Una aproximación basada en aprendizaje automático para diversos problemas de procesamiento de lenguaje natural en redes sociales

Maite Giménez Fayos

Trabajo Final del Máster Universitario en
Inteligencia Artificial Reconocimiento de Formas e Imagen Digital.

Tutores: Lluís F. Hurtado y Ferran Pla

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Universidad Politècnica de València
València
Abril 2016

Tutores: Lluís F. Hurtado y Ferran Pla Maite Giménez Fayos

Una aproximación basada en aprendizaje automático para diversos problemas de procesamiento de lenguaje natural en redes sociales

Resumen

Este trabajo se centra en la resolución de distintas tareas propias del procesamiento automático del lenguaje natural, para lo cual se empleó una aproximación basada en algoritmos de aprendizaje automático.

Las tareas consideradas fueron: la detección del idioma, el análisis de sentimientos y la creación de perfiles de usuario. Se trata de tareas propuestas en competiciones internacionales y que han dado lugar a diversas publicaciones.

Todas estas tareas se plantearon utilizando datos extraídos de redes sociales, en particular textos de Twitter. En general, los textos que pueden encontrarse en estos medios poseen una serie de características (textos cortos y agramaticales) que plantean nuevos retos para el procesamiento del lenguaje natural.

En cada caso, se estudia el estado del arte y se propone un modelo que se ajuste a los requisitos de la tarea. Para ello, se emplean los recursos y los algoritmos de aprendizaje automático supervisado más adecuados. Finalmente, se ha analizado los resultados y se plantean futuras modificaciones que mejoren el comportamiento de los sistemas planteados.

Advisors: Lluís F. Hurtado y Ferran Pla Maite Giménez Fayos

A machine learning approach for natural language processing tasks in social media

Abstract

This work is focused on solving several Natural Language Processing tasks, for which an approach based on machine learning algorithms was used.

The tasks addressed were: language identification, sentiment analysis and author profiling. These tasks were proposed by international competitions which have led to publish several papers.

A data set of social media texts were used in these tasks, mainly from Twitter. Overall, these texts present some characteristics (short and ungramatical texts) that are challenging for Natural Language Processing techniques.

For each task, the state-of-the-art is studied and a model to solve the task is proposed. In order to create a valid model, several resources and supervised machine learning techniques were used. Finally, the results obtained were analyzed and improvements to the model were proposed to enhance the behavior of the model.

Índice general

Índice de figuras

Índice de tablas

Acrónimos

AP Author Profiling. 62–65, 71, 82

BOW Bag of Words. 7

DAG Directed Acyclic Graph. 15

IR Information Retrieval. 7, 12

LID Language Identification. 22–24, 35
LSA Latent Semantic Analysis. 44

MLE Maximum-Likelihood Estimation. 11, 12
MSE Mean Square Error. viii, 20, 58, 59

NLP Natural Language Processing. 2, 12, 22, 38, 39, 62–65, 84

RMSE Root Mean Square Error. 68, 78, 80, 81

SA Sentiment Analysis. 38–41
SVM Support Vector Machine. 14, 32, 34, 40, 54, 56, 58, 79, 81
SVR Support Vector Regression. 56

1 Introducción

Antes de introducirnos en la descripción detallada del trabajo realizado como conclusión del "Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital", queremos detenernos en este capítulo para contextualizar el ámbito teórico en el que se enmarca y enfatizar el interés de la comunidad científica en el estudio de la lingüística computacional.

1.1. Descripción del problema, motivación y objetivos

Comenzaremos por definir el objetivo de la lingüística computacional siguiendo la aproximación que podemos encontrar en el libro de Manning and Schütze [42]. Dicha área se enfoca en ser capaz de explicar y caracterizar el lenguaje natural que empleamos los hablantes de una lengua en nuestra comunicación, bien sea oral o escrita. Se trata de un campo de estudio interdisciplinar en el que confluyen la Lingüística y la Inteligencia Artificial.

Las primeras aproximaciones a este estudio se centraron en intentar compilar un conjunto de reglas que describieran el lenguaje, con la esperanza que este conocimiento permitiera comprenderlo.

Paulatinamente se evolucionó hacia la construcción de gramáticas formales que faciliten de un modo riguroso el avance en el estudio de la lingüística computacional.

Sin embargo, una de las mayores dificultades del Procesamiento Automático del Lenguaje Natural (*Natural Language Processing (NLP)*), radica tanto en la complejidad como en la ambigüedad del mismo. A pesar de que el conjunto de símbolos que componen un lenguaje sea discreto, el estudio pormenorizado de todas las posibles combinaciones de símbolos y usos tiende a infinito. Esta versatilidad del lenguaje natural enriquece la comunicación pero es responsable también de la dificultad que entraña la definición de un conjunto de reglas finito que lo caracterice.

En este trabajo final de máster se explora una aproximación estadística del procesamiento automático del lenguaje. Esta aproximación intenta describir patrones o estructuras en el lenguaje. De este modo podemos aplicar técnicas bien conocidas del aprendizaje automático para intentar resolver problemas derivados del procesamiento automático del lenguaje natural.
El aprendizaje automático supervisado consiste en, dado un conjunto de datos de entrenamiento compuesto por datos de entrada $x \in \mathcal{X}$ y etiquetas de clase para cada dato de entrada $y \in \mathcal{Y}$, obtener un modelo $f : \mathcal{X} \to \mathcal{Y}$ capaz de predecir la clase ante un nuevo dato de entrada. Si $\mathcal{Y} = \{1, 2, \ldots, C\} \subset \mathbb{N}$ estamos ante un problema de clasificación, en cambio si el dominio de la salida del predictor no es finito lo trataremos como un problema de regresión.
Los modelos estadísticos con los que trataremos se caracterizan por una aparente sencillez, lo que facilitará la computación pero han demostrado un buen rendimiento en las distintas tareas en los que han sido aplicados.

En general, todos los modelos tienen en común el método de entrenamiento y que están compuestos por tres módulos que se adaptan a la tarea concreta. Todos los sistemas propuestos se entrenan siguiendo estas tres etapas:

- **Entrenamiento**: en esta primera etapa se entrena el modelo empleando un conjunto de datos manualmente o semi-manualmente etiquetado, denominado comúnmente *"gold standard"*. En esta fase se determinan los mejores parámetros del modelo para la tarea considerada, por ejemplo los pesos.
- **Validación**: en esta segunda etapa ajustaremos los hiper parámetros del modelo. Nos servirá para evitar problemas como por ejemplo el sobre entrenamiento, lo cual consiste en, que el modelo haya aprendido los datos de ejemplo pero no es capaz de generalizar y por lo tanto de predecir correctamente nuevas muestras.

- **Test**: finalmente, en esta etapa se valida nuestro modelo con un conjunto de datos que no haya sido visto previamente. Emplearemos esta etapa para obtener los datos de rendimiento de nuestro sistema como pueden ser la precisión o el valor-F (*F-score*).

En la mayor parte de las tareas los datos que se proporcionan están divididos para ajustarse a cada una de las etapas de entrenamiento de un clasificador. Pero en caso de que no existieran particiones del corpus se ha realizado una validación cruzada para garantizar la independencia de los resultados de test. Cuando en la tarea se haya empleado esta técnica se expondrán el número de particiones empleadas así como cualquier otra característica técnica relevante.

Otra de las características que comparten todos los sistemas desarrollados es la subdivisión de los mismos en los siguiente módulos:

- El **módulo de preprocesado**: este primer módulo es el encargado de recoger y preparar los datos de entrada para el sistema. Se aplican filtros que limpien los datos, lo cual facilitará la clasificación.
- El **módulo de extracción de características**: es el encargado de extraer un vector de características para cada dato de entrada. Es deseable que dos objetos similares se encuentren cercanos en el espacio de representación, así como que diferentes instancias de un mismo objeto tengan la misma representación.
- El **módulo de clasificación**: es el último módulo de los sistemas de aprendizaje que hemos desarrollado y en el se entrena un algoritmo de clasificación, o regresión, a partir del vector de características extraídas de los datos de entrada.

En la figura 1.1, se muestra gráficamente el esquema descrito con los módulos que tienen en común los clasificadores que se han desarrollado.

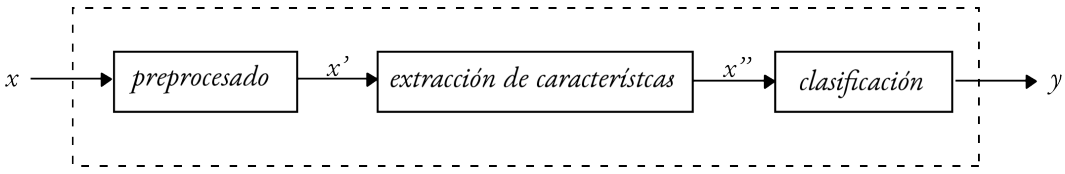


Figura 1.1: Esquema general de un clasificador.

Las aproximaciones estadísticas requieren de un gran volumen de ejemplos y/o contraejemplos para poder entrenar sistemas automáticos inteligentes. Afortunadamente para la lingüística computacional en las últimas décadas se ha realizado un gran esfuerzo por

recolectar textos y elaborar manualmente recursos léxicos que permitan explorar computacionalmente el estudio del lenguaje natural; esto es: diccionarios, corpora etiquetadas para distintas tareas, etc.

A esto se suma la revolución que han supuesto las redes sociales y con ellas la generación masiva de contenido digital. Lo que nos genera nuevos desafíos: ¿Cómo seremos capaces de gestionar la gran cantidad de volumen de información que se produce en tiempo real? ¿Cómo extraeremos información relevante de ese texto desestructurado? Debemos tener en cuenta que, particularmente en estos medios, las reglas de la lingüísticas se transgreden comúnmente, buscando usos creativos de la lengua, por contaminación de diversos idiomas, etc. De igual modo el lenguaje evoluciona adaptándose a los requisitos de las redes sociales.

En este trabajo se ha planteado abordar distintos problemas relacionados con el procesamiento del lenguaje natural sobre corpora extraídos de redes sociales. Se proponen distintos modelos basados en unidades lingüísticas y recursos manualmente elaborados para resolver cada una de las tareas planteadas: identificación del idioma, análisis de sentimientos y creación de perfiles de usuario. Para cada tarea se describirá en su capítulo correspondiente el corpus o corpora empleados, el modelo propuesto y los resultados obtenidos. El trabajo aquí recopilado ha sido presentado a distintas tareas internacionales y ha dado lugar a las publicaciones listadas en A.

1.2. Estructura del trabajo final de máster

El resto de la estructura de la tesis está organizada siguiendo el esquema que a continuación se detalla:

- **Capítulo 2**: Marco teórico.
  En este capítulo se definen los conceptos fundamentales que sustenta el trabajo que aquí se presenta. Se define el marco teórico para la representación del texto, los algoritmos de clasificación empleados así como las métricas que evalúan el comportamiento de los sistemas propuestos.
- **Capítulo 3**: Detección de idioma.
  El primer problema de procesamiento de lenguaje natural que exploraremos es la identificación del idioma dado un texto extraído de redes sociales, concretamente

de Twitter. Los idiomas contemplados en la tarea son los hablados en la península Ibérica: castellano, catalán, euskera, gallego, inglés y portugués.

- **Capítulo 4**: Análisis de sentimientos.
  Una tarea clave en el procesamiento de lenguaje natural es el análisis de sentimientos. El objetivo consiste en dada una afirmación realizada por un usuario en redes sociales determinar la polaridad de dicho mensaje. En este trabajo se presenta un sistema monolingüe entrenado con mensajes en inglés. En general se trata de identificar mensajes positivos, negativos o neutros. Aunque también presentamos un modelo capaz de determinar la polaridad con una mayor granularidad. Asimismo, hemos tratado el uso de lenguaje figurado en Twitter, en particular de ironía, y hemos analizado cómo estas características del mensaje afectan a los clasificadores presentados.
- **Capítulo 5**: Creación de perfiles de usuario.
  Finalmente se ha tratado la creación de perfiles de usuario empleando para ello textos publicados en redes sociales. El objetivo es descubrir características demográficas de los autores a partir de las características extraídas de los textos, como por ejemplo su sexo o su edad. Para esta tarea se ha desarrollado un sistema multilingüe capaz de trabajar con textos en castellano, inglés, italiano y holandés.

2
Marco teórico

A pesar de que para cada una de las tareas, que relataremos en los siguientes capítulos, se han desarrollado modelos distintos que se adapten a los requisitos particulares de las mismas, todos estos modelos comparten varias características. En primer lugar, la representación del texto se realiza basándose en *n-gramas* y los algoritmos de aprendizaje automático empleados se fundamentan en las Máquinas de Soporte Vectorial.
A continuación se definen las bases teóricas comunes para todos los modelos presentados en este trabajo.

2.1. Representanción del texto

Los datos de entrada con los que trabajamos en cada uno de los problemas consisten en textos extraídos de redes sociales, sin embargo los algoritmos de aprendizaje empleados únicamente son capaces de trabajar con valores de entrada numéricos.
Por lo tanto necesitamos un método que nos permita transformar cadenas de caracteres en datos de entrada válidos.

Aunque existen múltiples formas de proyectar los datos al espacio de representación esperado por los algoritmos de aprendizaje, en este trabajo nos centraremos en la representación de tipo bolsa de palabras empleando n-gramas.

En este apartado describiremos en profundidad estas técnicas de representación del texto fundamentales en los modelos que proponemos en este trabajo final de máster.

2.1.1. Bolsa de palabras

Una de las aproximaciones más populares para representar el texto sobretodo en tareas de clasificación es emplear el modelo de bolsa de palabras (*Bag of Words (BOW)*). Podemos encontrar este modelo de representación en distintos problemas como pueden ser: la recuperación de información *Information Retrieval (IR)*, visión por computador y por supuesto para el procesamiento del lenguaje natural.

El modelo bolsa de palabras representa la información teniendo únicamente en cuenta la aparición de los elementos de interés. En el caso del problema del procesamiento del lenguaje natural con el que estamos tratando, lo que haremos en este modelo será, en primer lugar, extraer el vocabulario de la tarea; después, para cada frase del corpus componemos un vector $v \in \mathbb{N}^{|V|}$ siendo $|V|$ la talla del vocabulario, y contamos el número de veces que aparece una palabra del vocabulario en la frase; de manera que tendremos para cada sentencia un vector indicando si aparece o no la unidad lingüística, la frecuencia de aparición, etc.

En el ejemplo 2.1.1 vemos como este modelo representaría un corpus formado por dos frases y un vocabulario de 11 palabras.

Ejemplo 2.1.1 Representación del texto usando una aproximación de bolsa de palabras.

Asumamos que nuestro corpus se compone de estas dos frases:

- w_1 = "El gato comerá pato dentro de un rato."
- w_2 = "El pato se esconde de un gato dentro de un zapato"

El vocabulario de este ejemplo está compuesto por las siguiente palabras:
{el, gato, comerá, pato, dentro, de, un, rato, se, esconde, zapato }.
Por lo tanto los vectores que representan la frecuencia de aparición de estas palabras

serán los siguientes:

- $v_1 = \{1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0\}$
- $v_2 = \{1, 1, 0, 0, 1, 2, 2, 0, 1, 1, 1\}$

Entrenaremos nuestro modelo con una matriz compuesta por esta representación de frases. Esta matriz tendrá la forma $V \in \mathbb{N}^{|V| x |M|}$ siendo M la talla del vocabulario.

Obviamente es fácil percatarse de los problemas que afloran al emplear esta aproximación: esta representación no tiene en cuenta el orden de las palabras en la frase, ni su función gramática, lo que puede conllevar, en el caso más extremo, que frases de sentidos totalmente contradictorios obtengan representaciones similares.

A pesar de esto, se trata de un modelo ampliamente empleado por su simplicidad y su eficiencia computacional. Y se ha empleado con éxito en todos los problemas de procesamiento del lenguaje natural que se han planteado en este trabajo.

2.1.2. N-Gramas

En la sección anterior hemos visto una representación del texto basada en bolsas de palabras. En este apartado se explica como esta misma aproximación puede ser explotada atomizando los elementos de interés. Para ello haremos uso de los n-gramas.

Un n-grama es una subsecuencia continua de n elementos de una secuencia que en nuestro caso será una secuencia de texto.
Los n-gramas permiten la representación vectorial de una secuencia de texto, lo que permite aplicar distintas técnicas matemáticas, como por ejemplo métodos estadísticos de análisis sobre esta representación.
Los n-gramas más comunes son los de talla 1 (*unigramas*), los de talla 2 (*bigramas*) y los de talla 3 (*trigramas*).
En la literatura podemos encontrar sistemas que entrenan con éxito modelos superiores a los trigramas. Por ejemplo Google dispone de un gran corpus en alemán, chino, español, francés, hebreo, inglés y ruso de hasta 5-gramas, creado a partir de 8,116,746 libros[38]. El Google n-grama corpus está disponible on-line [1] y ha sido empleado ampliamente en distintas investigaciones. [61, 19, 36]

[1] Puede acceder a este recurso en la https://books.google.com/ngrams

En el ejemplo 2.1.2 podemos ver un ejemplo con bigramas de palabras utilizando la aproximación basada en bolsa de palabras.

Ejemplo 2.1.2 Representación del texto bigramas de palabras.

