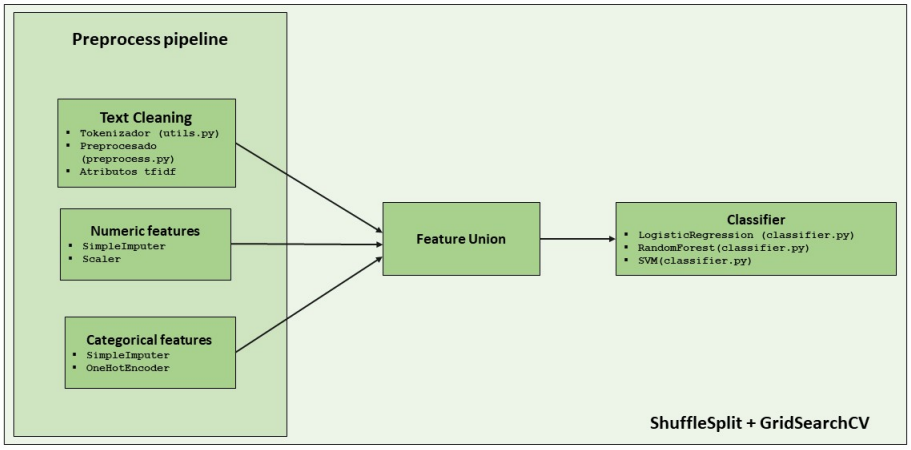

Figura 5.1: Arquitectura clasificador

En la primera etapa, se realiza un preprocesado diferenciado según el tipo de atributo. En este sistema, se consideran tres tipos de atributos distintos: variables categóricas, numéricas y texto. Para cada atributo, se aplican diferentes métodos como la tokenización, el escalado o la sustitución de emoticonos. Tras esto, se unifican los distintos tipos de atributos procesados en un conjunto de datos común que será la entrada de la última fase. En la última etapa, se emplean algoritmos de clasificación supervisada con la intención de obtener un modelo predictivo capaz de detectar textos machistas. Se emplean tres métodos distintos de aprendizaje automático en este último paso: Regresión logística, Random Forest y Máquinas de vectores de soporte (SVM, "Support Vector Machine"). Para evaluar los resultados obtenidos con estos algoritmos, se realiza una búsqueda de parámetros de entrada (GridSearchCV) y, posteriormente, se utilizan estos parámetros para realizar una validación cruzada en el conjunto de testeo.

5.1. Preprocesado

La primera fase del sistema de clasificación lleva a cabo diferentes acciones relacionadas con el preprocesado. Se realizan tareas tan importantes como la división del texto en tokens que permitirá que la información lingüística sea tratada por las sucesivas etapas del sistema de clasificación. Como se

introdujo, el tipo de preprocesado depende del tipo de atributo considerado.

5.1.1. Texto

El atributo que contiene el texto del tweet es el más importante para la realización del sistema automático de detección del machismo. Idealmente, se debería encontrar en este atributo todas las señales necesarias que indican si el mensaje es machista o no. Para este atributo, se aplican tres métodos distintos: preprocesado, tokenizador y creación de atributos tf-idf. En el método de preprocesado, se llevan a cabo las siguientes acciones:

- Reemplazo de emojis: Se reemplazan los emojis por una descripción de lo que representa el dibujo. En este caso, resulta útil poder identificar el emoji que se está utilizando ya que puede modificar el significado de la frase. Por ejemplo, en la frase de la imagen 5.2 se puede observar como el emoji permite identificar como machista la expresión:

Figura 5.2: Uso de emoji en contexto machista

- Filtrado de URLs: Se reemplazan las URLs por la palabra "*twurl*".
- Reemplazo de acentos: Se eliminan los acentos propios del castellano.
- Filtrado de usuarios: Se reemplazan las menciones de los usuarios (las palabras que comienzan por "@") por la palabra "*twuser*". De este modo, se identifica cuando se utilizan las menciones omitiendo el usuario concreto.
- Convertidor de hastags: Se realiza una conversión para los *hastags* que utilizan las mayúsculas como separador. Por ejemplo, la frase "*#FelizDía*"se convertiría a "*Feliz Día*".
- Filtrado de hastags: Se reemplazan los *hastags* (las palabras que comienzan por "#") por la palabra "*twhastag*".
- Convertidor a minúsculas: Se convierten todos los caracteres a minúscula.
- Reemplazo de exclamaciones: Se reemplazan los signos de exclamación por la palabra "*twexclamation*".

- Reemplazo de interrogaciones: Se reemplazan los signos de interrogación por la palabra "*twinterrogation*".
- Reemplazo de signos de puntuación: Se eliminan los signos de puntuación.

Para ilustrar el funcionamiento de la etapa de preprocesado, se supone el siguiente mensaje:

Figura 5.3: Ejemplo de preprocesado

Utilizando el preprocesado se obtendría la siguiente oración transformada: "*esta es la reina de las feministas de verdad o no twinterrogation twuser thumbs_up twurl*". Como se puede observar, se han convertido los caracteres a minúscula, se ha reemplazado el caracter de interrogación, se ha sustituido la mención del usuario, se reemplaza el emoji por una descripción y se reemplaza la URL.

A partir del texto preprocesado, se realiza la tokenización de cada mensaje. Para llevar a cabo este proceso se utiliza la clase "*TweetTokenizer*" disponible en la librería "NLTK". Se trata de un tokenizador desarrollado específicamente para el texto generado en Twitter. Tras la tokenización, se obtiene para cada mensaje una lista de unidades independientes que, mayoritariamente, representarán palabras, emoticonos y signos de puntuación.

Una vez realizada la tokenización del mensaje, se llevan a cabo tres procesos: filtrado de stopwords, reemplazo de abreviaturas y *stemming*. Las palabras vacías o stopwords constituyen el grupo de palabras sin un significado concreto, por ejemplo, artículos, preposiciones o conjunciones. Este tipo de elementos no aportan información adicional al contexto del mensaje y, por tanto, se eliminan antes del proceso de clasificación.

La tarea llevada a cabo en este trabajo presenta una dificultad añadida por el entorno en el que se utiliza el lenguaje. En las redes sociales, y en Twitter en concreto, se publican mensajes cortos y, frecuentemente, no siguen las reglas convencionales del idioma. De este modo, el uso de abreviaturas o emoticones está muy extendido en este tipo de plataformas. Es por ello, que realizar diccionarios específicos para cada idioma que permitan normalizar este tipo de contenido es muy importante previo a la tarea de clasificación. En este trabajo, se utiliza el diccionario en castellano realizado en

(Helena Gómez-Adorno, 2016) que compila diccionarios de palabras "slang", abreviaturas, contracciones y emoticonos que ayudan al preprocesamiento de textos publicados en redes sociales. Un ejemplo del tipo de expresiones que se reemplazan en este proceso se podría ilustrar del siguiente modo: "*Esta es aki la reuna de las feministas de verdad o no?*". En el ejemplo anterior, se encuentra la palabra coloquial .ªki"que normalizada al español sería "aquí".

Por último, se aplica un método de stemming para reducir las palabras a su raíz. En este caso, se utiliza el algoritmo de Porter (Porter, 1980) desarrollado en la década de los ochenta por Martin Porter y basado en un conjunto de reglas aplicadas en cascada para obtener la raíz de las palabras.

Para ilustrar el proceso anterior, se supone el ejemplo de la figura 4.3. Aplicando el proceso de tokenización se obtendría la siguiente lista de tokens: ['reina', 'feminista', 'verdad', 'twinterrog', 'twuser', 'thumbs_up', 'twurl']. Como se puede observar, se obtiene una lista de palabras, emoticonos y signos de interrogación entre los cuales no se encuentran las palabras vacías.

A partir del texto tokenizado, es necesario representar la información contenida en el texto de un modo interpretable por las etapas posteriores del sistema de clasificación. Para el presente trabajo, se realiza esta representación utilizando vectores de términos *tf-idf* mediante los unigramas de los tokens obtenidos del proceso anterior.

5.1.2. Atributos numéricos

Otro tipo de atributo que se utiliza en el presente sistemas son los numéricos. En este caso particular, se consideran los siguientes atributos numéricos:

- display_text_width: número de caracteres del tweet.
- favorite_count: número de veces que el tweet ha sido marcado como favorito.
- retweet_count: número de veces que el tweet ha sido retwiteado.
- followers_count: número de seguidores del usuario que publica el tweet.
- friends_count: número de personas seguidas por el usuario que publica el tweet.
- listed_count: número de listas en las que está inscrito el usuario que publica el tweet.

- statuses_count: número de tweets publicados por el usuario que publicó el tweet.
- favourites_count: número de tweets que el usuario que publicó el tweet marcó como favorito.

El uso de este tipo de atributos permite tener en cuenta distintos aspectos del contexto en el que se genera el tweet. Por ejemplo, existe un grupo de atributos como favorite_count o retweet_count que mide la popularidad del tweet. Este tipo de atributos permite valorar la posible propagación de un mensaje machista en Twitter. Por otra parte, existen campos como followers_count que miden la popularidad del usuario que publica el tweet. Además, otros campos como display_text_width demuestran tener una relevancia especial para los tweets etiquetados con categoria "DUDOSO" (figura 4.9).

En los atributos numéricos se realizan únicamente dos procesos: imputación de valores nulos y escalado. En este caso, se imputan los valores nulos sustituyéndolos por 0 y, para el escalado, se realiza una estandarización para que en los valores numéricos se consiga una media nula y una desviación estándar de uno.

5.1.3. Atributos categóricos

El último tipo de atributo que se emplea son los atributos categóricos. Para este sistema, se consideran los siguientes atributos categóricos:

- source: tipo de dispositivo con el que se publica el tweet.
- respuesta: indica si el tweet es una respuesta a otro.
- respuesta_screen_name: nombre del usuario al que se responde.
- hastag_presence: indica la presencia de "hastags" en el tweet.
- url_presence: indica la presencia de URLs en el tweet.
- media_type: indica si el tweet contiene imagenes o videos.
- mentions_presence: indica la presencia de la mención a algún usuario en el tweet.
- verified: indica si el usuario que publica el tweet es verificado por Twitter.

Al igual que en el caso de los atributos numéricos, el uso de algunos de estos atributos categóricos intentan recoger el contexto en el que se publica el mensaje. Por ejemplo, el campo "verified" mide la influencia del usuario y los atributos "url_presence" o "media_type" indica si el mensaje comparte otro contenido distinto a su texto.

En los atributos categóricos únicamente se aplica una transformación para convertir esta información a tipo numérico y que sea posible utilizarlo en el posterior algoritmo de clasificación. En este caso se emplea la codificación "one-hot" que crea un nuevo atributo por cada valor del atributo categórico asignando 1 ó 0 según la existencia o no de ese valor para cada registro.

5.2. Unión de atributos

En la segunda fase del sistema, se combinan todos los atributos que han sido preprocesados en la etapa anterior. Para cada tipo de atributo, se ha descrito el procesamiento realizado con la idea de acondicionar el conjunto de datos que se utilizará para ajustar un modelo de aprendizaje supervisado.

En esta etapa se combinan los tres tipos distintos de atributos que se consideran:

- texto: contiene los atributos tf-idf extraídos del texto preprocesado.
- numéricos: se incorporan todos los atributos numéricos considerados habiendo realizado previamente un proceso de estandarización.
- categóricos: se incorporan todos los atributos categóricos considerados habiendo realizado una transformación de sus valores a atributos numéricos.

La salida de esta etapa será un conjunto de datos que contendrán los tres tipos de atributos descritos, que serán la entrada de la última fase del sistema.

5.3. Clasificación

La última etapa del sistema realiza el ajuste de un algoritmo de aprendizaje de máquina a los datos obtenidos en las fases anteriores.

El objetivo final del presente trabajo es la detección del lenguaje machista en las redes sociales. Este problema se puede modelar como una clasificación de documentos en los que se asignará una categoría a cada uno de los mensajes que conforman el corpus.

En la actualidad, la mayoría de trabajos llevan a cabo esta tarea mediante algoritmos o ténicas de aprendizaje supervisado (Zimmerman, 2018; Thomas Davidson, 2017; E. Fersini y Anzovino, 2018b). Para la tarea de clasificación se emplean tres algoritmos distintos disponibles en "scikit-learn": Regresión logística, Random Forest y SVM. Para todos ellos, se realiza una búsqueda de parámetros óptimos para distintas configuraciones teniendo en cuenta el tiempo de computación de esta búsqueda:

- Línea base 1: Se clasifican todos los registros del testeo con la categoria mayoritaria.
- Línea base 2 (atributos de texto): Se aplica regresión logística con la siguiente búsqueda de parámetros C = [1, 10], class_weight'= [None, 'balanced'].
- Regresión logística (atributos numéricos, categóricos y texto): Se aplica una regresión logística a todos los atributos disponibles con la siguiente búsqueda de parametros: C = [1, 10], class_weight'= [None, 'balanced'].
- Random Forest (atributos numéricos, categóricos y texto): Se utiliza el algoritmo Random Forest con todos los atributos disponibles y los siguientes parámetros: n_estimators- [250, 450], bootstrap'= (True, False), max_depth'= [None, 30].
- SVM (atributos numéricos, categóricos y texto):Se utiliza el algoritmo SVM con todos los atributos disponibles y los siguientes parámetros: C = [1, 10, 100, 10000], gamma = [0.001, 0.1, 0.6, áuto'], kernel = 'rbf'.

Se ha seleccionado un subconjunto muy reducido para cada configuración debido a las limitaciones computacionales.

Capítulo 6

Evaluación y discusión

En el siguiente capítulo se presentan los procedimientos de evaluación así como los resultados de los experimentos realizados. Además, se realiza una comparativa y discusión de los distintos resultados obtenidos. Para evaluar el sistema propuesto, se definen dos experimentos según el procedimiento aplicado para realizar la evaluación del sistema. En el primero, se reservan una parte de los datos para realizar una búsqueda de los hiperparámetros óptimos para cada algoritmo de clasificación mientras que en el segundo se emplean los parámetros por defecto para evitar el sobreajuste. Con estos experimentos, se pretende evaluar el rendimiento del sistema propuesto para detección del machismo en redes sociales. Asimismo, se evaluará el efecto que tiene en el sistema el desbalanceo existente en la clase del conjunto de datos.

6.1. Metodología de evaluación

6.1.1. Métricas de evaluación

Para la evaluación de los resultados en clasificación textual o de documentos se utiliza comúnmente la matriz de confusión. Se trata de una herramienta que representa en cada columna el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. En la siguiente imagen se presenta un esquema de la matriz de confusión:

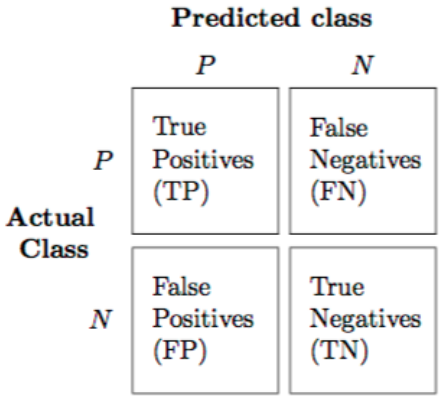

Figura 6.1: Matriz de confusión

Esta tabla está formada por verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. De este modo, si un documento es clasificado por el sistema automático en la misma categoría que la clasificación manual, se considerará como un verdadero positivo o negativo (*True Positive*, TP o *True Negative*, TN), mientras que si el documento es clasificado por el sistema con una categoría diferente, se estará ante un falso negativo o falso positivo (*False Positive*, FP o *False Negative*, FN).Utilizando estos cuatro componentes se calculan las medidas principales para evaluar los resultados:

- Tasa de acierto o exactitud (*accuracy*): representa el porcentaje de aciertos en relación a todos los documentos clasificados.

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

- Precisión: representa la fracción de asignaciones correctas frente al total de asignaciones positivas realizadas para esa clase. Es decir, realiza una medida de la tasa de acierto para un valor de la clase.

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

- Cobertura (*recall*): representa la fracción de asignaciones positivas respecto al conjunto real de elementos pertenecientes a la clase. Es decir, realiza una medida de la capacidad que tiene el clasificador de detectar elementos de esa clase.

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

- Medida-F: combina las medidas de precision y cobertura.

$$Medida - F = \frac{2xprecisionxcobertura}{precision + cobertura}$$

6.1.2. Colección de evaluación

Las colecciones de evaluación son conjuntos de datos etiquetados con información relevante para la tarea para la cual han sido desarrollados. En este caso, las colecciones de evaluación para clasificación de documentos están compuestas por textos, ya sean oraciones, párrafos o documentos completos, de distinta naturaleza y que están etiquetados con categorías concretas. Por ejemplo, para el presente trabajo, existen tres valores posibles para esta categoría: "MACHISTA", "NO_MACHISTA" y "DUDOSO".

Estos conjuntos de evaluación permiten intuir el rendimiento de los sistemas de clasificación y compararlo con el de otros sistemas. Asimismo, en los sistemas de clasificación supervisados, son clave para poder ser entrenados utilizando un subconjunto de la colección.

Para el presente trabajo, se utilizará como conjunto de evaluación del sistema de clasificación de contenido machista el corpus presentado en el capítulo 4. Se trata de un corpus compuesto por 3600 tweets recopilados mediante el uso de expresiones que pueden conllevar actitudes machistas.

Para recuperar esta información se utilizaron los siguientes términos: "feminazi", "loca del", "a la cocina", zorra, "como una niña", "las feministas", niñata, "como una mujer", "en tus días", "a fregar", mojigata, marimacho, nenaza, "para ser mujer", "odio a las mujeres", lagartona, "A las mujeres hay que", "las mujeres no deberían", "las mujeres de hoy en día", "mujer al volante", "mujer tenías que ser", "mucho feminismo pero", "pareces una puta", "para ser chica". De este modo, se recopilaron todos los mensajes escritos en la red social que contuvieran estos términos durante las fechas 1/07/2018 - 31/12/2018.

El propósito principal de este corpus es la obtención de texto con alto contenido machista así como expresiones que, aún pudiendo ser machistas, no lo sean en ese contexto. De este modo, se pretende obtener un conjunto rico en el uso de expresiones que pueden conllevar actitudes machistas en diferentes contextos.

6.1.3. Líneas base (*baseline*)

Como se ha ido introduciendo en el capítulo 2, las referencias en el campo de detección del machismo son muy reducidas y, por tanto, es complejo encontrar algún trabajo comparable con el sistema desarrollado. Es por esto que en este trabajo se han desarrollado dos líneas base con las que comparar los resultados obtenidos por el sistema diseñado. La primera de ellas plantea un sistema de clasificación basado en una regresión logística sobre los atributos *tf-idf* utilizando los unigramas de cada documento. De este modo, se plantea un sistema sencillo pero pudiendo ser, en ocasiones, mucho más efectivo que otras aproximaciones más complejas que utilizan bi-gramas o categorías gramaticales de los términos. Por tanto, se trata de un *baseline* difícil de batir.

La segunda línea base está basada en un clasificador sencillo que predice siempre la clase mayoritaria. En este caso, como se puede observar en la tabla 4.7, la clase mayoritaria sería "NO_MACHISTA" y, por tanto, este sistema clasificaría todos los registros de entrada con este valor de clase. La intención de esta línea base es comparar los resultados del sistema con otro hipotético no informado que no es capaz de "aprender" ningún patrón del conjunto de datos de entrenamiento.

6.1.4. Experimento 1: Búsqueda de hiperparámetros mediante la optimización de la medida F1

El primer experimento realizado trata de configurar cada algoritmo de clasificación para la tarea especifica que van a desarrollar. Como se introdujo en el capítulo 5, para la tarea de clasificación se emplean tres algoritmos distintos disponibles en "scikit-learn": Regresión logística, Random Forest y SVM. En este primer experimento, se realiza una búsqueda para los siguientes parámetros:

- Regresión logística: C = [1, 10], class_weight'= [None, 'balanced'].
- Random Forest: n_estimators- [250, 450], bootstrap'= (True, False), max_depth'= [None, 30].
- SVM: C = [1, 10, 100, 10000], gamma = [0.001, 0.1, 0.6, áuto'], kernel = 'rbf'.

Para ello, se sigue el procedimiento presentado en la figura 6.2 de forma

iterativa. En el primer paso, se realizan 10 repartos o subconjuntos aleatorios del corpus en dos conjuntos de datos: entrenamiento (training) y testeo (testing). Para el conjunto de training, se reservan el 30 % de los datos y para el test, el resto. Para cada uno de los diez repartos, se utiliza el conjunto de training para la búsqueda de hiperparámetros y el testing para evaluar los resultados con los parámetros encontrados.

Para la búsqueda de parámetros, se realiza una validación cruzada (*cross validation*) con cinco grupos realizando el entrenamiento en cuatro de ellos y el testeo en el grupo restante. Este proceso se repite para los cinco grupos y permite obtener los parámetros que mejor han funcionado en el proceso según el valor de la medida F1.

En la siguiente etapa del proceso se utiliza el segundo conjunto de datos reservado para el testeo (el 70 % de todo el corpus) y se realiza la evaluación final. Utilizando los parámetros de entrada obtenidos en la etapa anterior, se realiza una validación cruzada con 10 grupos del conjunto de testing. De nuevo, en este proceso se realiza un reparto en 10 grupos, donde nueve de ellos serán utilizados para el entrenamiento y el grupo restante para el testeo, repitiendo el proceso diez veces, una por grupo. Estas dos etapas descritas se repiten para los diez repartos indicados al inicio.

Este experimento permite medir el resultado de un sistema diseñado especificamente para esta tarea pues la configuración de los algoritmos de clasificación se realiza según los datos del corpus. Además, al realizar diez iteraciones para el proceso, la varianza de los resultados se reduce y son menos dependientes del tipo de datos con el que se ha entrenado. La desventaja principal de este método es que es necesario reservar un conjunto de datos para la búsqueda de parámetros y se reduce la información de la que dispondrá el sistema de clasificación definitivo que realizará la predicción.