Repetiremos el corpus visto ene el ejemplo 2.1.1, pero esta vez lo representaremos internamente haciendo uso de bigramas de palabras.
El vocabulario, en este caso, está compuesto por las siguiente palabras:
{el gato, gato comerá, comerá pato, pato dentro, dentro de, de un, un rato, rato <EOF>, el pato, pato se, se esconde, un gato, un zapato, zapato <EOF>}.
Y los vectores que representan este corpus son los siguientes:

- $v_1 = \{1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0\}$
- $v_2 = \{0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1\}$

Pero podemos ir un paso más allá y en lugar de utilizar palabras como elemento básico, podemos utilizar caracteres. Es una aproximación muy usual en la literatura para algunos de los problemas que estamos presentando en este trabajo. Si seleccionamos los caracteres dentro de los límites de las palabras estamos hablando de *n-gramas intrapalabras*. En cambio si empleamos una ventana deslizante obtendremos *n-gramas interpalabras*, los caracteres que forman los n-gramas pueden pertenecer a más de una palabra.

El ejemplo 2.1.3 muestra un ejemplo de esta aproximación.

Ejemplo 2.1.3 Representación del texto mediante n-gramas.

En este ejemplo vemos que unigramas, bigramas, trigramas y cuatrigramas podemos encontrar en las palabras “lenguaje natural”.

- Unigramas = { l, e, n, g, u, a, j, t, r}
- Bigramas = { le, en, ng, gu, ua, je, na, at, tu, ur, ra, al}
- Trigramas = { len, eng, ngu, gua, uaj, aje, nat, atu, tur, ura, ral}
- Cuatrigramas = { leng, engu, ngua, guaj, uaje, natu, atur, tura, ural}

En general, el número máximo de n-gramas que encontraremos está acotado superiormente por l^n siendo l el número de símbolos en el alfabeto y n el número de n-gramas que se tienen en cuenta. Esta cota superior suele ser mucho menor que el número de palabras de un vocabulario y para algunos problemas, como la identificación de idioma o la clasificación de texto[14], la frecuencia de aparición de los n-gramas es suficiente y simplifica

los cálculos.

Sin embargo, se puede argumentar que los n-gramas presentan varios problemas como, por ejemplo que no capturan de forma explícita información lingüística o la dificultad para lidiar con las palabras no vistas durante el entrenamiento, a las que denominamos palabras fuera de vocabulario, por lo que se hace uso de distintos métodos de suavizado.

La ley de Zipf

Es obvio predecir que en nuestro lenguaje existen palabras más frecuentemente empleadas que otras. Pues bien, la ley de Zipf [83] expresa esta idea del siguiente modo: la palabra iésima más frecuente en el lenguaje natural presenta una frecuencia inversamente proporcional al *ranking* r.

$$f_i \propto \frac{1}{r} \tag{2.1}$$

De modo que si la palabra más frecuente aparece f_1 veces, la segunda más frecuente aparecerá la mitad de veces, la tercera aparecerá un tercio,etc.
Se trata de una regla empírica, que nos dice que siempre existen unas palabras que dominan el problema. La ley de Zipf caracteriza un conjunto de datos en lenguaje natural y en este trabajo se utiliza para caracterizar la complejidad de los problemas para tomar decisiones sobre el tipo de modelo que se propone.
La ley de Mandelbrot es una generalización de la ley de Zipf, menos empleada en la literatura, a pesar de que, en general, se ajusta más a la distribución real de los datos. En el libro de Manning and Schütze [42], se puede encontrar una descripción detallada con ejemplos de ambas reglas.

2.1.3. Modelos de lenguaje estadísticos

No obstante, además de almacenar los n-gramas en un vector podemos, empleando la teoría de la probabilidad, crear un modelo de lenguaje estadístico.

Un modelo de lenguaje estadístico es un modelo matemático que permite calcular la probabilidad de una secuencia W [43]. Para dicho cálculo, se asume que la probabilidad de un n-grama w depende de cierta historia vista h.

La probabilidad de que W variables aleatorias tomen el valor de la secuencia[2] w_1^n, que puede descomponerse utilizando la regla de la cadena:

$$\begin{aligned} P(w_1^n) &= P(w_1)P(w_2|w_1)P(w_3|w_1^2)\dots P(w_n|w_1^{n-1}) \\ &= \prod_{i=1}^{n} P(w_n|w_1^{n-1}) \end{aligned} \tag{2.2}$$

Sin embargo, para estimar $P(w_n|w_1^{n-1})$ necesitaríamos encontrar todo el contexto w_1^{n-1} en el conjunto de datos de entrenamiento, pero el lenguaje es creativo y cualquier posible contexto puede no aparecer[43]. Por lo tanto, se aplica la asunción de **Markov**, que simplifica el modelo suponiendo que la probabilidad de aparición de un n-grama no depende de toda la historia previa sino de los últimos n-gramas vistos, por ejemplo en el caso de los modelos de lenguajes basados en bigramas la probabilidad del n-grama w_n dependerá únicamente de la probabilidad condicional del n-grama previo w_{n-1}, en el caso de los modelos basados en trigramas la probabilidad del n-grama w_n dependerá de los n-gramas w_{n-2}^{n-1}, etc.
En general, aproximaremos un modelo N-gramas, siendo N el número máximo de n-gramas que tenemos en cuenta, como:

$$P(w_1^n) \approx \prod_{k=1}^{n} P(w_k|w_{k-N+1}^{k-1}) \tag{2.3}$$

Estas probabilidades pueden estimarse mediante el método de máxima verosimilitud *Maximum-Likelihood Estimation (MLE)*, normalizando la frecuencia de aparición de los n-gramas en el corpus de entrenamiento. Por consiguiente estimaremos un modelo de N-gramas de la siguiente forma:

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})} \tag{2.4}$$

Los modelos de lenguaje estadístico se comportan razonablemente bien y el entrenamiento de los mismos es sencillo. En este apartado hemos presentado la aproximación

[2] Por simplicidad utilizaremos la notación propuesta por Martin and Jurafsky [43] de modo que reescribiremos la probabilidad de que la variable aleatoria X_i tome el valor *wi* siguiendo el criterio $P(X_i = w_i) \rightarrow P(w_i)$

más simple. Con lo visto hasta aquí para aquellos n-gramas que no hayamos visto durante el entrenamiento, MLE no le asignará ninguna probabilidad de aparición (la probabilidad será cero). Por esto, se aplican métodos de suavizado, que permiten distribuir cierta probabilidad de los eventos vistos en el conjunto de entrenamiento entre los eventos no vistos.Entre los métodos de suavizado más comunes podemos encontrar *backoff* que combinan n-gramas de distintas tallas, como puede ser un modelo entrenado con trigramas y bigramas, interpolación lineal, etc. Sin embargo, explicar estas técnicas queda fuera del ámbito de este trabajo.

2.1.4. TF-IDF

Una técnica muy común empleada en *IR* y que se aplica comúnmente a muchas tareas de *NLP* consiste en ponderar la frecuencia de aparición de un término t en función de su relevancia en el documento c, como podemos ver a continuación:

$$tf(t,c) = \frac{f(t,c)}{max\{f(w,c) \forall w \in c\}} \tag{2.5}$$

Para evitar que términos muy comunes pero con poca relevancia se ponderen muy favorablemente se incorpora un factor de frecuencia inversa de documento que atenúa el peso de estos términos.

$$idf(t,C) = log\frac{|C|}{|c \in C : t \in c|} \tag{2.6}$$

Siendo C, la cardinalidad del corpus o el número de documentos en el corpus.
Finalmente calcularemos la importancia de cada término siguiendo el coeficiente tf-idf mediante la ecuación 2.7.

$$tf - idf(t,c,C) = tf(t,c) \times idf(t,C) \tag{2.7}$$

La técnica presentada en este apartado mejora la estimación de los modelos basados en conteo o frecuencia de aparición de términos.

2.2. Algoritmos de aprendizaje automático

Comenzaremos definiendo el aprendizaje automático con una cita clásica de Mitchell[45]: " Se dice que un programa es capaz de aprender de una experiencia E a realizar una tarea T con un rendimiento P, si se observa una mejora en P al realizar la tarea T con respecto a la experiencia E.".

Cada una de las tareas T se describirán en detalle en su capítulo correspondiente, pero como ya hemos enunciado previamente consistirán en problemas no resueltos del procesamiento del lenguaje natural: identificación del idioma, análisis de sentimientos y creación de perfiles de usuario.
Estas tareas se pueden abordar utilizando básicamente dos técnicas:

- *Clasificación*: un clasificador debe decidir, dada una entrada, a qué clase pertenece. El número de posibles clases es finito y conocido de antemano. Por lo tanto, el clasificador deberá aprender una función $f(x)$, $f : \mathbb{R}^n \to \{1 \dots c\}$, que ante una entrada x determine su clase c.
- *Regresión*: se trata de tareas que deben predecir valores numéricos dada una entrada. De modo que necesitamos entrenar una función $f : \mathbb{R}^n \to \mathbb{R}$.

Análogamente, describiremos la experiencia E en cada capítulo en profundidad, en todos los casos se trata de textos escritos en lenguaje natural y extraídos de redes sociales.
El conjunto de textos de ejemplo que se proporciona a un algoritmo de aprendizaje automático como experiencia se denomina *corpus*.
Los algoritmos de aprendizaje automático puede dividirse a grandes rasgos en algoritmos supervisados y algoritmos no supervisados.

- Los *algoritmos supervisados* son aquellos capaces de aprender a partir de datos etiquetados, es decir, para cada muestra de entrenamiento sabemos a priori su clase.
- En el caso de los *algoritmos no supervisados* únicamente tenemos las características de las muestras de entrenamiento y el objetivo de estos algoritmos es descubrir algún tipo de estructura (*cluster*) en los datos que permita realizar generalizaciones.

En este trabajo se ha contado con datos etiquetados y se ha planteado el uso de algoritmos supervisados. Aunque por la naturaleza de los datos, es sencillo recopilar gran volumen de información no etiquetada; el etiquetado es manual y costoso, por lo que

sería interesante ampliar el trabajo aquí presentado utilizando métodos no supervisados. Esto se analizará en el capítulo 6.

Definiremos las métricas P con las que evaluar el rendimiento del sistema en el apartado 2.3.

Finalmente, emplearemos este apartado para definir las bases del algoritmo capaz de aprender a partir de datos.

2.2.1. Máquinas de soporte vectorial

Las máquinas de soporte vectorial (*Support Vector Machine (SVM)*)[16] forman parte de la familia de clasificadores que se denominan de margen máximo. Estos clasificadores buscan un hiperplano que separe de forma óptima las muestras de dos clases.
Entre todos los posibles hiperplanos separadores buscamos aquel que cumpliendo una serie de restricciones sea el de margen máximo. Intuitivamente, se trata del hiperplano que pase más lejos de las muestras de entrenamiento.

Este problema se puede formalizar como el problema de encontrar una función $f(x_1, \ldots, x_n)$ discriminante óptima sujeta a una serie de restricciones $g(x_1, \ldots, x_n) = 0$. Podemos pensar en g como el margen del hiperplano separador que además debe cumplir un conjunto de condiciones. La forma canonica del hiperplano separador de margen máximo se obtiene minimizando la función discriminante y puede resolverse empleando la técnica de los multiplicadores de Lagrange, que se basa en el teorema de Karush-Kuhn-Tucker.

El conjunto de muestras de entrenamiento más cercano al hiperplano separador óptimo que es el que pasa más lejos del resto de muestras de entrenamiento tal que los multiplicadores de Lagrange óptimos son distintos de cero y por lo tanto, la función discriminante vale ± 1 dependiendo de a cual de las dos clases pertenezcan. Por lo tanto, el vector solución es una combinación linear de las muestras de entrenamiento, cuyos coeficientes son los multiplicadores de Lagrange y las etiquetas de clase.

Si disponemos de pocas muestras de aprendizaje podremos obtener la función discriminante empleando métodos analíticos. Sin embargo, si tenemos un gran número de muestras de entrenamiento es necesario recurrir a métodos iterativos que superen los pro-

blemas relacionados con el coste computacional, en este caso se trata de una implementación compleja que añade cierta heurística y por lo tanto la solución que encontremos será sub-óptima.

En la práctica, es muy poco probable encontrar con conjuntos de muestras linealmente separables por lo que se incluye cierta tolerancia. Por lo tanto, se relajan los requisitos incorporando los *márgenes blandos* a las máquinas de soporte vectorial, que permiten que algunas muestras no se clasifiquen correctamente.

Las máquinas de soporte vectorial son clasificadores binarios. En el caso del problema de clasificar múltiples clases, como son los problemas aquí presentados, existen distintas alternativas para conseguir clasificadores multiclase. Las dos aproximaciones más comúnmente empleadas son: en primer lugar la aproximación (*one-vs-all*), en la que entrenaremos C clasificadores, siendo C el número de clases a clasificar y cada clasificador compara una clase contra el resto, la segunda aproximación se denomina (*one-vs-one*) y consiste en entrenar tantos clasificadores como pares de clases se pueden formar, que serán $\frac{C(C-1)}{2}$ clasificadores binarios.

Ante una nueva muestra desconocida debemos asignarle una clase con los clasificadores SVM que hemos entrenado. Podemos emplear alguno de los siguientes criterios para llevar a cabo esta decisión:

- Votación: Cada uno de los clasificadores "vota" cual de sus dos clases consideran que son más probables. Para cada clase predecimos la etiqueta de clase que más haya sido votada. En caso de empate deberemos tomar una decisión, esto implica añadir una heurística.
- Eliminación en un grafo acíclico no dirigido (*Directed Acyclic Graph (DAG)*): Construimos un grafo acíclico dirigido y en cada nodo tendremos los clasificadores *one-vs-one* que tenemos inicializados y ajustados. En un nodo que compara la clase I, J, si este clasificador predice que la muestra no es de la clase I seguiremos explorando el grafo por la rama J. Lo que buscamos por lo tanto es un camino en un DAG, que se puede implementar con un coste lineal con la talla del grafo. Este método no permite empates. Construiríamos este grafo ordenando los clasificadores en función de la entropía de las clases que clasifiquen.

Como hemos comentado previamente, la mayor parte de las muestras procedentes de problemas reales no pertenecen a clases linealmente separables. Pero sabemos que un con-

junto de N muestras será linealmente separable en el peor de los casos en un espacio de N-1 dimensiones. Sin embargo, como se verá en el apartado 2.2.2 aparece el problema de la dispersión de los datos.

Por lo tanto, emplearemos funciones que nos permitan proyectar los datos de un espacio de representación a otro. Estas funciones se denominan *kernels*.
Los *kernels* realizan una transformación de espacio de forma eficiente,(*kernel trick*), que nos permiten trabajar de forma implicita en un espacio de dimensionalidad mayor, pero los cálculos se realizan en el espacio sin transformar.
El problema ahora consiste en encontrar el vector de pesos y un umbral en el espacio transformado de forma que las muestras seleccionadas como vectores soporte que esten bien clasificadas en el espacio transformado, sean canónicas y se encuentren lo más lejanas posible al hiperplano separador. También podemos incluir muestras, que sin estar bien clasificadas, formen parte de los vectores soporte porque incluimos márgenes blandos.
En la literatura normalmente los *kernels* y máquinas de soporte vectorial aparecen siempre unidos, y aunque esto no tendría porque ser siempre así, en estos trabajos siempre emplearemos un *kernel* que realice el cambio de espacio y entrenaremos después una máquina de soporte vectorial.

Los *kernels* que más emplearemos en estos trabajos son: el *kernel* lineal y el *kernel* polinómico.
Los *kernels* lineales son los más simples, realizan una transformación de tipo $\langle x, x' \rangle$.
Los parámetros relevantes a evaluar para las máquinas de soporte vectorial y los *kernels* lineales son:

- C: este parámetro nos indica la tolerancia del sistema que estamos entrenando. Si el valor de C es muy grande muchas muestras de entrenamiento pueden estar mal clasificadas, mientras que un C pequeño incluye muestras bien clasificadas y por lo tanto seleccionará menos vectores soporte.
- ε: este parámetro determina el criterio de parada de nuestro algoritmo.

Por otro lado, los *kernels* polinómicos realizan un cambio de espacio de tipo $(\gamma\langle x, x' \rangle + r)^d$. Son *kernels* algo más complejos que los lineares por lo que pueden adaptarse mejor a las muestras de entrenamiento.
Los parámetros a ajustar en este tipo de *kernels* serán:

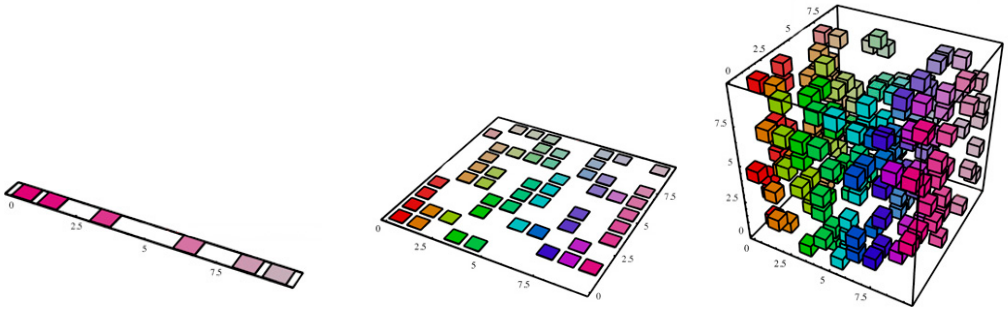
- C y ε igual que en el caso del kernel lineal.