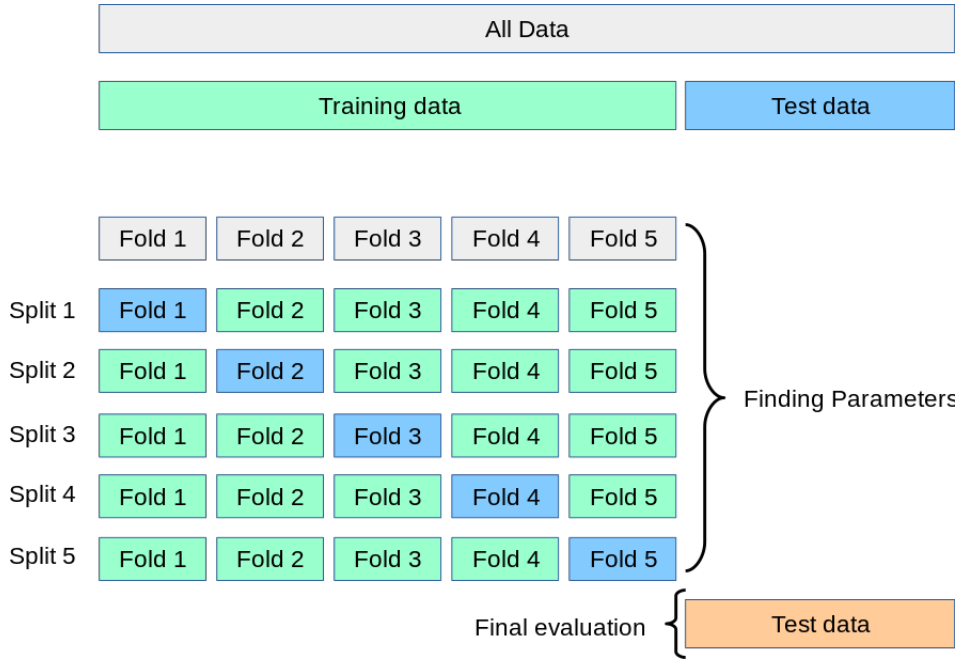

Figura 6.2: Búsqueda de hiperparámetros mediante la optimización de la medida F1

6.1.5. Experimento 2: Cross validation con parámetros por defecto

El segundo experimento consiste en una única validación para todo el corpus utilizando los parámetros por defecto para todos los algoritmos de clasificación utilizados. En este caso, se ha optado por una validación cruzada con diez grupos. En la figura 6.3 se presenta un ejemplo equivalente para cinco grupos.

En este procedimiento, se realiza una división del conjunto en 10 grupos del mismo tamaño del corpus y, de forma iterativa, se utilizarán nueve de ellos para el entrenamiento y el grupo restante para el testeo.

Este método permite evaluar un sistema más general cuyos parámetros de configuración no estén diseñados para los datos de entrenamiento de los que se disponen. De este modo, se podría mejorar la capacidad de generalización del sistema y evitar un posible sobreajuste.

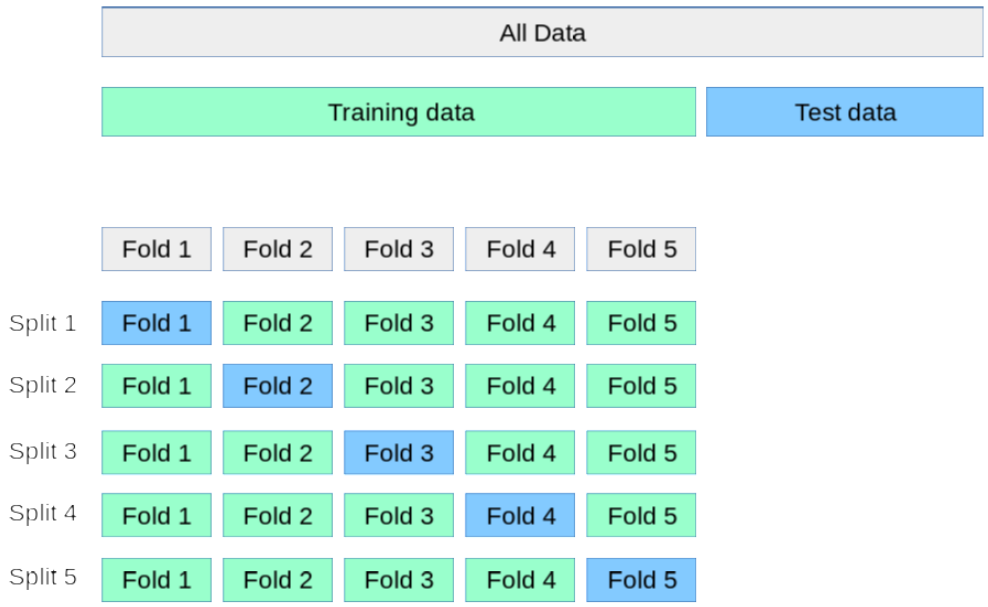

Figura 6.3: Validación cruzada k=5

6.2. Resultados experimento 1

La tabla 6.1 muestra los resultados promedio de exactitud (*Accuracy*), medida F1, cobertura (*Recall*) y precisión. Con el método de evaluación descrito para el primer experimento, el algoritmo *Random Forest* alcanza una mayor tasa de acierto y precisión mientras que la regresión logística alcanza los mejores resultados para la medida F1 y *recall*.

En relación a la comparación de los resultados obtenidos por el método con las dos líneas bases, en este caso las diferencias con respecto a la aproximación basada en unigramas son de unos cuatro puntos porcentuales para cada medida. Por tanto, el sistema mejora esta primera aproximación en todas las medidas pero, como se preveía, la línea base ya es un buen punto de partida del sistema.

En el caso de la línea base basada en el clasificador sencillo de la clase mayoritaria, sí se pueden observar grandes diferencias en las métricas de calidad. Esto indica que cualquiera de las soluciones propuestas será mucho más adecuada que un clasificador basado en una única regla sencilla.

Los resultados presentados en la tabla 6.1 muestran que el sistema de clasificación desarrollado en el presente trabajo se comporta mejor que las lineas base en todas las métricas de calidad. Estos buenos resultados parecen

	Accuracy	**F1**	**Recall**	**Precision**
Baseline (tf-idf)	0.68	0.59	0.62	0.59
Baseline	0.61	0.2	0.3	0.24
LR	0.7	**0.62**	**0.64**	0.62
RF	**0.72**	0.6	0.57	**0.67**
SVM	0.7	0.61	0.63	0.61

Tabla 6.1: Resultados experimento 1

confirmar la hipótesis de que cualquier modelo utilizando todos los atributos disponibles en el corpus resulta más apropiado que solo el uso del texto disponible para cada registro.

En concreto, la diferencia entre el mejor de los clasificadores del método propuesto y la línea base basada en unigramas es de 4 %. Esta diferencia se debe principalmente a la información que aportan los atributos numéricos y categóricos para la tarea del clasificación. Esto se confirma en la figura 4.9, donde se puede observar como la longitud del tweet y el número de *retweets* tienen influencia en la clasificación. Se puede intuir como en el caso de los tweets dudosos, la longitud media es mucho menor que en el caso de los machistas y no machistas. Además, los tweets dudosos no presentan gran cantidad de *retweets* si se comparan con el resto de clases. Es por esto, que utilizar estos atributos en el sistema de clasificación aporta información útil y permite mejorar los resultados.

Esta hipótesis se reafirma con los resultados de la figura 6.4. En esta figura, se observan los valores SHAP (*SHapley Additive exPlanations*) (Strumbelj y Kononenko, 2014) para los tweets dudosos. Esta técnicas permite desglosar cada predicción individualmente para mostrar el impacto que tiene cada atributo en la clasificación. El primer atributo (*display_text_width*) coincide con la longitud del tweet, y, se confirma que a menor valor en este atributo (menor longitud del tweet) más probable es que el sistema clasifique el texto como dudoso. Cabe destacar otros atributos como los términos como "nenaza" y "feminazi" en los que se observa que la presencia de estos términos reduce la probabilidad de que el sistema clasifique el tweet como dudoso. Esto se debe a que la existencia de estos términos en los tweets se han asociado con comportamientos machistas durante el proceso de etiquetado, como se puede puede observar en la figura 6.5 y 6.6. Otro atributo interesante que provocó inconvenientes durante el proceso de etiquetado manual del corpus

es la existencia del término "lagartona". En este caso, la existencia de este término provoca que aumente la probabilidad de que el tweet sea dudoso, esto ocurre por la ambigüedad de la palabra que puede hacer referencia a una persona que vende su cuerpo a cambio de dinero o a una persona pícara. Esta confusión provoca que, si existe el término en un mensaje, aumente la probabilidad de ser dudoso. Otro término que ha provocado confusión durante el etiquetado ha sido "niñata". Como se puede observar en la figura 6.6, la existencia de este término aumenta la probabilidad de que el tweet sea no machista pues, por si solo, este término no conlleva una actitud machista. A pesar de esto, se trata de un término que, según el contexto en el que sea empleado, podría ser considerado machista.

Estos atributos numéricos y categóricos influyen también para el resto de valores de la clase. De hecho, se puede observar en la figura 6.7 como las tres primeras variables en importancia son variables no textuales. Por ejemplo, la variable "statuses_count" indica la cantidad de mensajes emitidos por el usuario que publica el tweet y, en este sistema, tiene un impacto razonable para clasificar los tweets como machistas y no machistas. Esto podría explicarse por el hecho de que, cuanto más activo sea el usuario en la red social, más probable es que exprese sus ideas acerca del machismo, ya sea a favor o en contra con el uso de expresiones machistas.