- Grado del polinomio(d): el grado máximo del polinomio. Para este experimento probaremos kernels polinómicos de grado 2, 3, 4 y 5.
- γ: El valor de γ determina la pendiente de la función polinómica.
- r: coeficiente independiente.

Los algoritmos de aprendizaje descritos en este apartado son los que hemos empleado fundamentalmente para resolver los problemas, empleando la extensión de las máquinas de soporte vectorial para trabajar también con los problemas de regresión.
Si en algún problema en particular se empleara otra estrategia se describirá en el propio apartado.

2.2.2. El problema de la dimensionalidad

A priori, podríamos suponer que con cada incremento del espacio de representación estaríamos mejorando el comportamiento del algoritmo de aprendizaje automático, puesto que aumentamos la dispersión de los datos y por lo tanto debería ser más simple encontrar un hiperplano separador. Sin embargo, lo que ocurre en realidad es que si aumentamos el espacio de representación de los datos sin incluir más muestras de entrenamiento nos encontraremos ante el problema del sobreentrenamiento (*overfitting*), es decir, nuestro sistema no es capaz de generalizar y en consecuencia, no será capaz de predecir correctamente una nueva muestra no vista.



(a) Espacio 1-dimensional con 10 posiciones

(b) Espacio 2-dimensional con 10^2 posiciones

(c) Espacio 3-dimensional con 10^3 posiciones

Figura 2.1: Ejemplo del problema de la dimensionalidad. En el caso de una dimensión únicamente necesitamos diferenciar 10 áreas de interés. Con dos dimensiones, el algoritmo deberá ser capaz de diferenciar entre 100 áreas distintas y por lo tanto necesitaremos ver al menos 100 muestras de aprendizaje. Por último en el caso de 3 dimensiones necesitaremos distinguir entre 10^3 regiones del espacio lo cual complica todavía más el problema. En general, en un problema con d dimensiones y v valores a distinguir en cada eje, necesitaremos ver $O(v^d)$.

Se trata de un problema estadístico que surge porque el número de configuraciones posibles que resuelven el problema es mucho mayor que el número de muestras de entrenamiento. La figura 2.1 ilustra este problema. [3]

Además, al incrementar el espacio de representación necesitaremos ajustar más parámetros en nuestro algoritmo de aprendizaje, lo que puede hacer que nuestro problema no sea factible computacionalmente.

2.3. Métricas empleadas para evaluar el rendimiento de los sistemas

Un punto fundamental para determinar objetivamente el comportamiento de los modelos que proponemos en este trabajo, es definir una serie de métricas de evaluación. Las métricas descritas en esta sección son bien conocidas y permitirá comparar los resultados aquí presentados con los reportados en la literatura.

2.3.1. Exactitud, precisión, exhaustividad y valor-F

Comenzaremos describiendo las métricas para los clasificadores binarios. Sin embargo, antes de revisar estas métricas definiremos los siguientes conceptos:

- Verdaderos Positivos (*true positives (tp)*): el clasificador ha predicho que una instancia pertenece a la clase y es correcto.
- Falsos Positivos (*false positives (fp)*): el clasificador ha predicho que una instancia pertenece a la clase y no es correcto.
- Verdaderos Negativos (*true negatives (tn)*): el clasificador ha predicho una instanciaque no pertenece a la clase y es correcto.
- Falsos Negativos (*false negatives (fn)*): el clasificador ha predicho que una instancia no pertenece a la clase y no es correcto.

Para tareas de clasificación, una de las métricas más simples y ampliamente empleadas es la exactitud (*accuracy*). La cual nos indica el porcentaje de muestras correctamente clasificadas por el sistema. Su fórmula es la expresada en la ecuación 2.8.

[3] Se trata de un ejemplo extraído del libro de Goodfellow et al. [27]

$$Accuracy = \frac{tp + tn}{tp + fp + tp + fn} \tag{2.8}$$

Otra medida muy común nos indica la precisión del sistema definida en la ecuación 2.9.
En este caso, nos indica el porcentaje correctamente clasificado sobre el total de datos a los que se ha predicho que pertenecen a la clase.

$$Precision = \frac{tp}{tp + fp} \tag{2.9}$$

La precisión indica como de repetible y robusto es un sistema mientras que la exactitud indica cuan acertadas son las predicciones realizadas por este sistema. Un buen sistema debe ser preciso y exacto.

Análogamente podemos emplear la exhaustividad (*recall*) para evaluar los modelos propuestos. Esta métrica se dercribe en la fórmula 2.10. En este caso, nos indica el porcentaje correctamente clasificado sobre el total de datos que pertenecen a la clase.

$$Recall = \frac{tp}{tp + fn} \tag{2.10}$$

Estas métricas se pueden generalizar para problemas multiclase. La exactitud se computará como el $\frac{\text{num. aciertos}}{\text{total de muestras}}$. Para calcular la precisión y la exhaustividad recorreremos la matriz de confusión (M) y obtendremos que la precisión para clase i-ésima es $Precision_i = \frac{M_{ii}}{\sum_j M_{ji}}$, mientras que la exhaustividad de la clase i-ésima es $Recall_i = \frac{M_{ii}}{\sum_j M_{ij}}$.

Por último, definiremos el valor-F (*f-measure*). Se trata de una métrica que resume tanto la precisión como la exhaustividad, como puede comprobarse en las siguientes fórmulas: 2.11 y 2.12; es una media armónica de estas dos métricas. Es muy común fijar el valor de $\beta == 1$.

$$F \quad = (1 + \beta^2) \frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{2.11}$$

$$F - 1 = 2 \frac{precision \cdot recall}{precision + recall} \tag{2.12}$$

Cuando necesitamos calcular estas métricas para problemas de más de dos clases necesitamos un método para promediar la aportación de cada clase al total. Existen dos aproximaciones: *micro* y *macro*. En el caso de la aproximación *micro* se calculan las métricas globalmente sumando el número de verdaderos/falsos positivos/negativos. Mientras que en la aproximación *macro* simplemente calcularemos la media de la precisión o la media de la exactitud, en este caso, la métrica no se ve afectada porque las clases no estén balanceadas.

2.3.2. Error cuadrático medio

Cuando estamos ante problemas de regresión, las métricas descritas en el apartado anterior no son adecuadas.
Imaginemos el caso de que dada una muestra de entrada su etiqueta real sea por ejemplo 2.3, y un algoritmo de aprendizaje predice un valor de 3 mientras que otro predice un valor de 9, en ambos casos estaríamos ante una muestra icorrectamente clasificada y obtendríamos el mismo resultado si empleamos las métricas del apartado 2.3.1, pero el primer algoritmo está más próximo al valor real que el segundo. Por lo tanto necesitaremos estimadores o métricas más apropiadas para estas tareas.
Tanto el error cuadrático medio (MSE) como la distancia coseno, que se describe en el apartado 2.3.3 son métricas empleadas para evaluar el comportamiento de los sistemas entrenados en problemas de regresión.

Sea $\hat{\mathcal{Y}}$ el vector con n predicciones y $\mathcal{Y}$ el vector con los valores correctos para esas muestras.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{\mathcal{Y}}_i - \mathcal{Y}_i)^2 \tag{2.13}$$

Nuestro objetivo será minimizar el error medio cuadrático de las predicciones realizadas por el algoritmo de aprendizaje entrenado.

2.3.3. Distancia coseno

Si creamos un vector con las predicciones $\hat{\mathcal{Y}}$ y un vector $\mathcal{Y}$ con los valores reales, podemos evaluar el algoritmo de aprendizaje empleando la distancia coseno entre estos dos vectores en un mismo espacio vectorial $\mathcal{D}$. Este espacio vectorial será n-dimensional, sien-

do n el número de muestras que hemos clasificado.

La distancia coseno mide el coseno del ángulo que forman estos dos vectores en el espacio vectorial. Esta métrica se deduce a partir del productor escalar de dos vectores (Ecuación 2.14)

$$\mathcal{Y} \cdot \hat{\mathcal{Y}} = \|\mathcal{Y}\|\|\hat{\mathcal{Y}}\| \cos \vartheta \tag{2.14}$$

Si desarrollamos esta ecuación obtendremos la fórmula de la distancia coseno. (Equación 2.15)

$$\begin{aligned} similarity &= \cos \vartheta \\ &= \frac{\mathcal{Y} \cdot \hat{\mathcal{Y}}}{\|\mathcal{Y}\|\|\hat{\mathcal{Y}}\|} \\ &= \frac{\sum_{i=1}^{n} \mathcal{Y}_i \hat{\mathcal{Y}}_i}{\sqrt{\sum_{i=1}^{n} \mathcal{Y}_i^2} \sqrt{\sum_{i=1}^{n} \hat{\mathcal{Y}}_i^2}} \end{aligned} \tag{2.15}$$

Siendo $\hat{\mathcal{Y}}_i$ y $\mathcal{Y}_i$ las componentes i-ésimas de los vectores $\hat{\mathcal{Y}}$ y $\mathcal{Y}$ respectivamente.

Si el vector con las predicciones y el vector con las etiquetas presentan la misma dirección y sentido formarán un ángulo de 0º, por lo tanto como el $\cos 0 = 1$ estos dos vectores presentarán una similitud máxima de 1. Si por el contrario, el algoritmo no ha predicho ninguna etiqueta correctamente los dos vectores serán perpendiculares y su distancia coseno será 0 ya que $\cos 90 = 0$.

Con la distancia coseno tenemos una métrica normalizada entre [0-1] que nos permite comparar las predicciones de dos algoritmos de aprendizaje distintos.

3
Detección de idioma

En este capítulo introduciremos el primer problema de NLP que se ha abordado: la detección automática del idioma.

Este capítulo se organiza mediante el siguiente esquema: comenzaremos introduciendo la tarea y los retos que esta presenta; a continuación, en la sección 3.2, repasaremos brevemente las aproximaciones que podemos encontrar en la literatura; en la sección 3.3, describiremos los detalles del "Taller sobre Identificación del Idioma", concurso propuesto en el marco de la XXX edición del Congreso Anual de la Sociedad Española para el Procesamiento del Lenguaje Natural y donde se ha presentado este trabajo; presentaremos los dos modelos propuestos en dicho concurso en la sección 3.4 y la evaluación de los mismos en la secciones 3.5 y 3.6; concluiremos este capítulo analizando los resultados obtenidos y reflexionaremos acerca de como mejorar el trabajo realizado.

3.1. Introducción al problema

La identificación de idioma, *Language Identification (LID)*, se ha tratado tradicionalmente como un problema de clasificación de texto. Una definición formal del problema sería: Dado un texto, de una longitud variable, la tarea consiste en decidir el idioma o idiomas en el que está escrito de entre un conjunto de idiomas posibles.
Es innegable que en Internet el idioma predominantes es el inglés. Sin embargo, en los úl-

timos años y en especial con la popularización de las redes sociales, esta predominancia se ha mitigado, permitiéndonos encontrar una amplia representación de distintas lenguas. El elevado número de idiomas, algunos de ellos muy similares entre si, supone un reto que ha reavivado el interés por resolver el problema de LID en redes sociales.

Twitter proporciona una etiqueta de idioma en los meta-datos de cada tweet. La precisión del clasificador interno de Twitter no es óptima[1]. Sin embargo, como predomina el uso del inglés en la distribución de idiomas empleada por los usuarios de la plataforma, la precisión global del sistema se mantiene en un admirable 99 %. [2].
Sin embargo, si la tendencia se mantiene, cada vez podremos encontrar más hablantes de lenguas distintas al inglés, lo cual complica la tarea de la identificación del idioma. En los resultados aquí presentados, en el apartado 3.6, podemos ver como el comportamiento del identificador automático de Twitter cae bruscamente ante circunstancias más desafiantes.
Esta tarea es capital en el procesamiento automático del lenguaje. Muchos sistemas hacen uso de recursos especialmente diseñados para un idioma concreto. Por lo tanto los errores que se cometan en la fase inicial de la identificación del idioma se propagarán por el sistema, penalizando el rendimiento global.
Debemos ser capaces de desarrollar sistemas de identificar las múltiples lenguas empleadas por los usuarios, o en caso contrario estaremos obviando la información que los usuarios no angloparlantes pueden ofrecernos.

En este contexto, se propuso el taller tweetLID 2014 en el que los participantes debían identificar el idioma de tweets escritos en la península Ibérica. El resto de los detalles de la tarea se describirán en la sección 3.3.

3.2. Estado del arte

El trabajo de Gold[25] en la década de 1960 supuso el punto de partida para la tarea del LID. Pero no fue hasta la década de 1990 cuando se popularizó como una tarea de clasifi-

[1]Twitter ha desarrollado su propio sistemas de aprendizaje automático para identificar el idioma de un tweet e incorpora esta información como meta-dato dentro de un tweet. https://blog.twitter.com/2013/introducing-new-metadata-for-tweets

[2]El equipo de desarrollo ha realizado un estudio del impacto de los idiomas menos representados en los resultados de su clasificador que pueden consultarse en: https://blog.twitter.com/2015/evaluating-language-identification-performance

cación supervisada dentro del área de estudio del Procesamiento del Lenguaje Natural. El interés por esta tarea se ha reavivado en los últimos años con el surgimiento y popularización de las redes sociales.

En las primeras aproximaciones [9] se abordó el problema construyendo un conjunto de reglas que modelaba el conocimiento experto sobre la identificación de idioma. Esta aproximación se vio fuertemente influenciada por el juego de caracteres utilizado en un idioma, y aunque era capaz de identificar un conjunto de idiomas, que no comparten características entre si, su comportamiento caía bruscamente cuando tratamos con idiomas similares o textos cortos.

Cuando tratamos con textos largos y estructurados, las aproximaciones basadas en n-gramas de caracteres, que ya hemos visto en la sección 2.1.2, han demostrado tener éxito resolviendo la tarea con unas tasas de error muy reducidas. Hasta el punto de que en la literatura, podemos encontrar este problema resuelto en ciertas condiciones [14], con unas tasas de precisión del 99.8 %.
Además de las aproximaciones basadas en n-gramas, también podemos encontrar sistemas que explotan la presencia o frecuencia de caracteres característicos de ciertas lenguas, la presencia de determinadas palabras, bases de conocimiento, etc[72].

En el trabajo de Baldwin and Lui [6] se incide en que la tarea de LID no se encuentra resuelta, especialmente cuando no tratamos con texto normativo. En este estudio, los autores analizan la importancia de la longitud del texto para la identificación del idioma en documentos web, obteniendo tasas de acierto que oscilan entorno al 70 % para documentos cortos y alrededor del 90 % para documentos largos. Por lo tanto, en circunstancias reales, todavía es posible mejorar notablemente el rendimiento de los sistemas dedicados al LID sobretodo para textos cortos y agramaticales.

Además, las redes sociales planean nuevos retos en el LID. En las redes sociales no tenemos que lidiar únicamente con textos extremadamente sintéticos, sino con múltiples dificultades asociadas a los nuevos usos de la lengua en estos medios, como por ejemplo: frases agramaticales, lenguaje específico, el uso de emoticones, de abreviaturas, etc. En esta línea, encontramos distintos trabajos[41, 12] que emplean técnicas que se intentan adaptar a este dominio y demuestran lo relevante de crear sistemas capaces de identificar el idioma adaptándose a las limitaciones que presenta el lenguaje en las redes sociales. Por

ejemplo, el trabajo de Goldszmidt et al. [26] discrimina entre idiomas empleando modelos de lenguaje estadísticos entrenados a partir de las frecuencias de caracteres aprendidos de la Wikipedia. Asimismo, el trabajo de Carter et al. [12] introduce información de Twitter, como puede ser: el identificador de idioma del perfil del usuario, el contenido de los enlaces que puedan aparecer en un tweet, el idioma en el que identifica Twitter la conversación a la que pertenece el tweet, etc.

Las plataformas de microblogging como Twitter han despertado el interés de la comunidad científica porque se trata de una excelente plataforma para analizar el comportamiento y los contenidos que los usuarios generan.
Se han celebrado competiciones internacionales para evaluar los sistemas de identificación de idioma desarrollados por distintos grupos de investigación. Como por ejemplo: *"Innovative Use of NLP for Building Educationsl Applications"* (BEA)[3] en el marco del NAACL (*North American Chapter of the Association for Computational Linguistics – Human Language Technologies*) en 2013, la tarea sobre discriminación de idiomas similares (DSL-2014)[4] desarrollada para el COLING (*International Conference on Computational Linguistics*) de 2014 o el TweetLID que presentamos en este trabajo.

3.3. Descripción de la tarea

Con el fin de afrontar los retos en el reconocimiento automático del idioma descritos en el apartado anterior, se desarrolló una tarea en el marco de la conferencia de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN) en 2014, denominada TweetLID[5], donde se animaba a los distintos grupos de investigación a presentar sus sistemas, estableciendo un marco comparativo común que se recoge en el artículo Zubiaga et al. [84], que será referente para los trabajos futuros en este área de investigación. La tarea propuesta consistió en identificar el idioma en el que estaban escritos un conjunto de tweets. Los idiomas que se tuvieron en cuenta en esta tarea fueron los seis idiomas más hablados en la Península Ibérica: castellano, portugués, catalán, gallego -todas estas lenguas romances-, euskera e inglés.
El hecho de que cuatro lenguas pertenezcan a una misma familia y al bilingüismo o trin-

[3] http://www.cs.rochester.edu/~tetreaul/naacl-bea8.html
[4] http://corporavm.uni-koeln.de/vardial/
[5] http://komunitatea.elhuyar.eus/tweetlid/?lang=es_es

lingüismo de los hablantes, que provoca errores gramaticales y que aparezcan palabras en distintos idiomas en un mismo tweet, dificulta la tarea.
Concretando, los tres problemas fundamentales de la identificación de idioma en los que se centra la tarea TweetLID son:

1. Detección de idiomas similares. Aquellos idiomas que derivan de una misma familia de lenguas comparten diversas características que dificultan la identificación de los mismos empleando algoritmos de aprendizaje automático.
2. Detección de múltiples idiomas en un mismo texto. Lo cual es especialmente crítico en regiones donde las personas son bilingües o trilingües, como es el caso de distintas regiones de la península Ibérica, en la que los hablantes pueden mezclar aleatoriamente en una misma sentencia los idiomas hablados en esta región así como el inglés.
3. Identificación de idioma en textos cortos procedentes de redes sociales. En este tipo de textos no disponemos de información de contexto suficiente, se cometen errores sintácticos y los hablantes suelen emplear argot y abreviaturas para expresarse lo cual dificulta la tarea.

En la tarea podemos encontrar tweets multilingües, es decir tweets que contienen palabras en varios idiomas, y en estos se deben identificar los distintos idiomas -dos o incluso tres- presentes, pero no es necesario segmentar la identificación para determinar que parte del tweet está escrito en uno u otro idioma.
Cuando no es posible determinar el idioma en el que está escrito un tweet por la similaridad de las lenguas estamos ante tweets ambiguos. Por ejemplo, el tweet "Acabo de publicar una foto", podría estar escrito en castellano o en catalán.
Además, debido al uso que los usuarios de Twitter hacen del lenguaje existe la posibilidad de no ser capaz de distinguir en que idioma está escrito entre los distintos idiomas que se consideraron en la tarea. Por ejemplo, tweets que únicamente contienen onomatopeyas, en cuyo caso se deben identificar como indeterminados (*und*).
Finalmente si aparecen tweets en otros idiomas que queden fuera del ámbito de estudio de esta tarea, como tweets escritos en francés o alemán, deberán etiquetarse como *other*.

Para la elaboración del corpus de la tarea, se recopilaron 35000 tweets, los cuales fueron escritos entre el 1 y el 31 de Marzo de 2014. Para asegurarse la diversidad lingüística se empleó la geolocalización de Twitter. Los organizadores definieron cuatro regiones de interés: Portugal y tres regiones oficialmente bilingües y recogieron tweets localizados en: Girona (Cataluña), Lugo (Galicia) y Gipuzkoa (País Vasco). Se recolectaron 10000 tweets

en cada una de las zonas bilingües y 5000 geolocalizados en Portugal, hasta completar los 35.000 tweets que forman el corpus.

Esta colección de tweets fue anotada manualmente por personas capaces de hablar con un nivel competente en al menos tres lenguas. Lo cual les permite anotar de manera experta tweets escritos en cualquiera de las dos lenguas de los territorios bilingües además de tweets escritos en inglés.
Un 10 % de los tweet fueron anotados por una segunda persona. El acuerdo entre los anotadores oscila entre el 96 % en el caso de la región catalana y el 88 % de la región gallega. Lo cual indica la dificultad de distinguir entre ciertos idiomas incluso para las personas.

A los anotadores se les instruyó para que ignoraran hashtags (#), menciones a nombres de usuarios (@) y los nombres propios de personas, organizaciones, eventos, fechas, cantidades, etc., lo que genéricamente se denominan entidades nombradas, que pudieran estar en otro idioma por ejemplo el tweet: *"14 de Febrero cita romantica con 'the walking dead'."* se considera un tweet escrito solo en castellano a pesar de que aparezcan palabras en inglés.

Se han definido ocho etiquetas de clase: *eu* para euskera, *ca* para catalán, *gl* para gallego, *es* para castellano, *pt* para portugués, *en* para inglés, *und* para aquellos tweets que no fuera posible determinar en que idioma fueron escritos y *other* para aquellos tweets escritos en un idioma que no se contempla en la tarea, como alemán o francés.
Los tweets mutilingües fueron etiquetados concatenando los posibles idiomas en los que estaban escritos utilizando el símbolo + para unir los diferentes idiomas presentes, así por ejemplo un tweet escrito parcialmente en euskera y parcialmente en castellano se etiquetaría como *es+eu*.
Para el caso de tweets ambiguos, en los cuales el anotador no podía diferenciar el idioma por la similaridad entre los mismos, se etiquetaban con ambos idiomas separando los idiomas utilizando el símbolo / , por ejemplo un tweet escrito en catalán o en castellano se etiquetaría como *es/ca*.

La distribución de idiomas en el corpus puede verse en la tabla 3.1.

Debido al fuerte desbalanceo del corpus, el 61.22 % está escrito en castellano, se empleó una métrica que evitaba favorecer aquellos sistemas que identificaran mejor los idiomas cuya probabilidad a priori fuera mayor. El *ranking* se hizo empleando la métrica F1-macro

Idioma	Porcentaje
Castellano (es)	61.22 %
Portugués (pt)	12.35 %
Catalán (ca)	8.46 %
Inglés (en)	5.63 %
Gallego (gl)	2.75 %

Idioma	Porcentaje
Undefinido (und)	2.25 %
Euskera (eu)	2.16 %
Multilingüe (a+b)	2.14 %
Ambiguo (a/b)	1.79 %
Otros	1.26 %

Tabla 3.1: Distribución del idioma en que estaban escritos los tweets del corpus recolectado para la tarea TweetLID.

definida en 3.1, (la notación es equivalente a la descrita en el apartado 2.3.1).

$$F-1 = \frac{1}{|C|} \sum_{i \in C} \frac{2 \cdot TP_i}{2 \cdot TP_i + FP_i + FN_i} \tag{3.1}$$

Para evaluar correctamente los tweets monolingües y multilingües los autores de la tarea propusieron los siguientes criterios de evaluación: si un tweet es monolingüe pero el sistema lo identifica como multilingües, siendo una de las clases la correcta, se incrementa tanto la cuenta de TP como la de FP, en el caso de los tweets etiquetados manualmente como multilingües se sigue la misma política para cada uno de los idiomas presentes, finalmente en el caso de los tweets ambiguos se considera un TP si el idioma identificado se encuentra entre el listado de los posibles idiomas. Para más información consultar el artículo de Zubiaga et al. [84].

El conjunto de datos se dividió de manera aleatoria en dos: el primero de ellos compuesto por 14.991 tweets para que los participantes entrenaran su sistema y 19.993 tweets para evaluar el comportamiento de los sistemas propuestos.
Dos modalidades para participar en el concurso fueron definidas:

- *Con restricciones*: los participantes únicamente podían utilizar el texto de los tweets etiquetados que proporcionaba la organización para entrenar sus modelos.
- *Sin restricciones*: en la cual los participantes podían utilizar tanto los tweets etiquetados de la modalidad con restricciones, como cualquier recurso adicional que consideraran apropiado para resolver la tarea.

Centraremos nuestros esfuerzos en solucionar la tarea sin restricciones, empleando tanto el corpus de entrenamiento como información externa que describiremos a continuación.

3.4. Modelo propuesto

En la línea del trabajo desarrollado por Goldszmidt et al. [26], nuestro sistema utiliza datos de la Wikipedia[6] como fuente de conocimiento fundamental para construir nuestro sistema identificador de idioma.

Se desarrollaron dos modelos para abordar esta tarea.

El primero de ellos basado en un lenguaje estadístico siguiendo la filosofía descrita en el apartado 2.1.3. La idea era entrenar tantos modelos de lenguajes estadísticos como idiomas queríamos identificar a partir del corpus extraído de la Wikipedia. Este modelo identificaría el idioma de un tweet, minimizando la perplejidad del mismo contra distintos modelos de lenguaje empleados. Para entrenar este sistema utilizamos el paquete SRLIM [74], entrenando modelos de 4-gramas basados en caracteres y basados en palabras. Sin embargo, la complejidad computacional de este modelo nos hizo descartarlo. Lo incluimos por completitud.

El segundo modelo que propusimos, y que finalmente constituyó nuestra participación en el concurso, empleaba una máquina de soporte vectorial, descrita en el apartado 2.2.1. En este modelo se adaptó un algoritmo de clasificación binaria, como son las máquinas de soporte vectorial, para conseguir un clasificador multiclase. Para lo cual se optó por seguir una aproximación *one-vs-one*. El número de clasificadores que se entrenó fue determinado por el número de etiquetas distintas vistas en el corpus de entrenamiento, en este caso fueron un total de 36 clasificadores. En la aproximación que se presentó se considera por ejemplo la etiqueta *es*+*en* como una clase totalmente diferente a *es* y *en*.
Esta aproximación presenta varias limitaciones: aquellas combinaciones no vistas durante la fase de entrenamiento no pueden ser predichas por nuestro sistema y las combinaciones poco frecuentes no estarán bien representadas. En contrapartida, presenta la ventaja que no necesita ajustar empíricamente umbrales para predecir etiquetas monolingüe o bilingües. Resumiendo, se trata de una aproximación competitiva en la literatura y por lo tanto decidimos optar por esta aproximación en nuestro modelo.

[6] https://en.wikipedia.org

En una fase preliminar intentamos desarrollar el modelo empleando como clasificador una aproximación basada en los k-vecinos más próximos, sin embargo los resultados no fueron prometedores por lo que se decidió no utilizar esta aproximación.

El resto de los detalles del entrenamiento, así como la selección de los parámetros del modelo se describen más extensamente en el apartado 3.5.

3.4.1. Extracción de información no restringida

Los textos que podemos extraer de la Wikipedia son de naturaleza muy variada, tanto por los contenidos como por los estilos de escritura de los distintos autores. Disponemos de textos para todos los idiomas que se pretenden identificar en la tarea.

La Wikipedia es de dominio público y puede descargarse en formato XML, lo cual nos ha permitido extraer la información con facilidad.[7] Los textos descargados directamente de la Wikipedia contienen además de las etiquetas propias de un fichero XML, etiquetas de marcado internas de la Wikipedia (similares a las etiquetas empleadas en LaTeX). Por lo tanto, antes de entrenar ningún modelo con estos datos, necesitamos preprocesarlos. El preproceso que se propone consistió en eliminar todas estas etiquetas de marcado para quedarnos de este modo con texto plano.
El tamaño de los datos disponibles en la Wikipedia varían en función del idioma oscilando entre los 11 GB de texto plano en inglés y los 565 MB del gallego.

Para entrenar los modelos propuestos, hemos recolectado textos en los distintos idiomas que debemos reconocer en la tarea: inglés, catalán, euskera, gallego, español y portugués; así como textos escritos en alemán, francés e italiano, que utilizaremos para entrenar el idioma fuera de la tarea *other*.

A partir del texto plano se creó una bolsa de palabras para entrenar nuestro sistema de aprendizaje automático. Únicamente entrenaremos con las palabras distintas más frecuentes en cada idioma. Definiremos un umbral variable para cada idioma que determine el número mínimo de repeticiones en el corpus para que una palabra forme parte de nuestro conjunto de entrenamiento. Estableceremos el umbral empíricamente, que nos

[7] Para más información acerca de las descargas de la información contenida en la Wikipedia consultar el siguiente enlace https://en.wikipedia.org/wiki/Wikipedia:Database_download.

	Idiomas de la tarea					
	Catalán	Castellano	Euskera	Gallego	Inglés	Portugués
Nº de palabras	55.320	132.968	25.474	20.261	346.658	78.848

	Idiomas fuera de la tarea		
	Alemán	Francés	Italiano
Nº de palabras	275.784	383.415	156.493

Tabla 3.2: Talla del vocabulario extraído de la Wikipedia

permitiera realizar la experimentación en un tiempo razonable dentro de los plazos del concurso. En la tabla 3.2 podemos ver el número de palabras por idioma que se han obtenido a partir de los datos de la Wikipedia.

La selección de palabras con más repeticiones tiene como efecto una *limpieza* del corpus en los idiomas en los que se descartan más palabras, como puede ser el inglés o el castellano.
Además mantuvimos las *stopwords* o palabras vacías, porque estas palabras muy frecuentes en su idioma puede ser relevantes para la identificación del mismo.

3.5. Entrenamiento

En este apartado describiremos la fase de entrenamiento que nos ha permitido ajustar el modelo propuesto a los requisitos funcionales de la tarea. Se describirá, por lo tanto, el proceso de experimentación incluyendo el preprocesado de los datos, la selección del clasificador así como el ajuste de sus parámetros.

Uno de los principales problemas, como ya hemos descrito previamente, es el de los errores gramaticales que cometen los usuarios en las redes sociales. Los usuarios de Twitter tienden a aplicar con más laxitud las reglas gramaticales que los usuarios de la Wikipedia.
Uno de los errores gramaticales más comunes es el la repetición de caracteres en una palabra, a veces incluso los usuarios cometen este error para enfatizar el mensaje que quieren transmitir. Para intentar paliar este problema se propone un sencillo preprocesado basa-

do en la eliminación de los caracteres repetidos tres o más veces.
Además, se eliminaron las menciones a usuarios, enlaces y fotografías incluidas en el texto del tweet, puesto que se trata de información que no ayuda a determinar el idioma en el que se escribió el tweet. Hicimos un proceso de experimentación gradual limpiando cada vez más el corpus de entrenamiento y el mejor sistema que obtuvimos fue el que combinaba todo el preproceso aquí descrito.
A pesar de tratarse de un preproceso sencillo, nos permitió mejorar levemente el rendimiento del modelo propuesto como se comentará en la tabla 3.3.

Para el desarrollo del sistema empleando el *toolkit scikit-learn* [58] que integra distintos algoritmos de aprendizaje automático del estado del arte, lo cual nos ha permitido explorar distintos algoritmos y configuraciones de los mismos para abordar la tarea.

Se emplearon dos conjuntos de datos para entrenar el modelo propuesto: los tweets de entrenamiento facilitados por la organización de la tarea y los datos de la Wikipedia. Para la representación tanto de los tweets como del texto extraído de la Wikipedia exploramos dos posibilidades, una basada en bolsa de palabras y otra basada en bolsa de caracteres, ponderadas en ambos casos utilizando la técnica tf-idf descrita en el apartado 2.1.4. Por cada tweet se generaron n muestras de entrenamiento, tantas como palabras formaran dicho tweet. Tras la vectorización se obtuvo un vector de 240.173 características. Sin embargo, debido a la similitud entre los idiomas estudiados, la aproximación basada en bolsa de caracteres se comportaba mucho peor durante esta fase de experimentación, como se puede observar en la tabla 3.3, por lo tanto se descartó, centrándonos en una aproximación basada en bolsa de palabras.
Estos dos corpus son los únicos recursos empleados para entrenar nuestro sistema. Por lo tanto el único recurso externo que empleamos es el texto de la Wikipedia.
Con estos recursos se entrenó una máquina de soporte vectorial, se consideraron distintos *kernels* durante la fase de entrenamiento, pero finalmente se optó por un *kernels* lineal.

Tras seleccionar el clasificador se ajustó el parámetro que regula la tolerancia de la máquina de soporte vectorial realizando una validación cruzada con cinco particiones (*5-fold*). En cada experimento se entrenó la SVM con la Wikipedia y con una de las cinco particiones aleatorias de los tweets de entrenamiento y exploramos los valores de C en el rango $C \in \{2^{2i-1} | -3 < i < 15, i \in \mathbb{Z}\}$. Este parámetro se ajustó empleando la macro

Descripción del sistema	Macro P	Macro R	Macro F1
Bolsa de palabras de la Wikipedia	58.89 %	42.61 %	49.43 %
Bolsa de caracteres de la Wikipedia y tweets de entrenamiento	60.87 %	37.77 %	46.62 %
Bolsa de palabras de la Wikipedia y tweets de entrenamiento	62.57 %	57.79 %	60.08 %
Bolsa de palabras de la Wikipedia y tweets de entrenamiento y preprocesado	67.77 %	56.68 %	61.73 %
Bolsa de palabras de la Wikipedia y tweets de entrenamiento, preprocesado y ponderación de las clases	74.16 %	70.07 %	72.06 %

Tabla 3.3: Evaluación de los sistemas durante la fase experimental realizando una validación cruzada con cinco particiones.

F-1, el valor final seleccionado para C fue 2.

Los resultados de más relevantes de este proceso de experimentación se muestran en la tabla 3.3. El sistema que finalmente fue enviado para su evaluación en el concurso se destaca en negrita en la última fila de esta tabla.

Disponíamos de un vocabulario infrarepresentado tanto para el euskera como para el gallego, como hemos visto en la tabla 3.2, lo cual provocaba que el clasificador no fuera capaz de aprender correctamente las muestras de entrenamiento de estos idiomas. En la tabla 3.4 se pueden observar las variaciones en el comportamiento de nuestro sistema para el caso de los tweets monolingües.
Por lo tanto se decidió ponderar las muestras de estas dos clases de modo que el peso asociado a estas muestras fuera el doble que el resto. En la tabla 3.3 se observa la diferencia entre el sistema entrenado de modo que todas las muestras pesen lo mismo (cuarta fila) y el uso de la ponderación de las muestras de euskera y gallego (última fila), lo cual ayudó al mejorar el rendimiento del clasificador como pue.