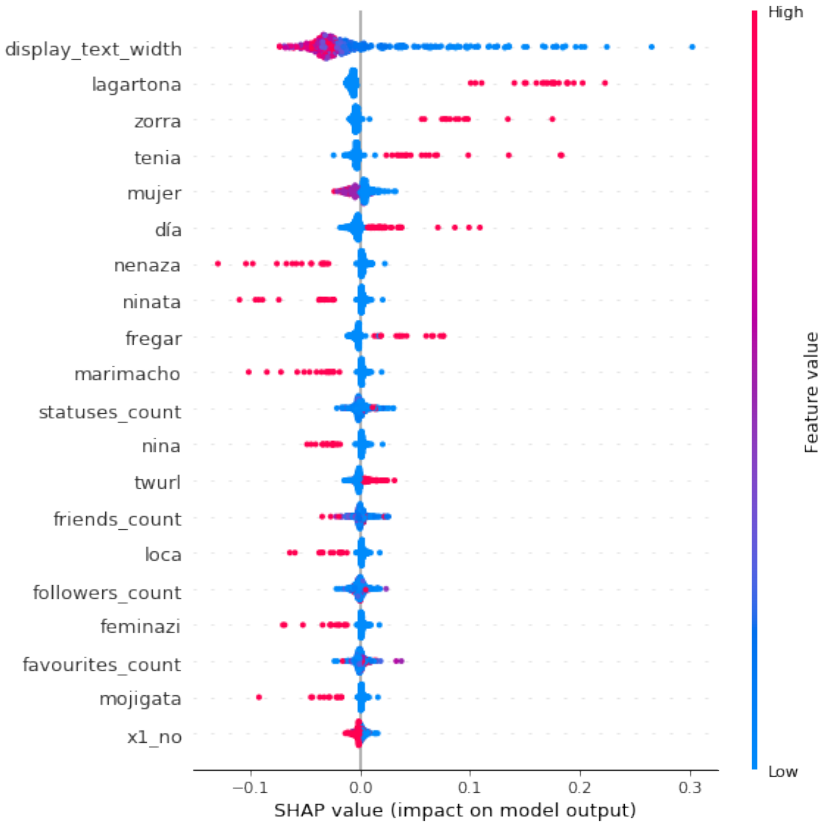

Figura 6.4: Valores SHAP para tweets dudosos

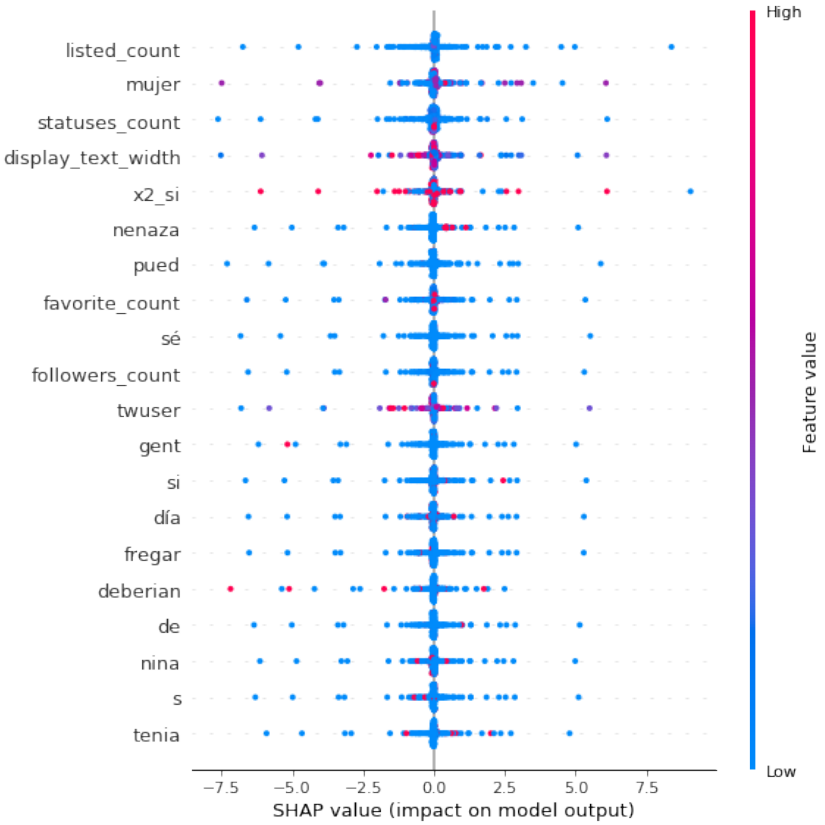

Figura 6.5: Valores SHAP para tweets machistas

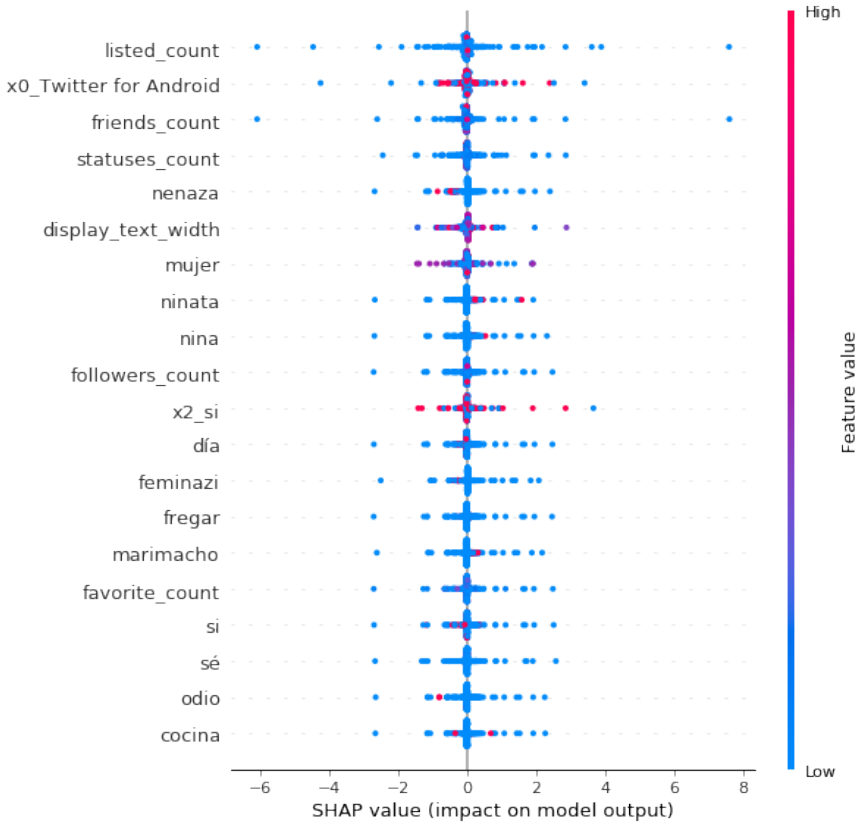

Figura 6.6: Valores SHAP para tweets no machistas

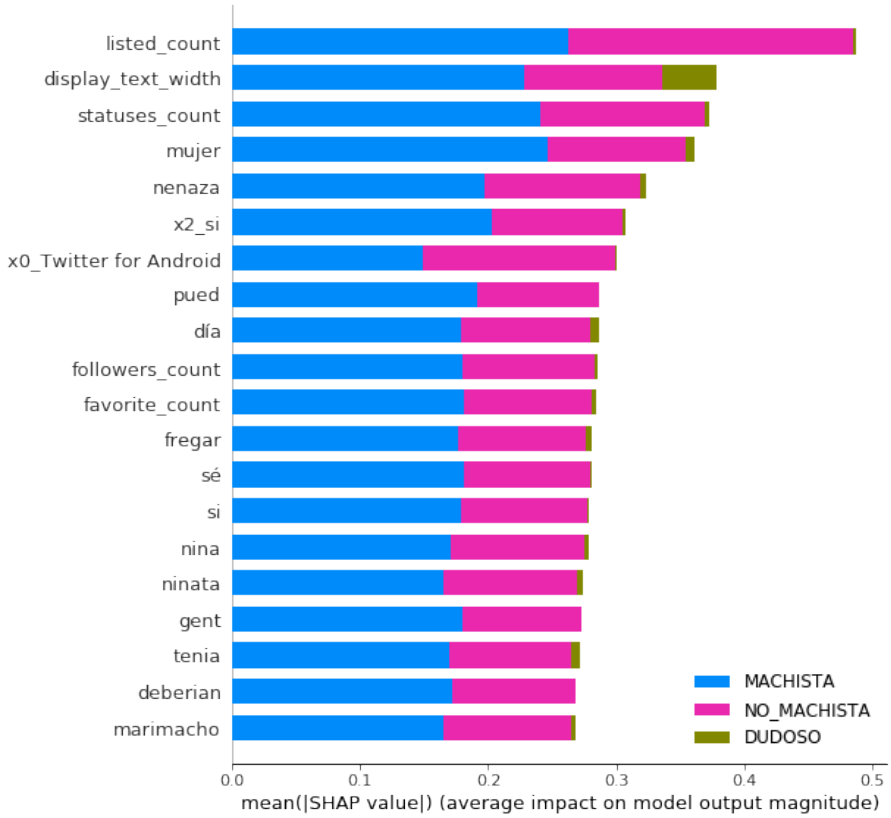

Figura 6.7: Impacto de los atributos

La tabla 6.1 muestra también ligeras diferencias entre los resultados obtenidos según el algoritmo de clasificación utilizado. En este experimento, RF y LR obtienen el mejor valor para dos de las cuatro métricas de calidad evaluadas, sin embargo, las diferencias con SVM no son demasiado amplias.

A priori, cabía esperar mejores resultados para el clasificador SVM pues ha sido aplicado de un modo exitoso en referencias previas para clasificación textual (Canós, 2018; Endang Wahyu Pamungkas y Patti, 2018). Este tipo de algoritmo suele funcionar mejor que otros como los árboles de decisión en problemas donde las matrices de datos son muy "dispersas" y el concepto de distancia entre los diferentes puntos adquiere mayor importancia, como ocurre al realizar la representación de texto mediante los atributos tf-idf. Sin embargo, en este problema en particular, solo se trabajan con 222 atributos en total antes de aplicar el algoritmo de clasificación, esto podría provocar que este tipo de técnica deje de tener la efectividad esperada.

En cuanto a los resultados obtenidos con el algoritmo RF, es importan-

RF	DUDOSO	MACHISTA	NO_MACHISTA
DUDOSO	70	32	80
MACHISTA	31	407	387
NO_MACHISTA	39	124	1350

Tabla 6.2: Matriz de confusión para una iteración de RF

te señalar que se ha conseguido unos valores elevados en la tasa de acierto y en precisión debido a la gran capacidad de la técnica para detectar los tweets no machistas. Esto se puede confirmar mediante la matriz de confusión representada en la tabla 6.2 donde se observa que el valor de clase "NO_MACHISTA" es claramente el que mejor se clasifica mediante esta técnica.

La elevada precisión del algoritmo RF para los tweets no machistas se debe al efecto del desbalanceo de la clase. Como ya se introdujo en los capítulos anteriores, y debido a la naturaleza del problema, existen una gran predominancia de los tweets no machistas en el corpus etiquetado manualmente. Esto provoca que algoritmos como los árboles de decisión, puedan tener un sesgo en sus predicciones hacia la clase predominante (Liu y Chawla,). Otro tipo de técnicas como SVM pueden alcanzar mejores rendimientos en este tipo de problemas (Ustuner, 2016). Pese a esto, RF no se aleja demasiado en otras métricas como F1 por el uso conjunto de numerosos árboles de decisión y que permite mejorar el rendimiento que se obtendría con un único árbol de decisión.

En cuanto a los algoritmos SVM y LR, los resultados muestran un comportamiento muy similar en ambos casos. Esto se debe principalmente al uso de un "kernel" lineal para el algoritmo SVM, lo que provoca que la frontera de decisión entre clases sea lineal, al igual que ocurre con el algoritmo LR. En ambos casos, los resultados muestran una menor capacidad para detectar los tweets no machistas pero se mejora el equilibrio entre el resto de valores de la clase. En la tabla 6.3, se puede observar cómo estos modelos mejoran notablemente la detección de tweets dudosos y machistas. Esto se debe, probablemente, a un mejor funcionamiento de estos métodos ante clases desbalanceadas.

Es importante también destacar que en cualquiera de los dos modelos de clasificación, la principal fuente de errores proviene de la detección de tweets machistas. En la tabla 6.3, se puede observar cómo en esta clase

LR	DUDOSO	MACHISTA	NO_MACHISTA
DUDOSO	116	36	36
MACHISTA	96	479	247
NO_MACHISTA	91	250	1169

Tabla 6.3: Matriz de confusión para una iteración de LR

se produce la mayor parte de los errores donde se clasifican erróneamente más del el 30 % de los tweets machistas. En concreto, la mayor parte de los errores se produce en la clasificación de tweets de este tipo como tweets no machistas. Para estudiar esta fuente de error, se ha realizado un estudio en profundidad para las clasificaciones producidas con este error. Por ejemplo, el tweet "*@CopitoDeSnow_ Ahora es cuando digo "no está mal para ser mujer""* se ha clasificado erroneamente como no machista por el sistema. En la figura 6.8 se representa la contribución de los atributos más importantes para la clasificación ofrecida por el sistema. En este caso, la existencia de los términos "ser" y "digo" además de la inexistencia del término "nenaza", aumentan la probabilidad de que el sistema clasifique el tweet como no machista mientras que los atributos que disminuyen la probabilidad de esta clasificación serían el tamaño del tweet y la publicación de éste desde un dispositivo Android. Como se puede observar, no se detectan términos textuales que reduzcan la clasificación de no machista y, por ello, el tweet se clasifica erróneamente. Para este ejemplo, el sistema falla en la detección del grupo "mal para ser mujer" como una expresión que claramente conlleva una actitud machista.