Idioma	Valor-F1
Castellano	0.8835
Catalán	0.9065
Euskera	0.5990
Gallego	0.5011

Idioma	Valor-F1
Inglés	0.7443
Portugués	0.8856
Ambiguo	0.7002
Indefinido	0.1768

Tabla 3.4: Evaluación por idioma durante la fase de entrenamiento realizando una validación cruzada con cinco particiones

En definitiva, tras esta fase de entrenamiento, se envió al concurso un sistema basado en SVM con un *kernel* linear entrenado utilizando una aproximación de bolsa de palabras con datos de la Wikipedia y con los tweets proporcionados por la organización. Los datos se preprocesaron y las muestras de euskera y de gallego se ponderaron de modo que fueran más significativas.

3.6. Evaluación en la tarea

En la tarea sin restricciones propuesta en el TweetLID, participaron nueve equipos que presentaron aproximaciones muy dispares tanto en los modelos como en los recursos empleados. Entre los recursos externos que los equipos emplearon podemos encontrar: texto extraído de periódicos, de la Wikipedia, o del Europarlamento[8]. Los algoritmos de clasificación automática más comunes fueron las máquinas de soporte vectorial y Naïve Bayes. Los detalles de estos sistemas pueden encontrarse en el artículo de Zubiaga et al. [84], donde también se puede encontrar una evaluación pormenorizada de los distintos sistemas que queda fuera del ámbito de este trabajo.

La organización propuso como referencia dos sistemas base para evaluar los sistemas presentados a la tarea: las etiquetas que asigna Twitter en los meta-datos a cada tweet y Textcat[32], un sistema que se considera estado del arte para textos normativos de cierta longitud. Los resultados que obtiene estos dos sistemas base, denominados *baseline 1 y 2* en la tabla 3.5, indican la complejidad de la tarea.

[8] http://www.statmt.org/europarl/

Rank	Descripción del sistema	Macro P	Macro R	Macro F1
1	Citrius-imaxin II	80.2 %	74.8 %	75.3 %
2	ELiRF @ UPV II	73.7 %	72.3 %	69.7 %
3	**ELiRF UPV I**	**74.2 %**	**68.6 %**	**68.4 %**
4	Citrius-imaxin I	69.6 %	65.9 %	65.5 %
5	LYS @ UDC	68.2 %	68.8 %	58.1 %
Baseline 1	Twitter	45.7 %	49.8 %	46.3 %
Baseline 2	Textcat	58.6 %	48.0 %	44.7 %

Tabla 3.5: Evaluación de los sistemas en el concurso.

El sistema aquí presentado quedó en tercera posición. En la tabla 3.3 se listan las métricas de los cinco mejores sistemas presentados al TweetLID así como la de los dos sistemas base descritos anteriormente. Todos los sistemas presentados superaron estos sistemas base, lo cual indica la importancia del desarrollo de sistemas que aborden el problema de LID para Twitter.

En la evaluación realizada por los organizadores de la tarea se confirma que la longitud del tweet es una característica determinante para la identificación correcta del texto. Para el caso de tweets de menos de 20 caracteres la mediana de los valores-F1 que obtuvieron los participantes es aproximadamente el 0.6, mientras que a partir de tweets de 81 caracteres la mediana sube por encima del 0.9.
De igual modo, como era de esperar la tarea de identificar múltiples idiomas dentro de un mismo tweet es más compleja y el comportamiento de todos los sistemas presentados cayó bruscamente pasando de una mediana próxima al 0.6 de valor-F1 en tweets monolingües a un valor-F1 aproximadamente del 0.35 en el caso de tweets multilingües.
Coincidimos con la organización en que la diversidad de sistemas y recursos que se han presentado será de utilidad como punto de referencia para trabajo futuro.

3.7. Conclusiones y trabajo futuro

En esta capítulo hemos presentado nuestra participación en el TweetLID empleando una aproximación basada en máquinas de soporte vectorial y empleando como recursos externos textos de la Wikipedia.
El modelo aquí presentado quedó en tercera posición obteniendo un macro valor-F1 de 68.4 %.
A pesar de lo que comúnmente podemos encontrar en la literatura, no puede asumirse que la tarea de identificación de idioma se encuentre resuelta en el caso de textos cortos agramaticales y multilingües extraídos de redes sociales, porque como se desprende de los resultados de la evaluación todavía queda mucho margen de mejora, puesto que el mejor sistema únicamente es capaz de obtener un macro valor-F1 de 75.3 %.

Esta tarea se ha centrado en los retos más complejos de la identificación del idioma, pero la evolución del uso de las redes sociales y del multilingüísmo en Internet nos indican que deberemos enfrentarnos a este tipo de escenarios constantemente.

Asumiendo por lo tanto que se trata de una tarea relevante, para el futuro proponemos distintas consideraciones:

- Desarrollar un sistema de clasificadores monolingües capaces de emitir etiquetas multilingües combinando la certidumbre de las predicciones que emiten cada uno de los clasificadores.
- Incluir fuentes de información de distinta naturaleza, que capturen más fielmente el estilo de escritura (con sus peculiaridades) que emplean los usuarios en las redes sociales.
- Balancear los corpus de entrenamiento, completando el vocabulario de palabras en gallego y en euskera fundamentalmente.
- Utilizar la geolocalización que proporciona Twitter en sus meta-datos como método de desambiguación o como una característica más de entrenamiento.

4

Análisis de sentimientos

En este capitulo se describe la tarea de la identificación de los sentimientos presentes en un texto. La comprensión de la intencionalidad del emisor de una sentencia es una tarea ardua, de gran impacto en distintas aplicaciones prácticas.

El capítulo se organiza del siguiente modo: en primer lugar introduciremos el problema que queremos abordar con las dificultades e impacto que conlleva; en la sección 4.2 haremos un resumen de las aproximaciones más relevantes que podemos encontrar en la literatura; continuaremos describiendo en la sección 4.3 los detalles de las dos tareas relacionadas con el análisis de sentimientos organizadas para el SemEval (*Semantic Evaluation Exercises*) en la edición del 2015 en las que hemos participado; los modelos con los que se participó en SemEval 2015 se presentarán en la sección 4.4; posteriormente la evaluación de los sistemas se presentará en las secciones 4.5 y 4.6; finalmente, analizaremos los resultados obtenidos y propondremos futuras líneas de trabajo para continuar mejorando los retos que esta tarea presenta.

4.1. Introducción al problema

Las personas somos capaces de comprender el mensaje transmitido por nuestro interlocutor, incluso en los casos en los que se emplee lenguaje figurado. Sin embargo, trasladar

ese conocimiento a un sistema computacional es una tarea ardua.
La tarea del análisis de sentimientos, *Sentiment Analysis (SA)*, ha sido estudiada intensamente en el área del NLP y tiene múltiples aplicaciones comerciales.
Citamos aquí la definición extraída del trabajo de Liu [39] que sienta las bases del estudio de SA: "El análisis de sentimientos, también denominado minería de opiniones, es el campo de estudio que analiza las opiniones, sentimientos, evaluaciones, actitudes y emociones que las personas emiten sobre distintas entidades como: productos, servicios, organizaciones, individuos, problemas, eventos distintos temas y sus atributos; aglutinando un conjunto de problemas variados."
En este capítulo nos centraremos únicamente en detectar de forma automática la polaridad expresado en tweets (polaridad positiva, negativa o netra).

Como ya se ha mencionado en el capítulo anterior, los usuarios de redes sociales como Twitter generan una cantidad ingente de información que permite la investigación de nuevos retos científicos.

Twitter [1] es un servicio de microblogging, que según las últimas estadísticas tiene 284 millones de usuarios activos, el 77 % fuera de EE.UU., que generan 500 millones de tweets diarios en 35 idiomas distintos. Es decir, se producen 5.700 tweets por segundo llegando a picos de 43.000 tweets por segundo.
Se estima [2] que el 50.9 % del contenido es información con cierto nivel de relevancia y se ha demostrado una herramienta capaz de movilizar opiniones dentro y fuera de Internet.
La reputación de marcas y personas, así como el impacto de campañas publicitarias, la evaluación de productos o servicios puede determinarse a partir de los datos publicados en Twitter.
En definitiva, las estadísticas listadas en esta apartado justifican el interés que despierta descubrir la opinión general sobre un tema concreto, puesto que esta opinión es de gran valor estratégico, pero el volumen de datos que los usuarios producen hace inviable realizar este trabajo de forma manual. Necesitamos, por lo tanto, sistemas capaces de automatizar este proceso.

[1] En el siguiente enlace se describe la información referente a Twitter: About twitter,inc. https://about.twitter.com/company. Acceso: 30-12-2014.

Desde 1998 se celebra la competición SemEval (*Semantic Evaluation*)[2] auspiciada por la ACL (*Association for Computational Linguistics*), cuyo objetivo es evaluar la capacidad de modelos computacionales para comprender el significado de un texto.
En esta competición se desarrollan diferentes tareas con el objetivo de abordar problemas emergentes en NLP. Estas tareas han evolucionado con cada edición planteando problemas más desafiantes, incorporando nuevos medios y formas de comunicación como son las que podemos encontrar en las redes sociales.
Durante los últimos años se ha propuesto una tarea de SA[49, 69, 70] sobre contenido literal (tarea 10), y en la pasada edición celebrada en 2015 se incorporó una tarea centrada en el problema de la detección de la polaridad en presencia de lenguaje figurado [21] (tarea 11).
Nos planteamos un modelo capaz de abordar, con ligeras modificaciones, las dos tareas de SA propuestas para la edición de 2015[50]. Nuestra aproximación utiliza técnicas de aprendizaje automático, en concreto el formalismo de las máquinas de soporte vectorial entrenadas de forma supervisada. Nuestro modelo explora las siguientes características:

- N-Gramas.
- Diccionarios de polaridad.
- Marcas textuales empleadas en Twitter.
- ...

Las tareas abordadas se describirán en el apartado 4.3 y los modelos que hemos propuesto se detallarán en el apartado 4.4, la tarea 10 se consideró como un problema de clasificación mientras que la tarea 11 fue tratado como un problema de regresión. En ambos casos se trató como un problema de aprendizaje supervisado sin restricciones, por lo tanto empleamos tanto los córpora etiquetados por los organizadores como recursos externos.

4.2. Estado del arte

La proliferación de *blogs*, redes sociales y las páginas de reseñas de productos y servicios han motivado el desarrollo de sistemas automáticos capaces de recuperar información, identificar la polaridad y en términos más generales comprender el significado de un texto.
El análisis de sentimientos ha sido ampliamente estudiado en la última década aplicando

[2] SemEval https://en.wikipedia.org/wiki/SemEval Acceso: 04-02-2016

diferentes aproximaciones. La definición más extendida de la tarea es aquella que se centra en clasificar los textos en tres categorías: textos positivos, negativos y neutros. Aunque en los últimos años podemos encontrar trabajos como el de Socher et al. [73], dónde se explora la predicción de la polaridad resente en un texto con una mayor granularidad.
Los primeros trabajos que se realizaron sobre el análisis de sentimientos emplearon tanto aproximaciones supervisadas [57] como aproximaciones no supervisadas[75]. El trabajo de Pang et al. [57] evaluó el rendimiento de distintos algoritmos de clasificación sobre un conjunto de datos formado por críticas de películas. Por otra parte, el trabajo de Whitelaw et al. [79] desarrolló una taxonomía con el objetivo de identificar adjetivos calificadores y adverbios intensificadores, que empleó sobre el mismo conjunto de datos mejorando los resultados que se habían obtenido previamente.
En el trabajo de Pang and Lee [56] podemos encontrar un amplio estudio de las distintas técnicas que se han empleado para realizar SA sobre textos publicados en Internet.
Por las aplicaciones de esta tarea, se ha invertido mucho esfuerzo en aplicar el conocimiento adquirido a los textos publicados en redes sociales.
Recientemente, distintos trabajos han tratado esta tarea utilizando distintas aproximaciones de aprendizaje automático como pueden ser: SVM, máxima entropía, árboles de decisión, naive bayes, etc [8, 53, 82].
Se puede encontrar una recopilación pormenorizada en el trabajo de Liu and Zhang [40] que presenta los trabajos más relevantes de SA del estado del arte. Los mejores sistemas, presentados en esta recopilación, que abordaron el problema de SA consiguieron alcanzar un valor-F1 próximo al 70 %, lo cual todavía nos permite un margen de mejora significativo.

Otra línea de investigación relevante para el SA consiste en el desarrollo de diccionarios de polaridad o lexicones y han demostrado ser de utilidad para esta tarea. Sin embargo, la mayoría de los lexicones están disponibles en inglés [33, 80, 46], sin bien podemos encontrar algunos diccionarios de polaridad más marginales en otros idiomas [60, 78].

El estudio de SA en Twitter es mucho más reciente. La plataforma surgió en 2006 y los primeros trabajos que hacen uso de datos extraídos de Twitter fueron publicados en el 2009, cuando la red social comenzó a popularizarse.
Vinodhini and Chandrasekaran [77] recoge el trabajo que se realizó para aplicar la tarea de SA en Twitter.

El lenguaje permite usos creativos para expresar ideas u opiniones, lo cual complica la tarea de SA, puesto que el sentido de las palabras no coincide con el que ha querido su autor/a transmitir. El estudio del uso del lenguaje figurado juega un papel clave en el SA puesto que puede invertir la polaridad de una frase como demuestra el trabajo de Maynard and Greenwood [44].
Por otra parte, la detección de la ironía, el sarcasmo y otras formas de lenguaje figurado constituyen una tarea por sí misma. En la literatura distintos trabajos proponen sistemas capaz de detectar el uso de lenguaje no literal. [13, 66, 30]

Como se ha introducido previamente, desde SemEval se proponen diversas tareas de SA en Twitter [49, 69] que atraen el interés de la comunidad científica. En los siguientes apartados describiremos las características de la edición del 2015 así como una descripción de nuestra participación diversas tareas de SemEval 2015 (tarea 10 y 11).

4.3. Descripción de la tarea

En la edición SemEval 2015 se propusieron dos tareas relacionadas con SA:

- **Tarea 10**: El objetivo de esta tarea consistía identificar de forma automática el sentimiento predominante en un tweet. Tres categorías de sentimientos fueron tenidos en cuenta en esta tarea: positivos, negativos y objetivos/neutros. Puede encontrase más información de la tarea en el artículo de Rosenthal et al. [70].
- **Tarea 11**: El objetivo de esta tarea era asignar un valor de polaridad en tweets escritos con lenguaje figurado (ironía, sarcasmo y metáfora) y por lo tanto estudiar el impacto del lenguaje figurado en la detección de la polaridad de un tweet. A diferencia de la tarea 10, se consideró la intensidad del sentimiento presente en un tweet en un rango entre $[-5, \ldots, 5] \in \mathbb{R}$, siendo -5 la polaridad más negativa y 5 la más positiva. Una descripción detallada de la tarea se encuentra en el artículo de Ghosh et al. [21].

La organización seleccionó textos publicados en Twitter. Esta plataforma de microblogging presenta problemas propios de las redes sociales: lenguaje abreviado, argot, etc., y otros propios como: la restricción al tamaño máximo de los mensajes -nunca más de 140 caracteres-, así como su idiosincrasia: *retweets*,*hashtags*, menciones a usuarios, favoritos, etc.

Todos los tweets recogidos fueron escritos en inglés. Esta tarea se centra únicamente en la detección del sentimiento en inglés, y por lo tanto es una tarea monolingüe.

Para los córpora de la tarea 10 se seleccionaron tweets que expresaban algún tipo de sentimiento sobre un conjunto de entidades nombradas populares entre un rango de fechas siguiendo la aproximación propuesta por Ritter et al. [67], se utilizó una herramienta para la detección de entidades nombradas en Twitter[68]. Los organizadores emplearon el lexicón SentiWordNet 3.0 [3] para asegurarse de que al menos una palabra de cada tweet tuviera un sentimiento asociado, además el valor del sentimiento de la palabra polarizada en SentiWordNet debía superar el umbral 0.3. Esto supone introducir un sesgo en los córpora, pero los organizadores afirman que se conserva la variabilidad de los tweets[70].

Los organizadores proporcionaron tres córpora para el entrenamiento de los sistemas recogidos entre Enero de 2012 y Enero de 2013. Estos corpora se agruparon de la siguiente forma:

- **Train**: 7.236 tweets para entrenar el sistema. Estos tweets se crearon desde el 12 de Junio del 2011 y el 2 de Noviembre de 2012.
- **Dev**: 1.242 tweets para ajustar el sistema. Estos tweets se crearon desde el 11 de Julio del 2011 y el 2 de Noviembre de 2012.
- **Dev test**: 2.880 tweet para evaluar el sistema durante el desarrollo del mismo. Estos tweets se crearon entre el 6 de Enero de 2012 y el 2 de Noviembre de 2012.