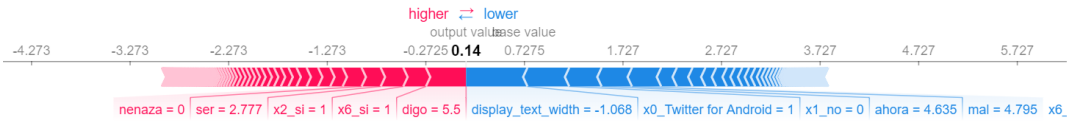

Figura 6.8: Valores SHAP para ejemplo 1

Otro ejemplo de error de este tipo sería el tweet "*La negativa a q el aborto sea legal, lleva implícito el elemento castigo: si follaste, pare.Porque las mujeres, no deberían follar tan alegremente, ni que fueran hombres*". De nuevo, se puede observar en la figura 6.9 como la inexistencia del término "nenaza" aumenta la probabilidad de que el tweet sea no machista. Esto se

produce porque, en general, todas las instancias que contenían este término se han considerado machistas y, por tanto, el modelo está sesgado hacia los valores de clase que se le presentaron durante el entrenamiento. Es interesante remarcar, que el término "deberían" contribuye a reducir la probabilidad de que el tweet se considere no machista, sin embargo, no lo suficiente para evitar el error del clasificador. Este término se repite mucho en el corpus para dar opiniones machistas acerca de cómo deben de comportarse las mujeres o colectivos feministas.

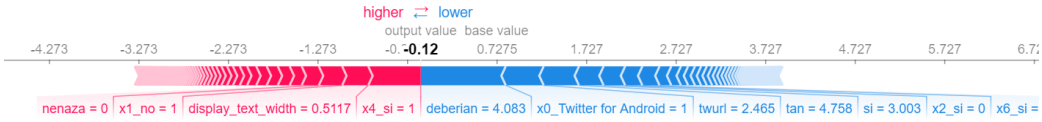

Figura 6.9: Valores SHAP para ejemplo 2

De nuevo, el sistema produce un error en la clasificación del tweet "*@damita2808 @berege7 @Mariagtriana Y los ojos? Uff demasiado dureza en la mirada para ser chica...no?*". Al igual que ocurre en los ejemplos anteriores, la inexistencia del término "nenaza" contribuye a clasificar el tweet como no machista al mismo tiempo que el sistema falla al detectar algunos de los términos de la expresión "demasiado para ser chica". Esto se produce al igual que en los casos anteriores, por la limitación de la aproximación basada en unigramas que utiliza frecuencias de términos para representar el texto a clasificar. Esto, supone una desventaja en corpus como el que se estudia en este trabajo, donde el vocabulario es muy heterogéneo y las expresiones o grupos de palabras clave no se repiten lo suficiente durante la etapa de entrenamiento del sistema. En este corpus, se trabaja con 222 atributos, lo que indica que se opera con un vocabulario limitado a menos de 200 términos que se repiten en al menos el 1 % de los tweets. Se han realizado pruebas reduciendo este umbral hasta un 0.3 % para aumentar el número de términos del vocabulario pero los resultados no han mejorado. Este mismo efecto se produce en el tweet "*Las mujeres no deberían usar sostén*" (figura 6.11) donde los términos textuales no tienen un peso relevante al realizar la clasificación. Por tanto, por un lado, hay que tener en cuenta que, pese a realizar el corpus mediante la búsqueda de unos términos bien definidos, la ambigüedad y riqueza del lenguaje provoca que el contexto en el que se usan sea muy diverso, más aún al tratarse de una red social. Por otro lado, el

método aplicado puede tener limitaciones importantes para trabajar en este contexto pues está basado en frecuencias de términos y empeora en entornos donde el uso de éstos no es homogéneo.

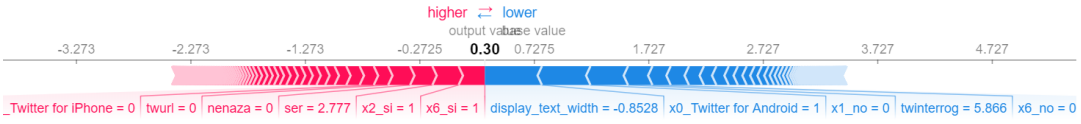

Figura 6.10: Valores SHAP para ejemplo 3

Figura 6.11: Valores SHAP para ejemplo 4

Otro efecto relevante se produce en los siguientes tweets clasificados erróneamente como no machistas por el sistema:

@EsppeonzAguirre "Por ejemplo,@IrantzuVarela hace unos sketches poniéndose unas barbas postizas despeinadas. Es lo que antes se llamaba un marimacho. El igualitarismo ha hecho mucho daño. Uno tiene que mandar y otro obedecer acríticamente

Buscad mujeres con valores. No prestéis atención a ninguna niñata feminista. No os relacionéis con ellas, salvo para educarlas. No dejemos que nos coma el NOM.

En ambos casos, se comparten tweets de una longitud considerable donde no existe ningún término individual que indique una actitud machista sino un conjunto de ellos que denotan este tipo de comportamientos. Por esto, se puede observar en las figuras 6.12 y 6.13 como los términos individuales más importantes en el corpus como "marimacho", "niñata" o "feminista" presentes en los tweets "compiten" para realizar la clasificación. La limitación de la solución propuesta para tener en cuenta el contexto provoca el error en estos casos. Para intentar reducir este efecto y considerar grupos de palabras, se han realizado pruebas incluyendo atributos tf-idf basados en bigramas sin éxito en los resultados.

Figura 6.12: Valores SHAP para ejemplo 5

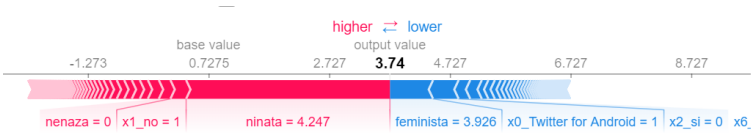

Figura 6.13: Valores SHAP para ejemplo 6

Otra fuente importante de errores que se ha detectado son las faltas de ortografía cometidas por los usuarios al utilizar la red social. Pese a que se ha utilizado un léxico para realizar un mapeo de las faltas más comunes, estos errores hacen que el sistema no pueda representar correctamente los términos como atributos del conjunto de datos.

6.3. Resultados experimento 2

La tabla 6.2 muestra los resultados promedio de exactitud (*Accuracy*), medida F1, cobertura (*Recall*) y precisión. Para este segundo experimento, el algoritmo RF consigue de nuevo la mejor tasa de acierto y precisión, mientras que en este caso es el algoritmo SVM el que obtiene los mejores resultados en cuanto a la medida F1 y el *recall*. Sin embargo, es importante destacar que el comportamiento de los tres clasificadores es bastante similar en líneas generales.

En relación con el método de evaluación propuesto en el experimento 1, se puede observar una pequeña mejoraría de un punto para la tasa de acierto, la precisión y la medida F1. Sin embargo, se mantienen muy similares los comportamientos de cada algoritmo de predicción. Este efecto, se puede deber a que los parámetros que obtenemos del método 2 no son tan buenos por los pocos datos que utilizamos (solo 25 % del corpus) y modifican muy poco el comportamiento de éstos por defecto.

Se ha observado mediante distintos repartos de datos en las etapas de búsqueda de parámetros y evaluación, que aumentar la cantidad de información disponible para la validación cruzada, mejoraba siempre los resultados

	Accuracy	**F1**	**Recall**	**Precision**
Baseline (tf-idf)	0.69	0.58	0.56	0.62
Baseline	0.61	0.2	0.3	0.24
LR	0.72	**0.63**	0.62	0.65
RF	**0.73**	0.61	0.58	**0.68**
SVM	0.72	**0.63**	**0.64**	0.63

Tabla 6.4: Resultados experimento 2

independientemente del valor de los parámetros de entrada de los algoritmos. Es decir, añadir más información a la etapa de búsqueda de parámetros no ha resultado tan efectivo como reservar la mayor parte de la información para entrenar y evaluar el sistema final.

En relación a la comparación de los resultados obtenidos por el método con las dos líneas bases, de nuevo, se mantienen las mismas tasas de mejoras que en el experimento anterior.

En cuanto a las diferencias entre los modelos empleados, se observan las mismas relaciones y los efectos producidos son equivalentes al experimento anterior. Si se observan detenidamente los resultados, se observará como las medidas de calidad que más aumentan respecto del experimento 1 serían la tasa de acierto y la precisión (la medida F1 aumenta como consecuencia de la precisión). Esto se produce porque, en realidad, este experimento detecta algunos casos más de tweets no machistas, probablemente porque se presenta más cantidad de información en el entrenamiento. Por tanto, la detección de tweets machistas seguiría siendo la principal fuente de error del sistema. Se ha comprobado, mediante un estudio de los errores, que se dan los mismos efectos descritos en el experimento 1.

6.4. Efecto del desbalanceo de la clase

Los resultados presentados hasta este punto, demuestran que el problema principal del sistema desarrollado es el sesgo producido hacia el valor de la clase mayoritaria.

Para evaluar el impacto del desbalanceo de la clase hacia tweets no machistas e intentar paliar este efecto, se ha realizado un experimento mediante un muestreo del corpus. Como se puede observar en la tabla 4.7, el valor de la clase “DUDOSO” sería el que cuenta con menos instancias dentro del cor-

	Accuracy	**F1**	**Recall**	**Precision**
LR	0.61	0.61	0.61	0.62
RF	0.68	0.68	0.68	0.68
SVM	0.63	0.63	0.66	0.63

Tabla 6.5: Resultados experimento 2 con balanceo de clases

pus. Por esto, se han realizado una prueba realizando un muestreo aleatorio de las otras dos clases para igualar la cantidad de registros entre las 3 clases que componen el corpus.

En la tabla 6.5 se pueden observar los resultados para cada uno de los métodos empleados en el presente trabajo, evaluados mediante una validación cruzada con 10 grupos y los parámetros por defecto (mismas condiciones que el experimento 2).

Como se puede observar, los resultados empeoran generalmente para la tas pese a que existe un equilibrio total en el número de registros por clase. Los algoritmos LR y SVM son los que más empeoran en este caso, una posible causa podría ser la escasa información con la que cuenta el sistema en este caso, 267 instancias por clase.

El único algoritmo que consigue mejorar alguna de sus medidas de calidad es RF. En este caso, se mejora notablemente la medida F1 y la cobertura, lo que indica, que en condiciones de equilibrio entre las clases, este algoritmo mejora notablemente la detección del resto de valores de la clase del corpus.