Los sistemas se evaluaron empleando dos córpora: el primer corpus era totalmente nuevo y estaba constituido por 3092 tweets extraídos de Twitter, un porcentaje de los tweets eran sarcásticos; y el segundo corpus compuesto por datos empleados en las ediciones anteriores: SMS y tweets que formaban parte del conjunto de datos de la edición de 2013 y, finalmente, un conjunto de tweets (literales y con presencia de sarcasmo) y textos extraído de las *LiveJournal*[4] empleados en la edición de 2014.
Los tweets de esta tarea fueron etiquetados manualmente [5] utilizando el servicio de Amazon *Mechanical Turk*[6].
El número de tweets aquí descritos por córpora corresponde al número de tweets que fueron recuperados cuando se desarrolló este sistema. Algunos tweets fueron borrados

[3] http://sentiwordnet.isti.cnr.it/download.php
[4] http://www.livejournal.com/
[5] Para más información sobre el proceso de etiquetado consultar el artículo de Rosenthal et al. [69].
[6] https://www.mturk.com/mturk/welcome

por sus autores y por lo tanto, ya no estaban disponibles.
La figura 4.1 muestra la distribución de la polaridad en el periodo temporal en el que fueron escritos los tweets que fueron considerados en la tarea. No se observa variación significativa entre las distintas clases lo que podría ser un indicio de la presencia de eventos que polarizaran los sentimientos expresados por los usuarios de Twitter en un sentido o en otro.

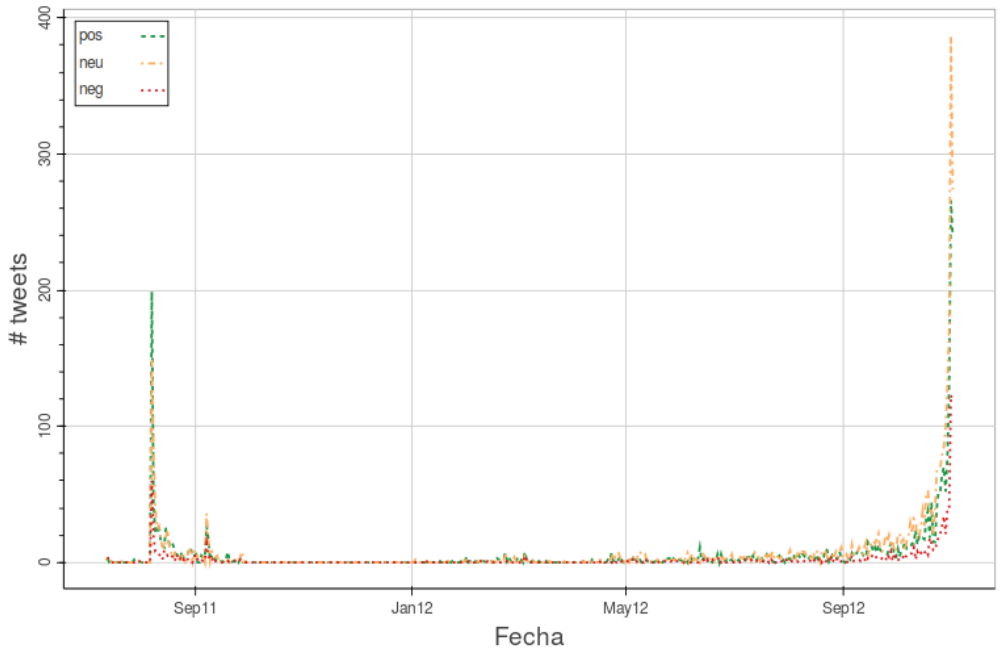


Figura 4.1: Distribución de la polaridad en función de la fecha de creación de los tweets de la tarea 10.

En la figura 4.2 podemos ver la distribución de la polaridad en el corpus de *train*, *dev* y *devtest*. En promedio un 16.53 % son tweets negativos, un 45.75 % son neutrales y un 37.72 % son positivos. Esta distribución de la polaridad en el corpus provocará que a nuestro sistema le cueste más identificar correctamente los tweets pertenecientes a la clase negativa, pues es la clase de la que menos muestras hemos visto durante el entrenamiento.

El vocabulario del corpus de *train* está formado por 25.973 palabras, el de *dev* por 6.700 palabras y el de *devtest* por 13.672 palabras tras eliminar las *stopwords*, es decir aquellas pa-

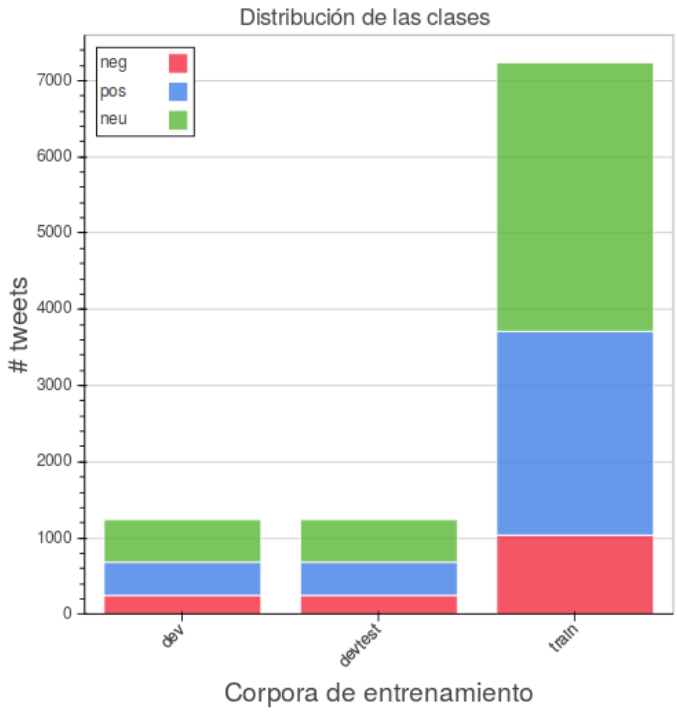


Figura 4.2: Distribución de la polaridad en los corpora de entrenamiento, dev y devtest de la tarea 10.

labras sin significado como artículos, pronombres, etc. Del vocabulario del test no hemos visto 7.611 palabras en el entrenamiento, esto es el 55.67 %, por lo tanto el tratamiento de las palabras no vistas puede determinar la comportamiento de nuestro sistema.

Por otra parte, los tweets seleccionados para abordar la tarea 11 se recolectaron inicialmente a partir de un conjunto de *hashtags*, como *#irony* y *#sarcasm*, para después ampliar el número de tweets que formaban el corpus siguiendo la aproximación de Li and Ghosh [37] basada en un análisis *Latent Semantic Analysis (LSA)*.
Los tweets se recogieron entre el 1 y el 30 de Junio de 2014.

El etiquetado de los tweets de la tarea 11 lo realizaron siete anotadores del equipo de organización de la tarea, tres de ellos eran hablantes nativos de inglés. La puntuación final que se le asignó a cada tweet corresponde a la media ponderada de todos los anotadores,

considerando el doble de relevantes las puntuaciones dadas por los hablante nativos [7].

Los organizadores facilitaron estos tweets agrupados en tres córpora:

- **Trial**: 986 tweets, de los que no se recuperó la fecha de creación. Este primer conjunto de datos se recuperó seleccionando a usuarios de Twitter más propensos a emplear figuras retóricas como pueden ser cómicos. Los participantes dispusieron de este corpus en la fase inicial de la tarea para que pudieran desarrollar sus sistemas.
- **Train**: 6.744 tweets para entrenar los sistemas creados entre el 22 de Octubre de 2009 y el 10 de Junio 2014.
- **Test**: 3999 tweets publicados entre el 25 de Noviembre y el 2 de Diciembre de 2014.

De nuevo las diferencias entre el número de tweets aquí reportados y los citados por los autores depende de la disponibilidad de los tweets cuando se desarrollaba el sistema.

La figura 4.3 muestra la distribución de polaridad de los tweets de *train* y *trial* de la tarea 11. Por simplicidad hemos agrupado la polaridad de cada tweet a su valor entero. También en esta tarea se observa que las clases están desbalanceadas. En este caso, los tweets con polaridad muy negativa son la mayoría, lo cual provocará que el sistema que desarrollemos sea más propenso a errar cuando identifique tweets neutrales o positivos, ya que estos están peor representados. Sin embargo, esta distribución de probabilidad parece la usual cuando estamos tratando con tweets en los que encontramos figuras retóricas como la ironía o el sarcasmo.

El vocabulario de los corpus de *train* de la tarea 11 esta compuesto por 21.829 palabras, 4.689 palabras componen el conjunto de datos de *trial* y 14.456 palabras componen el *test*. De nuevo eliminamos las *stop words* para obtener los vocabularios de los córpora. Encontramos 8.821 palabras en el test no vistas durante el entrenamiento, que constituyen un 38.58 % del vocabulario. Por lo tanto, aunque en menor medida, también en esta tarea el tratamiento de las palabras no vistas puede determinar el comportamiento de nuestro sistema.

A continuación estudiamos que la distribución de las palabras en el vocabulario siga la ley de Zipf, descrito previamente en 2.1.2. Cabe recordar que la ley de Zipf caracteriza un corpus de lenguaje natural y se cumple si la frecuencia de cualquier palabra es inversa-

[7] Para más información sobre el proceso de etiquetado y la construcción de los corpora ver artículo de Ghosh et al. [21].

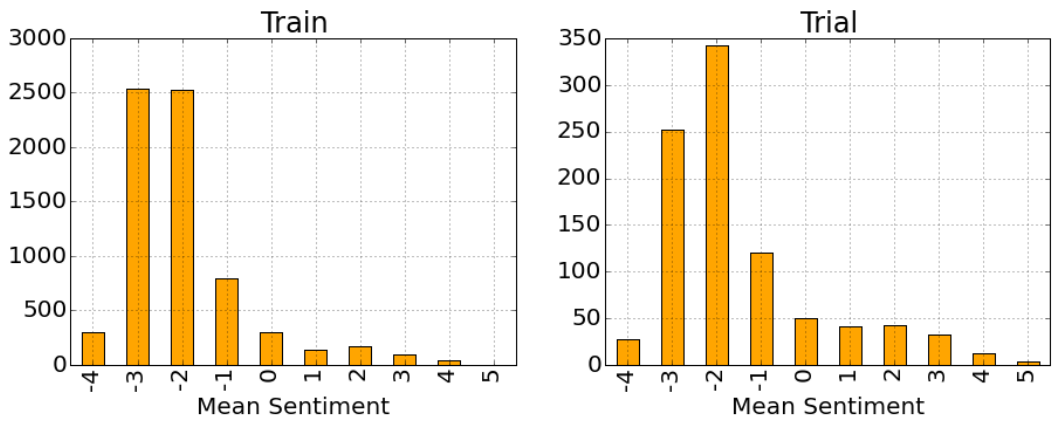


Figura 4.3: Distribución de la polaridad en los corpora de entrenamiento, dev y devtest de la tarea 11.

mente proporcional a su *ranking* ordenado por la palabra más frecuente. En la figura 4.4 vemos como efectivamente la distribución del vocabulario de los tweets de ambas tareas se aproxima a la distribución ideal.

Análogamente hemos estudiado la ambigüedad de las palabras en el corpus. En las figuras 4.5 y 4.6 podemos ver como las palabras con más *synsets* –conjuntos de significados– son las palabras menos frecuentes. Lo que nos indica que en estos córpora las palabras tienen, en general, poca ambigüedad por si mismas. Resulta interesante destacar que en los córpora de la tarea 11 los usuarios emplean palabras con menos sinónimos a pesar de tratarse de unos córpora dónde está muy presente la ironía y el sarcasmo.
Podemos destacar que las palabras con más de 60 *synsets* en ambas tareas son: *cut* y *break*; y que las palabras con más de 1.500 repeticiones en la tarea 10 son: *tomorrow*, *http* y *I* (ordenadas de menor a mayor frecuencia de aparición), mientras que la palabra con más repeticiones en la tarea 11 es *I*.

Por último, destacar que sólo el 19.98 % de los tweets de entrenamiento y el 20.31 % de los tweets de test tienen *hashtags* en la tarea 10 mientras que en la tarea 11, el 151.43 % de los tweets de entrenamiento (66.7 % en el corpus *train* y 163.05 % en el corpus *trial*) y el 88.24 % de los tweets de test contienen *hashtags*. Debido al proceso de recolección de los tweets diseñado por los organizadores de la tarea 11, es razonable encontrar muchos más tweets que emplean *hashtags*, sería interesante estudiar, sin las restricciones definidas por la organización, si el uso de *hashtags* es una característica propia del lenguaje figurado en

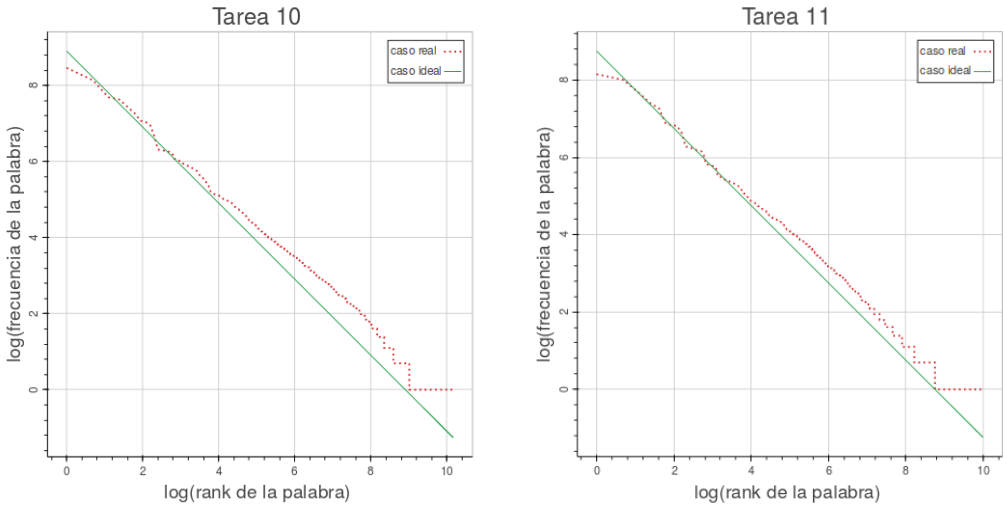


Figura 4.4: Distribución de las palabras siguiendo la ley de Zipf.

redes sociales. Los usuarios de Twitter emplean los *hashtags* para etiquetar el contenido de un tweet, por lo que es de esperar que su contenido sea relevante en la clasificación de la polaridad del tweet. Sin embargo los *hashtags* suelen concatenar varias palabras y la segmentación de las mismas constituye un problema en si mismo.

4.4. Presentación de la metodología propuesta

Como describimos en la sección 4.1, nos planteamos una metodología para construir modelos capaces de identificar el sentimiento presente en un tweet para las tareas 10 y 11 de SemEval. En ambos casos la metodología es muy similar y por esto la agrupamos en esta sección. En el apartado 4.5 describiremos los modelos que mejor se han comportado para cada tarea.

Sintetizando, nuestro proceso de trabajo ha consistido, tras estudiar el corpus como hemos visto en la sección anterior, entrenar distintos clasificadores usando fundamentalmente el texto del tweet y lexicones de polaridad. Se trata por lo tanto de un proceso de entrenamiento supervisado y restringido por el corpus proporcionado.

Dividiremos la metodología en dos partes: la extracción de características empleadas para entrenar los sistemas y la selección de los algoritmos de aprendizaje empleados para clasi-

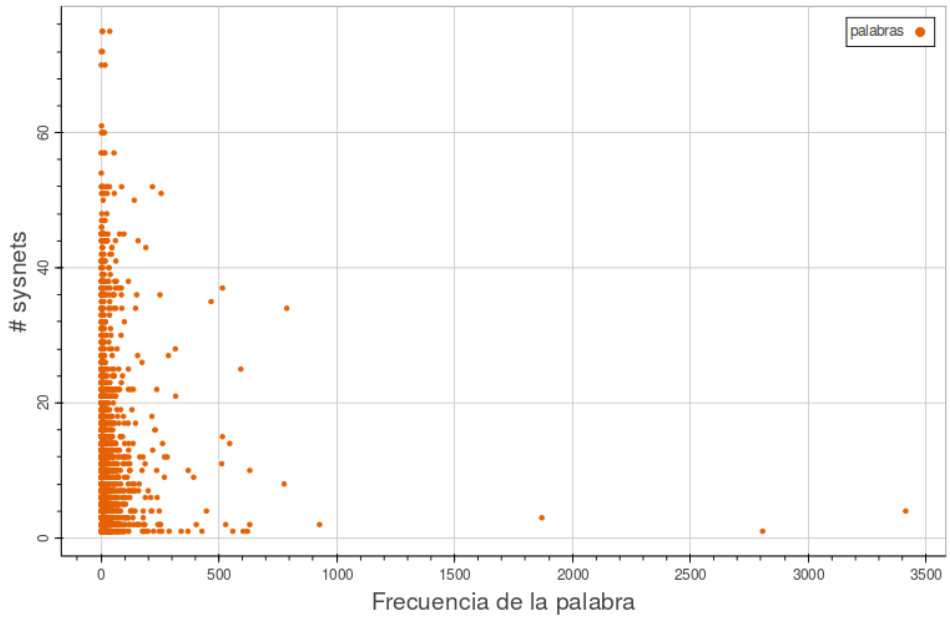


Figura 4.5: Número de significados posibles de cada palabra y el número de veces que se utiliza en el corpus de la tarea 10.

ficar las muestras.

4.4.1. Extracción de características

Hemos desarrollado nuestro sistema de modo que agrupa los tipos de características que podemos extraer de los tweets en función del tipo de vectorización que les aplicamos. Como los algoritmos seleccionados no pueden trabajar directamente con texto, las características extraídas de los córpora fueron agrupadas siguiendo estos criterios:

1. Características que vectorizamos usando los coeficientes tf-idf: en este grupo tenemos los n-gramas de palabras.
2. Características que no necesitamos vectorizar: fundamentalmente en este grupo de características incluimos los valores obtenidos a partir de diccionarios de polaridad o lexicones. Pero también incluimos en este grupo, características como: el número de palabras de un tweet completamente en mayúsculas, el número de *hashtags* presentes en un tweet, etc.

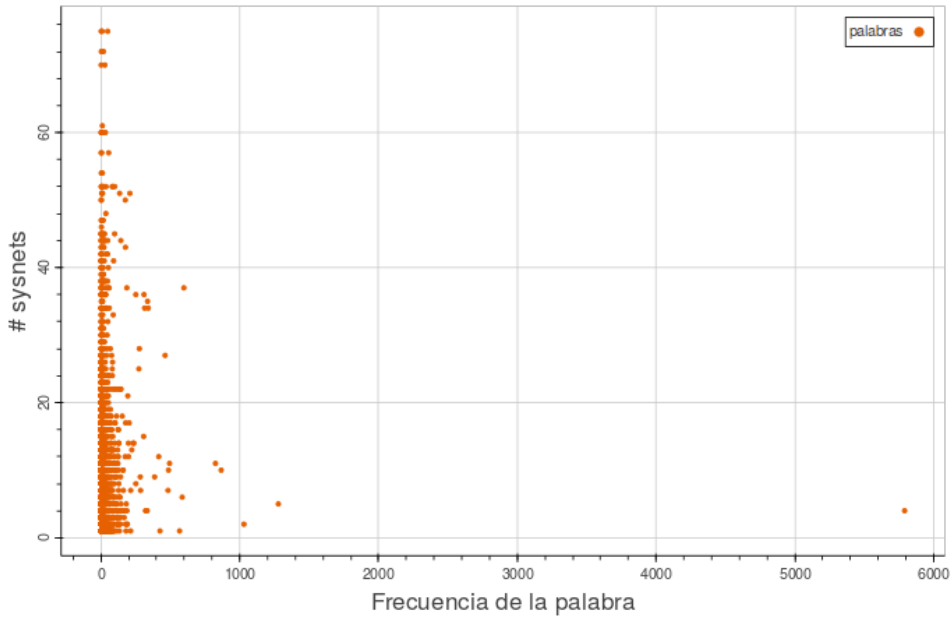


Figura 4.6: Número de significados posibles de cada palabra y el número de veces que se utiliza en el corpus de la tarea 11.

A continuación describiremos las características que fueron consideradas para desarrollar nuestros sistemas:

N-gramas. En primer lugar *tokenizamos* los córpora, esto es separar las palabras que componen una frase, teniendo en cuenta que la contracciones deben separarse formando dos palabras distintas, y eliminamos las palabras vacías (*stopwords*). Tras este preproceso extraemos los n-gramas de caracteres siguiendo una aproximación de bolsa de n-gramas. Representamos un tweet como un vector de coeficientes tf-idf. Para cada tarea adaptamos el rango de los n-gramas tenidos en cuenta el que mejor rendimiento obtuviese. En el caso de la tarea 10 el mejor caso corresponde a tener en cuenta n-gramas de 1 a 6 caracteres, mientras que en la tarea 11 el modelo se comporta mejor con n-gramas formados entre 3 y 9 caracteres.

Negación. La negación juega un papel fundamental en la detección de la polaridad, puesto que actúa, en muchos casos, invirtiendo la polaridad. Por lo tanto, tratamos la

negación siguiendo la aproximación de Pang et al. [57], etiquetamos con una marca especial cada palabra dentro de un contexto negativo. Definimos *contexto negativo* como el conjunto de palabras delimitado entre un adverbio de negación (*no*, *never*, *nothing*, *none*, etc.) y un signo de puntuación. Empleamos un listado [8] de adverbios negativos en inglés que hemos extendido para etiquetar los contextos negativos. Los resultados obtenidos en la fase de entrenamiento aplicando la gestión de la negación sólo mejoraban para la tara 10. Sin embargo, la tarea 11 no presentaba mejoras, e incrementaba la complejidad del sistema. Por lo tanto, estas características únicamente se aplicaron en la tarea 10.

Diccionarios de polaridad. Para extraer las características asociadas a estos lexicones, llevamos a cabo un preproceso de los tweets distinto al caso de los n-gramas y consistió en eliminar las palabras vacías, así como convertir todas las letras del tweet a minúsculas. Fueron considerados los siguiente cinco diccionarios de polaridad:

1. *Pattern*[18]: Para cada palabra en este lexicón encontramos el valor de polaridad y el de subjetividad de la misma, en un rango entre $[-1, \ldots, 1]$. Dado un tweet, calcularemos la polaridad y la objetividad de la frase sumando la de cada una de las palabras que la componen y normalizando por la longitud del tweet.
   $\frac{1}{|W|}\sum_{w \in W} Pattern_{polarity} w)$ y $\frac{1}{|W|}\sum_{w \in W} Pattern_{objectivity} w)$
2. *Afinn-111*[29]: Es un listado de palabras que tienen asociado un valor de polaridad entre $[-5, \ldots, 5]$. Para obtener la polaridad de un tweet simplemente sumamos la polaridad de cada una de las palabras del tweet que aparezcan en este lexicón y la normalizamos como se describió en el anterior lexicón.
   $\frac{1}{|W|}\sum_{w \in W} Afinn(w)$
3. *Jeffrey*[33]: Este lexicón tiene un listado de palabras positivas y otro de palabras negativas. Obtenemos dos valores numéricos asociados a cada tweet. Uno teniendo en cuenta las palabras positivas y otro teniendo en cuenta las palabras negativas.
   $\frac{1}{|W|}\sum_{w \in W} Jeffrey_{pos}(w)$ y $\frac{1}{|W|}\sum_{w \in W} Jeffrey_{neg}(w)$
4. *NRC*[48]: Análogamente, obtenemos un valor cuantitativo para cada tweet sumando la polaridad de cada palabra según este lexicón en un rango entre $[-1, \ldots, 1]$ y lo normalizamos por la longitud del tweet.
   $\frac{1}{|W|}\sum_{w \in W} NRC(w)$
5. *SentiWordNet*[4]: En este caso, para cada palabra podemos encontrar conjuntos de significados distintos (*Synsets* S), de modo que normalizaremos la puntuación ob-

[8] http://sentiment.christopherpotts.net/lingstruc.html

Tarea	*Tarea 10*		*Tarea 11*	
Lexicón	Train	DevTest	Train	Trial
Afinn	3.14 %	3.85 %	3.23 %	6.23 %
Pattern	4.28 %	5.21 %	7.06 %	9.62 %
SentiWordNet	45.21 %	51.26 %	32.22 %	48.60 %
Jeffrey	4.01 %	4.56 %	4.24 %	7.44 %
NRC	29.42 %	33.26 %	29.34 %	43.04 %

Tabla 4.1: Porcentaje de palabras con polaridad en los córpora de las tareas 10 y 11 utilizando diferentes lexicones.

tenida por cada palabra utilizando el número de significados posibles. Este lexicón nos proporciona tres valores: un valor que indica como de positiva es la palabra, otro indicando la negatividad y finalmente un valor que describe la objetividad de la palabra. Utilizamos los tres valores en nuestra aproximación.
$\sum_{w \in W} \frac{1}{|S|} \sum_{s \in S} SentiWordNet_{pos}(w, s)$, $\sum_{w \in W} \frac{1}{|S|} \sum_{s \in S} SentiWordNet_{neg}(w, s)$ y $\sum_{w \in W} \frac{1}{|S|} \sum_{s \in S} SentiWordNet_{obj}(w, s)$

Como haremos uso de diccionarios de polaridad para entrenar nuestro sistema, estudiamos el porcentaje de palabras de los vocabularios de las tareas que aparecen en los diccionarios considerados. En la tabla 4.1 vemos como la mayor parte de diccionarios de polaridad el vocabulario etiquetado que podemos encontrar no llega al 10 % con dos excepciones: SentiWordNet [5] y NRC [47], pero en el caso de SentiWordNet cada palabra puede tener más de un significado y por lo tanto debemos lidiar con la ambigüedad de las palabras.

Marcas sintácticas propias de Twitter. Para cada tweet contamos el número de *hashtags*, retweets, menciones y urls. Algunos *hashtags* como *#irony*, *#sarcasm*, ó *#not* son útiles para identiciar la presencia de lenguaje figurado en un tweet, de modo que también contamos el número de estos *hashtags* presentes.

Características estructurales o codificación. Se tuvo en cuenta la frecuencia de palabras completamente en mayúsculas por tweet y la frecuencia de palabras con dos o más caracteres repetidos. Trabajaremos en unicode lo que nos permite soportar emoticonos de manera transparente, como un carácter más.
Asimismo, hemos experimentado con otros conjunto de características como pueden ser: las etiquetas gramáticales, los n-gramas de palabras, etc. Sin embargo, los mejores resulta-

dos los obteníamos con sistemas entrenados utilizando combinaciones de las características aquí descritas.

4.4.2. Algoritmos de clasificación

Como describimos en el aparatado4.3 el objetivo de la tarea 10 era identificar la polaridad de un tweet entre tres posibles clases: positiva, negativa y neutral, por lo que abordamos la tarea como un problema de clasificación supervisada. Utilizamos una máquina de soporte vectorial con un *kernel* linear y realizamos el ajuste de parámetros utilizando el corpus *dev*.
Preliminarmente se experimentó con distintos clasificadores: Naïve Bayes, regresión logística, árboles de decisión, así como distintos *kernels* y algoritmos de optimización para la máquina de soporte vectorial; pero ninguno de estas aproximaciones mejoraron la descrita previamente. En la figura 4.7 vemos la exactitud de los 15 mejores sistemas. A partir de esta exploración inicial desarrollamos el sistema.

En el caso de la tarea 11, debemos identificar la polaridad de un tweet entre un rango de valores reales, por lo tanto los algoritmos de regresión son más adecuados. Empleamos una adaptación de las máquinas de soporte vectorial para regresión (SVR), también con un *kernel* linear, pero en este caso realizamos la selección de características mediante una validación cruzada con diez particiones aleatorias.
Para esta tarea, desarrollamos un sistema en dos pasos, el primero de los cuales predecía la polaridad, como acabamos de describir, mientras que el segundo paso consistía en revertir la polaridad predicha si se trataba de un tweet sarcástico. Identificamos un tweet sarcástico por la presencia de *hashtags* propios de esta categoría. Evaluamos distintas posibilidades: revertir completamente o parcialmente la polaridad con distintos pesos asociados a la presencia de distintas etiquetas. Sin embargo, ninguna de las configuraciones con las que experimentamos mejoraron el comportamiento de los sistemas de modo que se decidió descartar esta aproximación.
Hemos empleado el toolkit *scikit-learn*[58] a partir del cual hemos desarrollado un framework que nos permite definir modelos funcionales de clasificación. Estos modelos incluyen: funciones de preprocesado, extracción y vectorización de características de un tweet y la función de clasificación. Este framework puede recibir entre 1 y N modelos. Dado un tweet, se le asignará la polaridad más votada por los N clasificadores en el caso de la tarea

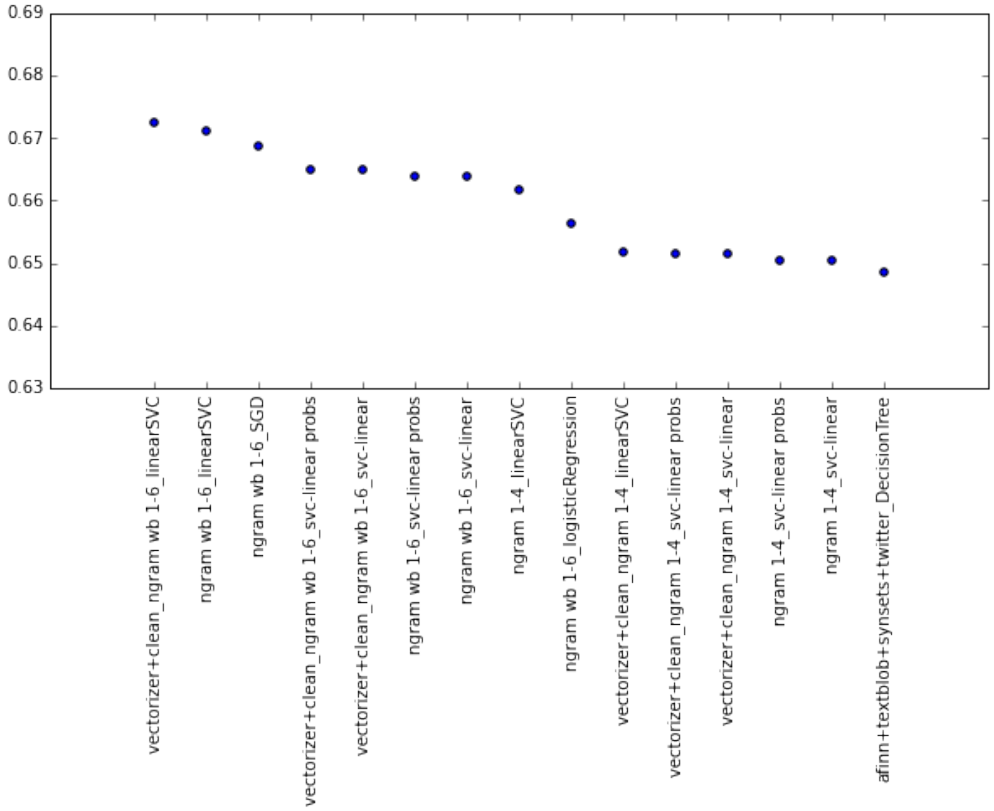


Figura 4.7: Resultados de exactitud obtenidos durante la experimentación inicial siguiendo distintas aproximaciones para vectorizar el texto como para entrenar el sistema de la tarea 10.

10 o la media aritmética de las predicciones en el caso de la tarea 11.
Esto nos permite entrenar y ajustar modelos seleccionando un conjunto de características extraídas del texto para después combinarlos, evitando los problemas de normalización entre características de distinta naturaleza, así como también nos permite preprocesar el corpus adoptando diferentes estrategias en función de las características que queramos extraer.

4.5. Entrenamiento

En esta sección expondremos la fase de experimentación que realizamos para seleccionar y ajustar los modelos propuestos a cada una de las tareas de SemEval.

4.5.1. Tarea 10

En primer lugar calculamos el *baseline* de la tarea. Si clasificáramos simplemente escogiendo la clase más probable obtendríamos un 26.49 % de valor-F1 y un 46.61 % de precisión. Partimos de estos resultados e intentaremos mejorarlos.
Tras estudiar los córpora y como anticipamos en el apartado anterior, nuestra aproximación consistió en entrenar distintos clasificadores con las características consideradas en el apartado 4.4.1.
Los modelos que se comportaron mejor estaban basados en clasificadores SVM con *kernel* linear. A continuación listamos las características de los cinco mejores modelos:

- **Modelo 1**: Un único clasificador basado en una SVM con *kernel* linear. Las características empleadas fueron:
  - 1-gramas a 6-gramas de caracteres del tweet.
  - 1-gramas a 6-gramas de caracteres del tweet etiquetado con el contexto negativo.
  - Los diccionarios de polaridad 1, 2, y 5.
  - Marcas sintácticas propias de Twitter.
  - Características estructurales.
- **Modelo 2**: Un único clasificador basado en una SVM con *kernel* linear. Las características empleadas fueron:
  - 1-gramas a 6-gramas de caracteres del tweet etiquetado con el contexto negativo.
  - Todos los lexicones descritos en el apartado 4.4.1.
  - Marcas sintácticas propias de Twitter.
  - Características estructurales
- **Modelo 3**: Un clasificador basado en una SVM con *kernel* linear entrenado con las siguientes características:
  - 1-gramas a 6-gramas de caracteres del tweet.
  - Los diccionarios de polaridad 1, 2, y 5.
  - Marcas sintácticas propias de Twitter.
  - Características estructurales.
- **Modelo 4**: En este caso entrenamos tres clasificadores basados en SVM con *kernel* linear. Cada clasificador fue entrenado con las siguientes características:
  - Todos los clasificadores utilizaron n-gramas desde 1-gramas a 6-gramas de caracteres del tweet.

- Cada clasificador añadía la información de un único lexicon. El primer clasificador fue entrenado con el lexicón 1, el segundo con el lexicón 2 y finalmente el último se entrenó con el lexicón 5.

La selección de la clase se realizó mediante una votación por mayoría de los tres clasificadores. Y en caso de empates se elegía la clase aleatoriamente.