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

En este trabajo fin de máster se ha presentado un nuevo sistema que pretende resolver el problema de detección del machismo en redes sociales. El sistema realiza un análisis supervisado y emplea distintos atributos unificados para, posteriormente, hacer uso de un algoritmo de aprendizaje de máquina que permita detectar expresiones y actitudes machistas en las redes sociales. Para cada atributo, se aplican diferentes métodos de preprocesado como la normalización de términos mediante un léxico de *slang* o la sustitución de emoticonos. Tras esto, se unifican los distintos tipos de atributos procesados en un conjunto de datos común que será la entrada de la última fase. En la última etapa, se emplean algoritmos de clasificación supervisada con la intención de obtener un modelo predictivo capaz de detectar las señales textuales que expresan lenguaje machista.

Se ha presentado MeTwo, un corpus desarrollado en el presenta trabajo, que ha sido utilizado para entrenar el sistema de clasificación descrito. El corpus aportado por este trabajo cuenta 3600 mensajes etiquetados por 3 anotadores para la tarea de detección del machismo. Hasta la fecha, y en conocimiento del autor de este trabajo, se trata del corpus etiquetado más extenso disponible en castellano para la tarea de detección del machismo.

Asimismo, en el capítulo 2, se ha realizado una extensa revisión del estado del arte en la detección del lenguaje ofensivo, tanto sexista como de

otro tipo, siendo posible detectar las principales limitaciones y problemas por resolver en la actualidad. La principal conclusión extraída es que se trata de una línea investigación muy actual y de creciente popularidad. Se trata de una tarea compleja, en la que aún no se ha investigado suficientemente como demuestran los escasos trabajos que tratan la detección de lenguaje machista en castellano.

Uno de los principales objetivos ha sido la detección de señales textuales que expresan lenguaje machista en castellano. En conocimiento del autor, solo una serie de publicaciones relacionadas con la competición (E. Fersini y Anzovino, 2018b) han abordado la problemática desde un punto de vista práctico sin una investigación para la composición del corpus utilizado en la etapa de entrenamiento. En este trabajo, se ha experimentado con diferentes términos y conceptos para componer un corpus que permita desarrollar un sistema específico para la tarea y los datos recolectados. De este modo, a lo largo del trabajo, se presenta el ciclo completo para la recolección de datos, preprocesamiento y construcción del sistema de clasificación para el lenguaje.

Por otro lado, se ha realizado una evaluación exhaustiva para determinar la viabilidad del modelo propuesto y valorar el desempeño del sistema desarrollado. Para ello, se ha utilizado el corpus MeTwo con distintos experimentos de evaluación del sistema de clasificación. Asimismo, se ha estudiado el efecto del desbalanceo de la clase producida por la naturaleza del problema tratado. Los resultados obtenidos con este método son prometedores y se consigue clasificar las instancias del conjunto de test con más de un 70 % de acierto. Pese a los resultados obtenidos, se han detectado distintas limitaciones debido a las técnicas utilizadas. Uno de los problemas detectados ha sido un excesivo sesgo hacia ciertos términos independientemente del contexto en el que se utilicen. Por ejemplo, el término “nenaza” tiene gran importancia en el clasificador para la detección de mensajes machistas. Asimismo, otra limitación está relacionada con los términos que no se repiten a lo largo del corpus pero presentan actitudes machistas. Estos dos efectos se producen por la representación textual basada en frecuencias de términos y que empeora en entornos donde el uso de éstos no es homogéneo.

Otro inconveniente del sistema desarrollado es la excesiva dependencia a la presencia de términos sin tener en cuenta el contexto. Debido al método de representación textual utilizado, el orden en el que aparecen las palabras

dentro del mensaje no es importante y, por ello, el método falla al detectar el machismo en textos donde no haya ningún término que, por si mismo, conlleve machismo en un gran porcentaje de ocasiones.

7.2. Trabajo futuro

El proyecto realizado hasta el momento, presentado como trabajo fin de máster, será el punto de partida para la tesis doctoral que se pretende desarrollar en los próximos años. Aunque los resultados obtenidos son prometedores, se han definido posibles mejoras así como líneas de investigación futuras.

En primer lugar, el método propuesto no realiza ninguna diferencia en el tipo de machismo empleado en el texto. Una posible línea de trabajo sería profundizar en los tipos de machismo existentes y realizar una ampliación del etiquetado en el corpus MeTwo. En la guía de anotación (ver Apéndice) se realiza una primera exploración en este hecho para presentar distintos tipos de machismo, como, por ejemplo, la sexualización, el descrédito o la dominancia.

Por otra parte, tal y como se ha podido comprobar de un modo experimental, sería interesante la ampliación a distintos fuentes de dato como periódicos, webs u otras redes sociales. En algunos experimentos, como en el apartado 6.4, se ha concluido que la bondad de los resultados sería mayor si se dispusiera de una mayor cantidad de información.

Asimismo, sería recomendable profundizar en las técnicas para la clasificación automática, por ejemplo, técnicas basadas en redes neuronales que alcanzan unos resultados prometedores en tareas relacionadas con el procesamiento de texto (Devlin, 2019). Este tipo de técnicas, junto con representaciones textuales basadas en *word embedding*, podrían ayudar a paliar algunas de las limitaciones detectadas en el sistema propuesto como, por ejemplo, la consideración del contexto en el que se utilizan los términos machistas.

Otra línea de trabajo futura podría ser la adaptación del sistema para el idioma inglés. Debido al diseño de este método, se podría adaptar fácilmente para ser capaz de detectar expresiones machistas en otros idiomas.

Sería recomendable la creación de un léxico machista específico para el español que permita añadir atributos en el entrenamiento del sistema pa-

ra detectar aquellos términos que más se repitan en los mensajes machistas. Este tipo de enfoque ha demostrado su efectividad en trabajos similares (Endang Wahyu Pamungkas y Patti, 2018) y se podrían implementar mediante léxicos disponibles en la bibliografía (De Mauro, 2016).

Finalmente, como salida del sistema, sería interesante la creación de un portal web que permita, mediante distintos tipos de visualizaciones, explorar los tipos de machismos en España, Europa, o el mundo. Podría servir para detectar y prevenir los tipos de machismos en cada zona geográfica, los términos más empleados y los medios más afines para transmitir este tipo de mensajes.

Bibliografía

Bibliografía

[Twi2017] 2017. https://verne.elpais.com/verne/2017/04/17/articulo/1492412360-237040.html.

[Aranbarri2014] Aranbarri, Garazi Urdangarin. 2014. Cosificación de las adolescentes en las redes sociales digitales, pág 43. Master's thesis, Universidad del País Vasco.

[Atlantic2016] Atlantic, The. 2016. https://www.theatlantic.com/technology/archive/2016/07/twitter-swings-the-mighty-ban-hammer/492209/.

[Aurangzeb Khan2010] Aurangzeb Khan, Baharum Baharudin. 2010. A review of machine learning algorithms for text-documents classification.

[Bernal2017] Bernal, Ana Isabel. 2017. https://verne.elpais.com/verne/2017/11/25/articulo/1511634518358211.html.

[Breiman2001] Breiman, Leo. 2001. Random forests.

[Canós2018] Canós, Jose Sebastián. 2018. Misogyny identification through svm at ibereval 2018. En *IberEval 2018*.

[Carreras et al.2004] Carreras, Xavier, Isaac Chao, Lluís Padró, y Muntsa Padró. 2004. Freeling: An open-source suite of language analyzers. En *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 04)*.

[Cohen1960] Cohen, Jacob. 1960. A coefficient of agreement for nominal scales. En *U.S. Geological Survey*.

[Cortes1995] Cortes, Corinna. 1995. Support-vector networks.

[De Mauro2016] De Mauro, T. 2016. Le parole per ferire. En *Internazionale (2016)*.

[Deerwester1990] Deerwester, Scott. 1990. Indexing by latent semantic analysis.

[Despoina Chatzakouy2017] Despoina Chatzakouy, Nicolas Kourtellisz, Jeremy Blackburnz Emiliano De Cristofaro Gianluca Stringhini Athena Vakaliy. 2017. Mean birds: Detecting aggression and bullying on twitter. En *WebSci*.

[Devlin2019] Devlin, Jacob. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

[Duggan2017] Duggan, M. 2017. Online harassment 2017. En *Pew Research Center, July 2017*.

[E. Fersini y Anzovino2018a] E. Fersini, P. Rosso y M. Anzovino. 2018a. Overview of the task on automatic misogyny identication at ibereval 2018. En *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018)*.

[E. Fersini y Anzovino2018b] E. Fersini, P. Rosso y M. Anzovino. 2018b. Overview of the task on automatic misogyny identification at ibereval 2018. En *IberEval 2018*.

[Endang Wahyu Pamungkas y Patti2018] Endang Wahyu Pamungkas, Alessandra Teresa Cignarella, Valerio Basile y Viviana Patti. 2018. Exploiting lexical knowledge for detecting misogyny in english and spanish tweets. En *IberEval2018*.

[F. Pedregosa2011] F. Pedregosa, G. Varoquaux, A. Gramfort. 2011. Scikit-learn: Machine learning in python. En *JMLR*.

[Fresno2006] Fresno, V. 2006. Representación autocontenida de documentos html: una propuesta basada en combinaciones heurísticas de criterios.

[Fulper y Rowe2014] Fulper, Rachael, Giovanni Luca Ciampaglia Emilio Ferrara Y. Ahn Alessandro Flammini Filippo Menczer Bryce Lewis y Kehontas Rowe. 2014. Misogynistic language on twitter and sexual violence. En *In Proceedings of the ACM Web Science Workshop on Computational Approaches to Social Modeling (ChASM)*.

[Georgios K. y Langseth2018] Georgios K., Heri R. y Helge Langseth. 2018. Detecting oensive language in tweets using deep learning. En *Department of Computer Science Norwegian University of Science and Technology.*

[Giraldo1972] Giraldo, Octavio. 1972. El machismo como fenómeno psicocultural. En *Revista Latinoamericana de Psicología.*

[Goenaga y Perez2018] Goenaga, A. Atutxa, K. Gojenola A. Casillas A. Daz de Ilarraza N. Ezeiza M. Oronoz A. Perez y O. Perez. 2018. Automatic misogyny identification using neural networks. En *IberEval 2018.*

[Goodfellow, Bengio, y Courville2016] Goodfellow, Ian, Yoshua Bengio, y Aaron Courville. 2016. *Deep Learning.* MIT Press. http://www.deeplearningbook.org.

[Group2015] Group, Stanford NLP. 2015.

[Guang Xiang2012] Guang Xiang, Bin Fan, Ling Wang Jason I. Hong Carolyn P. Rose. 2012. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. En *CIKM, 2012, Maui, HI, USA.*

[Guo2003] Guo, Gongde. 2003. Knn model-based approach in classification.

[Han Liu y Cocea2018] Han Liu, Fatima Chiroma y Mihaela Cocea. 2018. Identification and classication of misogynoustweets using multi-classier fusion. En *IberEval 2018.*

[Helena Gómez-Adorno2016] Helena Gómez-Adorno, Ilia Markov, Grigori Sidorov. 2016. Compilación de un lexicón de redes sociales para la identificación de perfiles de autor. En *Centro de Investigación en Computación, México.*

[Honnibal y Montani2017] Honnibal, Matthew y Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing.

[https://amiibereval2018.wordpress.com/2018] https://amiibereval2018.wordpress.com/. 2018. Automatic misogyny identification, ibereval 2018.

[International2017] International, Amnesty. 2017. Toxic twitter - a toxic place for women. En *Amnesty International Research.*

[Jurafsky2009] Jurafsky, Daniel. 2009. *Speech and Language Processing.* Prentice-Hall.

[Jurafsky y Coccaro1998] Jurafsky, Daniel y Noah Coccaro. 1998. Automatic detection of discourse structure for speech recognition and understanding.

[Kearney2018] Kearney, Michael W., 2018. *rtweet: Collecting Twitter Data.* R package version 0.6.7.

[Landauer y Dumais1997] Landauer y Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge.

[Liddy2001] Liddy, Elizabeth D. 2001. Natural language processing.

[Liu y Chawla] Liu, Wei y Sanjay Chawla. A robust decision tree algorithm for imbalanced data sets.

[Maria Anzovino y Rosso2018] Maria Anzovino, Elisabetta Fersini y Paolo Rosso. 2018. Automatic identification and classification of misogynistic language on twitter. En *Springer International Publishing AG, part of Springer Nature 2018.*

[Mark Hepple] Mark Hepple, Neil Ireso. Nlp-enhanced content filtering within the poesia project.

[Nina-Alcocer2018] Nina-Alcocer, Victor. 2018. Ami at ibereval2018 automatic misogyny identification in spanish and english tweets. En *IberEval 2018.*

[NLTK2018] NLTK. 2018. https://www.nltk.org/.

[Ovoputi2013] Ovoputi, Olutobi. 2013. Improved part-of-speech tagging for online conversational text with word clusters.

[Park y Fung2017] Park, Ji Ho y Pascale Fung. 2017. One-step and two-step classification for abusive language detection on twitter. En *Proceedings of the First Workshop on Abusive Language Online.*

[Pinkesh Badjatiya2017] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. En *2017 International World Wide Web Conference Committee (IW3C2)*.

[Porter1980] Porter, M. 1980. An algorithm for suffix stripping.

[Quartz2017] Quartz. 2017. https://qz.com/1101455/facebook-fb-is-hiring-more-people-to-moderate-content-than-twitter-twtr-has-at-its-entire-company/.

[RAEa] RAE, Definición Machismo. http://dle.rae.es/srv/search?m=30&w=machismo.

[RAEb] RAE, Definición Misoginia. http://lema.rae.es/dpd/srv/search?key=misoginia.

[Rehder1998] Rehder, B. 1998. Using latent semantic analysis to assess knowledge: Some technical considerations.

[Research2018] Research, Synergy. 2018. https://www.srgresearch.com/articles/aws-leading-public-cloud-market-all-major-regions.

[Resham Ahluwalia y Cock2018] Resham Ahluwalia, Evgeniia Shcherbinina, Edward Callow Anderson Nascimento1 y Martine De Cock. 2018. Detecting misogynous tweets. En *IberEval 2018*.

[Roesslein2009] Roesslein, Joshua. 2009. Tweepy.

[Russell y Norvig2002] Russell, S. y P. Norvig. 2002. *Artificial Intelligence: A Modern Approach*. Prentice Hall.

[Shushkevich y Cardiff2018] Shushkevich, Elena y John Cardiff. 2018. Classifying misogynistic tweets using a blended model: The ami shared task in ibereval 2018. En *IberEval 2018*.

[Simona Frenda y y Gomez2018] Simona Frenda, Bilal Ghanem y Manuel Montes y Gomez. 2018. Exploration of misogyny in spanish and english tweets. En *IberEval 2018*.

[Steven Bird y Loper2009] Steven Bird, Ewan Klein y Edward Loper. 2009. *Natural Language Processing with Python*. O'REILLY.

[Strumbelj y Kononenko2014] Strumbelj, Erik y Igor Kononenko. 2014. Explaining prediction models and individual predictions with feature contributions. En *Knowledge and information systems 41.3*.

[Tan2019] Tan, Mingxing. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks.

[Thomas Davidson2017] Thomas Davidson, Dana Warmsley, Michael Macy Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. En *ICWSM 2017*.

[Twitter2018a] Twitter. 2018a. https://developer.twitter.com/en/docs/api-reference-index.html.

[Twitter2018b] Twitter. 2018b. https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy.

[Ustuner2016] Ustuner, M. 2016. Balanced vs imbalanced training data: Classifying rapideye data with support vector machines. En *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLI-B7*.

[Wang2010] Wang, Alex Hai. 2010. Spam detection in twitter.

[Waseem2016a] Waseem, Zeerak. 2016a. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. En *Proceedings of 2016 EMNLP Workshop on Natural Language Processing and Computational Science.*

[Waseem2016b] Waseem, Zeerak. 2016b. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. En *Proceedings of NAACL-HLT 2016*, páginas 33–41.

[Watanabe2018] Watanabe, Hajime. 2018. Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. En *2018 IEEE*.

[Zimmerman2018] Zimmerman, Steven. 2018. Improving hate speech detection with deep learning ensembles. En *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Apéndice A

Guía de anotación

Trabajo Fin de Máster – 2018/2019

Detección del machismo en redes Sociales

Guía de anotación del corpus

Francisco Miguel Rodríguez

Descripción de la tarea

En el presente trabajo se propone la detección del lenguaje machista en español en la red de microblogging Twitter.

Para ello, la tarea principal es clasificar contenido procedente de esta red social en tres categorías distintas: machista, no machista y dudoso. A continuación, se definen las etiquetas propuestas:

- **MACHISTA**: tweets que contienen connotaciones machistas/ofensivas hacia las mujeres. Los tweets que pertenecen a esta categoría deben de contener actitudes discriminatorias o hacia las mujeres por su sexo. Ejemplo:

  > *"@EmanuelGPA Lo irónico es que lo dice una mujer, que "naturalmente" debería callarse y dedicarse a la cocina, limpiar y criar hijos"*

  En este ejemplo, se infravalora a la mujer de un modo expreso.

- **NO_MACHISTA**: tweets que no contienen connotaciones machistas. Ejemplo:

  > *"¿Por qué cuando ven a una chica con pelo corto piensan que es bi, torta o marimacho? ...Capaz es solo un look o que el pelo largo le da mucho calor... Y si es...¿Cual es el problema?"*

  Dentro de esta categoría se clasifican también mensajes xenófobos y ofensivos en general pero que no discriminan a las mujeres por su sexo. Ejemplo:

  > "*@kenia773 @LuisCarlos POR CIERTO, EN TU FOTO DE PERFIL SE PUEDE OBSERVAR QUE ERES BASTANTE VARONIL, ASÍ QUE SI NO ERES MARIMACHO, EMPIEZA A SERLO*".

- **DUDOSO**: tweets que, dependiendo del contexto, no presente en el tweet, podrían ser machistas (si el fragmento ofensivo se refiriese a mujeres). Ejemplo:

  "@hazteoir @PSOE Más vale que se marche a fregar !".

  Para poder considerar el tweet machista es necesario una referencia expresa a las mujeres, pese a que en el texto se pueda realizar una comparación con una mujer mediante una expresión machista.

Ejemplos para tweets machistas

Para que un tweet se considere machista, debe de representar una ofensa hacia las mujeres. De este modo, es necesario una referencia textual hacia éstas como en los siguientes casos:

N	Tweet	Clase	Argumento
1	"Pelea como una mujer"	MACHISTA	Dominancia/inferioridad física
2	"Esa chabona me da ASCO, esto para mí sería una 'feminazi', esta chica se pasó, de feminista no tiene nada, las feministas que conozco no son nada como ella, es una vergüenza que se denomine feminista. https://t.co/KgO5ifEDvc"	MACHISTA	Descredito/feminismo como algo negativo
3	"Hay cierto tipo de feminazi que me da cierto tipo de asco."	MACHISTA	Descredito
4	"Nada como una mujer después de una pelea."	MACHISTA	Sexualización
5	"-Dime algo que me haga sentir como una mujer.-Tienes razón."	MACHISTA	Descredito
6	"@MilloLaguneroSt @clarincom Igual capo yo que vos cambio de target; de las feministas 4 de 5 son incogibles https://t.co/RRrcBfuzti"	MACHISTA	Sexualización
7	"Los comunistas comen niños y las feministas radicales y materialistas apalean a gente trans."	MACHISTA	Descredito/feminismo como algo negativo
8	"Para reflexionar sobre feminismoPara las feministas que aún toleran la reflexión y el debate. https://t.co/s0w1E0LGz8"	MACHISTA	Descredito/feminismo como algo negativo
9	"@VictoriaSonntag @Aldana25Gomez Claro que sí, pero sabemos que la mayoría de las feministas no se molestan en arreglarse."	MACHISTA	Descredito/feminismo como algo negativo
10	"@cantina1138 No se le puede pedir mucho a alguien que hace vídeos diciendo estupideces como \Kathleen Kennedy ha utilizado Los Últimos Jedi como panfleto de su ideología feminazi\"""	MACHISTA	Descredito
11	"@escuelazurda Vaaaaamos a limpiar los cerebritos y embretarles desde chicos haciendoles ver algo que yo quiero.. Que vergüenza..Si fuera el colegio de mis hijos en ese momento los estaria retirando..Padres cuiden de sus hijos e hijas que les puede salir una feminazi o peor.. un socialista"	MACHISTA	Descredito

12	“Que asco me dá ver y escuchar a la Barón!Es un marimacho con pelo largo!Que grasa y ordinaria! Mujer Libre se dice? Este especimen habla del género? Fush!Menos mal que las mujeres Femeninas no tenemos nada que ver con las Feminazis Aborteras! @AngeldebritoOk @LosAngeles_ok”	MACHISTA	Descredito
13	“@Camividela3 Soy libre para dar mi opinión. Y si, obvio que si vos y la otra zorra no vienen a visitar”	MACHISTA	Sexualización/ descredito
14	“Ella dijo chau" Y ori dijo "menos mal que hay viento asi no se nos pega tu olor a zorra" JAJAJAJAJAJJA ame"”	MACHISTA	Sexualización/ descredito
15	“Señora si ustd tiene más de 40 años no use calzas, a menos que tenga el cuerpo de Pampita, nosotros no somos quienes para ver el hachazo en el barro ese... vieja zorra!”	MACHISTA	Sexualización
16	“@PrimaveraCNpe @elmundoes .@sanchezcastejon vas a condenar este crimen? #feminazi algo que decir? #silenciados @DocuSilenciados @CeltiusAbogados #UngaUngaArmy @ArmyUnga	MACHISTA	Descredito
17	“@pepinodeprimido @elespanolcom Es bueno q se publique para q no lo pueda negar la feminazi del \mitú\"." ”	MACHISTA	Descredito
18	“@ecuanimusjustus @ClaudiaLopez Y a todas estas, este marimacho ya habrá perdido la virginidad?”	MACHISTA	Sexualización
19	@TheNotoriousMMA @ProperWhiskey vas a perder,tu eres un hijo de puta,tu padre un borracho,tu mujer una zorra y tu madre una puta https://t.co/Ll1IeLGeof	MACHISTA	Sexualización/ descredito
20	@JuliMorinigoo @MacaaRepetto Rompe hogares encima o como conoce normalmente zorra	MACHISTA	Sexualización/ descredito
21	@Code_Bad_Kid Estaba demasiado frustrado como para seguirle el juego, y si le soltaba una hostia tal y como estaba no sería un simple colegueo agresivo.No. Esa zorra sigue jugando conmigo... pero voy a encontrarla por mi propia mano.Y se va. Literalmente va a hacerse con toda la luz de »	MACHISTA	Dominancia
22	@idntrustanyone Es que de verdad.... ¡Puedes tener a cualquier puta chica que quieras! ¿Porqué una zorra como ella?Yo si, y hazme caso que la conozco más que tú.	MACHISTA	Sexualización
23	Ada Colau es una niñata idiota vendida al patriarcado que se enamora del primer hombre que le dice cuatro cosas bonitas.	MACHISTA	Descredito
24	@RadfemNoEsTerf El feminismo sufrirá amplio retroceso , por que ya se abrió el caballo de troya marimacho?	MACHISTA	Descredito/
25	@Apaltami66 @infobae Es una mina. Re marimacho, pero no creo que tenga próstata ni otro órgano masculino.	MACHISTA	Sexualización
26	Las opciones son: Sos mujer o sos feminazi	MACHISTA	Descredito
27	Es impresionante como una mujer enojada puede terminar cada frase asociándolo a lo que causó el enojo. Que capacidad de improvisación.	MACHISTA	Estereotipo
28	@Lidiaa_98 @CarmenEntrenas7 @pile_campos madre mia las niñas estas con mas modernidades venga a fregar ya https://t.co/H8R7btOjlT	MACHISTA	Estereotipo
29	@RocioMorla_ @Iennicet Y tu no sabes nada mas que repetir lo que te dicen tu circulo de amigas feministas. No tienes autocrítica. Y si la tienes dime una sola cosa en la que estés en contra del feminismo. Seguro que ninguna pues te llevas el pack completo pues no tienes sentido crítico alguno. A fregar	MACHISTA	Estereotipo/dominancia
30	Que puta hueva leer la palabra feminazi. Se dice FEMINISTA, así, sin que te tiemblen los huevos.	MACHISTA	Descredito/feminismo como algo negativo
31	@WharfRat_DE @nacho3810 No se pueden decir más tonterías en un twitt. Vaya un concepto más patético que tiene esta feminazi de las mujeres.	MACHISTA	Descredito

32	Y luego andan llorando de que no toman en cuenta su opinión y argumentos, ¿Qué necesidad de llevar a niñas pequeñas a estas cosas? A fuerza quieren imponer su forma de pensar ... Feminista sí, feminazi no https://t.co/TMt7GkKeDV	MACHISTA	Descredito
33	@wakuwakuweak Mujer tenías que ser. (?) Perdón no me mates lo siento mucho.	MACHISTA	Dominancia/estereotipo

Ejemplos para tweets no machistas:

	Tweet	Clase
1	"@bbcmundo El dia que esta persona pueda tener un bebe vaginalmente, amamantar, sentir y amar como una mujer sera una mujer; de lo contrario sera un varon frustrado arropado por los medios. No se trata de sentir o pretender, se trata de la naturaleza"	NO_MACHISTA
2	"@MaguiiApablaza Decimelo a mi amiga q me tengo q bancar al cavernicola del novio de mi vieja tirando comentarios como\si las feministas quieren andar en tetas por mi mejor\" AHHHH QUE GANAS DE ESCUPIRLE LA CARA" "	NO_MACHISTA
3	"@BostonRubio @sweet_karol_ @queperezamedas @indecepcion Víbora! Esta niñata!"	NO_MACHISTA
4	"@Pablo_Iglesias_ Niñata de la complu repitiendo de memoria el catecismo podemita. Apañaos habríamos estado en manos de estos defensores de narcodictaduras en plena crisis económica."	NO_MACHISTA
5	"@nataliaferviu @SHOKidding @MichelGondry @JimCarrey No eres animalista,eres una niñata intentando llamar la atención.Deja de hacer el ridículo y madura. De nada...."	NO_MACHISTA
6	"@AgusVBagna Escrache al marimacho (?) Jajajajajaja"	NO_MACHISTA
7	"@kenia773 @LuisCarlos POR CIERTO, EN TU FOTO DE PERFIL SE PUEDE OBSERVAR QUE ERES BASTANTE VARONIL, ASÍ QUE SI NO ERES MARIMACHO, EMPIEZA A SERLO"	NO_MACHISTA
8	Lo de esta es como lo de Rosa Diez pero en niñata.Indignante es que no pares de mentir y manipular.Aqui la unica que está mirando para otro lado, en concreto para Cuenca, eres tú, descerebrada. https://t.co/CY3KT1gp1K	NO_MACHISTA
9	@okdiario @Jacobo7elbobo Por decir esta barbaridad esta niñata cobra una pasta gansa. Al Congreso y Senado mandan a los mas tontos separatistas . Es increible cuanta tontería junta	NO_MACHISTA
10	@soledadMartn1 jajajja pensé lo.mismo cuando la oí.insultando según ella a su propio padre jajajaja pobre niñata para mentir hay que tener buena memoria	NO_MACHISTA
11	A ver xd ¿Tu te crees que va a venir una niñata medio tonta, medio choni de 15 o16 años a vacilarme a mi a mis 21 tacos? Y encima de buena mañana. Te quedan muchos Petit suisse que comerte para poder ni si quiera intentarlo.	NO_MACHISTA
12	"@Quejiquenom @Nicormg En la rama histórica iba de puta madre, en el "niñata pija que se gana el pan con lo que muchos tenemos que ocultar" ya me pierde."	NO_MACHISTA
13	Dimite el director de Cáritas que mandó "a fregar" a la concejala de Cambiemos Murcia https://t.co/0XmeHnVn7j"	NO_MACHISTA
14	@UnTioNormal_XD @LeticiaDolera Igual igual no, a ningún hombre les dicen vete a fregar... pero igual de mal me parece que insulten sean hombres o mujeres y eso no es buena noticia.	NO_MACHISTA

15	Y peor siendo mujer, que para la mayoría de esta gente no llegas ni a persona. No sabéis el esfuerzo extra que supone, que digas lo que digas, sea o no interesante, sólo dirán: melafo, fea, feminazi, gorda, a fregar...	NO_MACHISTA
16	Dejemos YA el argumento de que las #putas quieren serlo y lo de aquello de "que se pongan a fregar" Nadie nace puta, las hacen los tratantes, #proxenetas y #puteros... (Acción de Yolanda Domínguez @yodominguez contra la publicidad de prostitución) #DiaEuropeoContraLaTrata	NO_MACHISTA
17	@gfigarela @LLacen El blanco para casa y el negro también pero a fregar	NO_MACHISTA
18	@Rexmas1alterno @sealofseasons @chteresita Desde el punto que dice feminazi y que los trans son retrasados mentales está siendo discriminatorio.	NO_MACHISTA

Ejemplos para tweets dudosos:

	Tweet	Clase
1	“@AmikuGamer No estás en tus días :v ”	DUDOSO
2	“@Guarromantico_ ¿Qué pasó? En la mañana nos amabas, ¿andas en tus días?”	DUDOSO
3	“@ErikaPSUV @CiudadCCS @NicolasMaduro En vez de estar jalando bolas ponte a limpiar Caracas q esta llenas de huecos defecaciones por todas partes hedionda a orine abaratada de basuras bueno tendrás tu recompensa marimacho”	DUDOSO
4	“@poliok11 @LafliaOk @AngeldebritoOk @laufer4 @Flor_de_P Tal cual!!!!Es MARIMACHO!!!!”	DUDOSO
5	Creo que mi perra es una zorra. Me las hace todas. Le faltan las nueve colas nomás... encima tiene el mismo color que el kitsune de Yuko.	DUDOSO
6	@PirisDaMotta_ Estas en tus días parece no te gusta nada hacete dar	DUDOSO
7	@esperanza_py @Haddad_Fernando A llorar a la cocina	DUDOSO
8	@Tomehh_ andá a fregar los platos, la concha de tu madre e..e	DUDOSO
9	@ivanherreradgo Amaneciste Sensible mi Ivan? O Andas en tus Días? #SinOfender #EsPregunta	DUDOSO
10	@Moscardo_M Yo sin sentimientos? Mujer tenias que ser..	DUDOSO
11	@martuu_413 Aaa, pero andas zorra también	DUDOSO
12	@SennAgus Mujer tenías que ser, me vas a hacer pedirte por favor? Ahre	DUDOSO
13	@Kylomihijomyson Mándale a fregar si no lo quiere entender.	DUDOSO
14	@Pablosop Bien, se nota que aprendiste. Ahora, lo mismo con \Feminazi\""	DUDOSO

15	@perfilcom QUE ZORRA QUE SOS HDP	DUDOSO
16	No hay nada que me joda mas que las imbeciles que dicen:-Necesita pija-Mira como se vistió!-Viste la foto que subió? Se ve re *adjunte adjetivo denigrante*-Esa ropa es re trucha-Seguro le vino-Esta se chamuya a tal y tal, re puta/zorra es-Viste como bailaba? Tremenda trola	DUDOSO
17	@kathya_vzla Cállate zorra https://t.co/98YuDQt3TQ	DUDOSO
18	Cuidado guarra... Y la vez aquella que dijeron lo de que tenía que repetir no sé qué de la zorra que la huele el coño? Veeeeenga hasta luego #SomosLaAudiencia18N	DUDOSO

Descripción del corpus

Para la tarea de etiquetado se provee un corpus en español que ha sido recopilado durante las fechas 1/07/2018-31/12/2018. Para ello, se han seleccionado una serie de términos o expresiones que, según el contexto, pueden conllevar a comportamientos machistas.

Durante el periodo de creación del corpus, se han recogido tweets que contengan los siguientes términos: *feminazi, "loca del", "a la cocina", zorra, "como una niña", "las feministas", niñata, "como una mujer", "en tus días", "a fregar", mojigata, marimacho, nenaza, "para ser mujer", "odio a las mujeres", lagartona, "A las mujeres hay que", "las mujeres no deberían", "las mujeres de hoy en día", "mujer al volante", "mujer tenías que ser", "mucho feminismo pero", "pareces una puta", "para ser chica"*.

En total, se provee un corpus compuesto por 3600 tweets, 150 tweets para cada uno de los términos listados anteriormente.

Compartir el corpus etiquetado

Para esta tarea, es necesario analizar cada uno de los 3600 tweets del corpus y seleccionar una de las 3 etiquetas propuestas. El fichero de datos proporcionado tiene formato .csv con la siguiente estructura de campos:

“status_id” **“categoria”** **“tweet”**

donde:

- **Status_id**: Identificador único del tweet
- **tweet**: Representa el texto del tweet
- **categoría**: A completar por el anotador con una de las 3 etiquetas.

Para esta tarea, **se debe de completar la columna categoría con una de las 3 etiquetas: machista, no_machista y dudoso.**

Un ejemplo sería el siguiente:

```
1023558950510313473    dudoso     @hazteoir @PSOE Más vale que se marche a fregar !
```

Al término de la tarea, se debe de enviar un correo a frodrigue1395@alumno.uned.es con el fichero respetando el formato descrito.

Agreement

Cuando cada uno de los anotadores haya etiquetado **el 20% del corpus (720 tweets)** es necesario enviar la muestra etiquetada al correo frodrigue1395@alumno.uned.es .

En este punto, se estudiará el acuerdo entre los anotadores para evaluar si se ha comprendido adecuadamente la tarea.

Contacto

Para resolver cualquier duda o problema, se puede enviar un correo a frodrigue1395@alumno.uned.es .

Todo list