- **Modelo 5**: También en este caso entrenamos tres clasificadores y la predicción se realiza mediante la votación de los tres clasificadores entrenados en este caso los modelos que componen el sistema son:
  1. Una máquina de soporte vectorial con *kernel* linear y entrenada utilizando las siguientes características:
     - 1-gramas a 6-gramas de caracteres del tweet.
     - 1-gramas a 6-gramas de caracteres del tweet etiquetado con el contexto negativo.
  2. Una máquina de soporte vectorial entrenada utilizando:
     - Los diccionarios de polaridad 1, 2, y 5.
     - Marcas sintácticas propias de Twitter.
     - Características estructurales.
  3. Una segunda máquina de soporte vectorial entrenada utilizando las siguientes características:
     - Los diccionarios de polaridad 3, y 4.
     - Marcas sintácticas propias de Twitter.
     - Características estructurales.

En la tabla 4.2 se pueden observar el comportamiento de nuestros mejores sistemas durante la fase de desarrollo.

	accuracy	*precision*	*recall*	F-1	$F-1_{neg}$	$F-1_{neu}$	$F-1_{pos}$
Modelo 1	0.6899	0.7035	0.6942	0.6826	0.5014	0.7303	0.6994
Modelo 2	0.7073	0.7201	0.7024	0.7013	0.5365	0.7407	0.7209
Modelo 3	0.6989	0.7146	0.7026	0.6901	0.4802	0.7391	0.7162
Modelo 4	0.6920	0.7074	0.6190	0.6816	0.4759	0.7391	0.7060
Modelo 5	0.6989	0.7146	0.7026	0.6542	0.3802	0.7221	0.6933

max

Tabla 4.2: Métricas obtenidas durante la fase de desarrollo de nuestros mejores sistemas en la tarea 10.

Finalmente enviamos el modelo 2, con el que mejores resultados obtuvimos durante la fase de desarrollo, para que fuera evaluado por los organizadores de la tarea 10 de SemEval.

4.5.2. Tarea 11

El desarrollo de la fase de experimentación en este caso fue análoga a la descrita en el apartado anterior. A partir de los mejores resultados obtenidos en la tarea 10 ajustamos nuestro sistema para resolver la tarea 11. Únicamente describimos el modelo que mejor se comportó durante la fase de experimentación, que consistió en:

- **Modelo 1**: Un único clasificador basado en una Support Vector Regression (SVR) entrenado con las siguientes características:
  - 3-gramas a 9-gramas de caracteres del tweet.
  - Los diccionarios de polaridad 1, 2, y 5.
  - Marcas sintácticas propias de Twitter incluyendo el número de *hashtags* utilizados para etiquetar lenguaje figurado (*irony*, *sarcasm* y *not*).
  - Características estructurales.

4.6. Evaluación en la tarea

Tras haber presentado los modelos que mejor se comportaron durante la fase de desarrollo del sistema, en esta sección mostraremos los resultados que obtuvimos en cada una de las dos tareas de SemEval 2015, tarea 10 y tarea 11, en las que participamos.

4.6.1. Tarea 10

La tarea de identificación del sentimiento presente en un tweet es una de las tareas que más interés despierta en la comunidad, prueba de ello es que en la edición de 2015 participaron 40 equipos. La mayor parte de ellos consiguieron superar el *baseline* de la tarea.
Los organizadores evaluaron los sistemas sobre un *test* extraído de Twitter (*Official Test*) y sobre cinco córpora que habían sido empleados en ediciones anteriores con el objetivo de medir el progreso de los sistemas presentados a la competición. Cabe destacar que el conjunto de tweets sarcásticos seleccionados para formar parte de la evaluación de la edición de 2015 fueron manualmente anotados, a diferencia de lo que ocurría las ediciones anteriores o en la tarea 11, en las que se confía en el etiquetado del propio autor del tweet para identificarlo como sarcástico.
Los algoritmos de aprendizaje más empleados por los participantes en la tarea han sido: SVM, máxima entropía, *Conditional Random Fields* y regresión linear. Entre las características más relevantes fueron las extraídas de los diccionarios de polaridad así como las

		F1	Rank	Mejor sistema	Peor sistema
Official Test	Twitter 2015	58.58	**24**	64.84	24.80
	Twitter Sarcasm 2015	43.91	34	65.77	22.25
Progress Test	LiveJournal 2014	68.33	28	75.34	34.06
	SMS 2013	60.20	28	68.49	26.14
	Twitter 2013	57.05	32	93.62	32.14
	Twitter 2014	61.17	**35**	74.42	32.2
	Twitter 2014 sarcasm	45.98	24	59.11	35.58

Tabla 4.3: Resultados de la evaluación oficial de la tarea 10 comparado contra el sistema que mejor y peor comportamiento presentó por corpus.

bolsas de palabras para representar los tweets.[9]

En la tabla 4.3 podemos ver los resultados de nuestro sistema comparado con los sistemas que participaron en la tarea. Comparamos el resultado del mejor y del peor sistema para cada corpus de evaluación. Con la aproximación aquí presentada, conseguimos la posición 24ª en el *test* oficial mientras que el sistema cae hasta la posición 34ª para el subconjunto de datos etiquetados como sarcásticos y hasta la posición 35ª para el caso del test de la edición de 2014.

Los sistemas que quedaron en las primeras posiciones emplearon aproximaciones similares a las aquí presentadas, lo que apunta a que el preprocesado y la representación de la información extraída de los diccionarios de polaridad tiene un fuerte impacto en el sistema final.

En general, el comportamiento de los sistemas presentados a la tarea se resienten ante los córpora en los que aparecen figuras retóricas como el sarcasmo. Lo que nos indica el impacto que estas tienen en el análisis de sentimientos.

4.6.2. Tarea 11

Esta tarea fue algo menos popular, si bien también imponía retos más complejos por la presencia de lenguaje figurado. Quince equipos participaron, muchos de ellos, como hicimos nosotros, se presentaron tanto a la tarea 10 como a la 11 por la fuerte relación existente entre las tareas.

Los organizadores de la tarea construyeron un corpus de evaluación a partir de conjuntos

[9] Consultar el artículo de Rosenthal et al. [69] con los detalles de la evaluación de la tarea.

de datos que recogían ejemplos de lenguaje figurado, como son: la ironía, el sarcasmo, la metáfora y un conjunto de control que no contenía lenguaje figurado.
Los organizadores desarrollaron tres sistemas para medir el *baseline* de la tarea y así poder comparar los resultados que los participantes en la tarea enviaron. Estos sistemas se basaban en una aproximación de bolsa de palabras y utilizaban: una SVM con la que obtuvieron un 0.390 de distancia coseno, un sistema basado en máxima entropía que consiguió un 0.426 de distancia coseno y finalmente un *decision tree* que obtuvo un 0.547 también en la distancia coseno. [10]
En la tabla 4.4 pueden verse los resultados oficiales de la evaluación de la tarea 11. Nuestro sistema consiguió quedar en quinto lugar en la evaluación general de la tarea, la que aglutina todos los subconjuntos. Sin embargo, contrariamente a lo que ocurría en la tarea 10, donde el comportamiento del sistema que presentamos empeoraba ante la presencia de lenguaje figurado, en esta tarea el comportamiento decae en el corpus sin lenguaje figurado hasta la octava posición. Esto puede explicarse debido a que nuestros sistemas están sobreadaptados al conjunto de entrenamiento y no han sido capaces de generalizar correctamente.
Cabe destacar que conseguimos quedar en primera posición en el corpus de sarcasmo.

	Distancia coseno	*Rank*	Mejor sistema	Peor sistema
Global	0.6579	**5**	0.758	0.059
Sarcasmo	**0.904**	**1**	0.904	0.412
Ironía	0.905	4	0.918	-0.209
Metáfora	0.411	5	0.655	-0.023
Otros	0.247	8	0.584	-0.025

Tabla 4.4: Resultados oficiales de la evaluación de la tarea 11 comparando nuestro sistema contra el mejor y el peor sistema presentado en cada categoría.

Además, los organizadores evaluaron el comportamiento de los sistemas empleando como métrica el error cuadrático medio (MSE), definido en el apartado 2.3.2. Los resultados se presentan en la tabla 4.5. Se puede comprobar la importancia de la métrica empleada para entrenar y ajustar los parámetros de los algoritmos de aprendizaje automático, puesto que ante las mismas características el comportamiento de nuestro sistema desciende, ya que nuestro sistema se ajustó mediante la minimización de la distancia coseno, que

[10] Para más información ver artículo de Ghosh et al. [21].

era la métrica oficial de la tarea.

	MSE	*Rank*	Mejor sistema	Peor sistema
Global	3.096	**8**	2.117	6.785
Sarcasmo	1.349	9	0.934	4.375
Ironía	1.034	8	0.671	7.609
Metáfora	4.565	4	3.155	9.219
Otros	5.235	5	3.411	12.16

Tabla 4.5: Evaluación de la tarea 11 empleando MSE.

4.7. Conclusiones y trabajo futuro

En esta sección se han presentado la participación en dos tareas de SemEval 2015, cuyo objetivo es proporcionar un conjunto de pruebas estándar para evaluar de forma equitativa y comparable las distintas aproximaciones propuestas para identificar el sentimiento presente en textos extraídos de redes sociales, prestando especial interés por Twitter.
Nuestra aproximación se basa en máquinas de soporte vectorial y explota distintos recursos como son los diccionarios de polaridad así como el propio texto de los tweets usando una aproximación basada en bolsas de n-gramas de caracteres y empleando los coeficientes tf-idf .
Esta aproximación consigue unos resultados competitivos, pero todavía nos queda mucho margen de mejora.
Es destacable como en ambas tareas, la presencia de lenguaje figurado marca el comportamiento de los distintos sistemas evaluados. Curiosamente, los cinco sistemas que, como nosotros, se han presentado a las dos tareas: CLaC-SentiPipe[55], UIR-PKU [28], UPF-taln [7], RGU [11] y SHELLFBK [20]; presentan un comportamiento similar al de nuestro sistema: en la tarea 10 se comportan mejor ante tweets literales, mientras que en la tarea 11 todos estos sistemas invierten su comportamiento presentando mejores resultados en el caso de los tweets que contienen lenguaje figurado.
Esto nos podría indicar que los modelos propuestos se adaptan a los datos de entrenamiento limitando la capacidad de los mismos para generalizar predicciones ante nuevos tweets no vistos. Por lo tanto, para abordar esta tarea en su totalidad, necesitamos definir

[11] Los desarrolladores de este sistema no presentaron el artículo que lo describe.

Tweet	Clase real	Clase predicha
@RushinConcert Driving from Nashville to ATL to see RUSH for the 11th time! Second row PITT baby!!! Yahoo!!	neutral	positva
Oomf today thought he can talk to me on some other shit wait til tomorrow lol	neutral	negativa
@MattPerekupka you're an angry elf tonight! Apples, Obamacare, relax its Friday...	neutral	negativa
The Celtics may have lost last night, but at least the Lakers did too!	neutral	negativa
Feeling sick happy I don't have to take SATs tomorrow.	positiva	negativa

Tabla 4.6: Ejemplo de tweets etiquetados erróneamente por nuestro sistema en la tarea 10.

sistemas capaces de modelar las distintas variantes del lenguaje figurado y su impacto sobre el sentimiento global presente en un tweet.
En la tabla 4.6 puede verse un extracto del estudio cualitativo que realizamos para analizar los errores que el sistema que propusimos realizaba sobre los tweets de la tarea 10. Nuestro sistema confundía fundamentalmente tweets de la clase negativa con tweets neutrales, como se podía prever analizando la probabilidad a priori de estas dos clases en los córpora. Además, en estos ejemplos podemos observar la dificultad de la tarea, para algunos de los tweets mal etiquetados por nuestro sistema automático también podrían llevar a confusión a un anotador humano.

Aunque queda fuera del ámbito de este trabajo, sería interesante estudiar, como trabajo futuro, el impacto de las características empleadas para entrenar los sistemas, para de este modo encontrar las claves para los comportamientos tan dispares de los sistemas a pesar de tratarse de sistemas muy similares.
Del mismo modo, sería interesante realizar un estudio de como afecta la presencia de lenguaje figurado en el análisis de sentimientos y continuar explorando el sistema propuesto en la tarea 11 que revirtiera la polaridad al encontrarse con lenguaje figurado.
Se han liberado nuevos recursos en este tiempo, algunos de ellos creados ad hoc por los participantes de SemEval, y sería interesante incluirlos en el sistema aquí propuesto para intentar mejorar el rendimiento del mismo.
Finalmente, nos plantemos ampliar el modelo aquí propuesto añadiendo otras características como pueden ser las etiquetas semánticas, que además también pueden emplearse

para desambiguar el significado de una palabra; o explorando bases de conocimiento como por ejemplo DBpedia[12].

[12] http://es.dbpedia.org/

5

Caracterización de perfiles de usuario

El último problema de NLP que se aborda en esta memoria es la caracterización de perfiles de usuario, *Author Profiling (AP)*, a partir de textos publicados en redes sociales en distintos idiomas. Se trata de una tarea relativamente reciente que ha despertado el interés de la comunidad científica por sus aplicaciones prácticas.

Este capítulo se organiza de acuerdo al siguiente esquema: comenzaremos introduciendo la tarea de la caracterización de perfiles de usuario; para continuar describiendo, en la sección 5.2, las aproximaciones que se han propuesto en la literatura; en la sección 5.3 describiremos los detalles de la tarea de AP propuesta en el marco del CLEF *Conference and Labs of the Evaluation Forum* en la edición de 2015 en la que participamos; el modelo propuesto se presentará en la sección 5.4 mientras que su entrenamiento y evaluación se describirá en las secciones 5.5 y 5.6 respectivamente; terminaremos el capítulo analizando los resultados y enumeraremos posibles líneas de trabajo para mejorar en la resolución de la tarea.

5.1. Introducción al problema

A lo largo de este trabajo se ha enfatizado la importancia de la información que proporcionan los usuarios en Internet. Sin embargo, carecemos de herramientas para extraer

información acerca de los propios usuario.

A diferencia de la identificación de usuario, la caracterización de perfiles no busca identificar al autor/a de un texto sino definir un modelo demográfico que le caracterice, por ejemplo, inferir la edad o el género de quien escribió un texto. La identificación de características demográficas del autor/a de un texto, es una de las tareas menos explotada de NLP, aunque sus aplicaciones despierten el interés en distintos ámbitos como pueden ser: la mercadotecnia [11], la lingüística forense [54], la sociología [1], etc.

En la literatura este problema se ha estudiado con textos normativos de longitud media o larga. Este tipo de textos presentan peculiaridades discursivas que hacen más probable encontrar características estadísticamente significativas para identificar rasgos acerca del autor/a de un texto.

En este trabajo nos centramos en la definición de perfiles de usuario de Twitter. Remitimos al lector/a a la justificación acerca de la importancia de esta red social que hicimos en capítulos anteriores[1].

Por las características de los tweets, textos cortos y en general agramaticales, la tarea de la caracterización de perfiles de usuario es más compleja.

Con el objetivo de crear un marco de trabajo común para grupos que trabajan en el problema de AP, desde el 2013 el PAN [2] propone una tarea en este área de investigación. En las ediciones anteriores, celebradas en 2013 y 2014, se recogieron textos de distintos sitios web (*blog*, Twitter, páginas de evaluación de hoteles, etc.). Sin embargo, en la edición de 2015 se acotó únicamente a textos extraídos de Twitter.

Entre las características demográficas que se deben identificar están: el género, la edad y en la edición de 2015 se incluyeron cinco rasgos de personalidad.

Se trata de una tarea multilingüe, ya que la organización proporciona corpora en varios idiomas. En las dos primeras ediciones se disponía de corpora en inglés y en español, mientras que en la tercera edición se añadieron corpora en italiano y en holandés. El resto de los detalles de la tarea se presentarán en la sección 5.3.

Propusimos un sistema que integraba cuatro modelos monolingües para cada uno de los idiomas que se consideraron en la tarea. Cada modelo se basaba en emplear algoritmos de aprendizaje automático entrenados utilizando n-gramas y lexicones para predecir las características demográficas del autor/a de los tweets. En la sección 5.4 describiremos las

[1] Ver las secciones: 3.1 y 4.1

[2] http://pan.webis.de/ Acceso: 12-02-2016

características de los modelos, así como detalles acerca de la adaptación de los recursos del idioma en el que fueron creados a otro.

5.2. Estado del arte

Como ya apuntábamos en la introducción, la identificación de los rasgos del autor/a de un texto ha atraído la atención de distintas disciplinas por las aplicaciones prácticas que tiene esta tarea.

Los primeros estudios de NLP que tratan la tarea de AP se centraron en la identificación del género y la edad.
Uno de los primeros trabajos desarrollados bajo estas premisas fue el de Pennebaker et al. [59]; en el se investigó la correlación entre ciertas características estilísticas y la edad o el género de su autor/a. Además, se vincula el lenguaje empleado con ciertas características psicológicas del mismo/a. Este estudio se realizó sobre texto normativo -al menos 250 caracteres- en inglés. Dicho trabajo incitó nuevas investigaciones en este área.[3, 31]
Con la eclosión de las redes sociales los trabajos de AP se centran en estos medios[52, 81, 51]. Estos sistemas utilizaron distintas características estilísticas como son: frecuencia de aparición de términos, POS, etc.; así como características de contenido como por ejemplo: n-gramas, conjuntos de palabras, diccionarios, etiquetas POS, etc; con las que se entrenaron una variedad de sistemas de aprendizaje automático como: máquinas de soporte vectorial (SVM), árboles de decisión, Naïve Bayes etc.
El trabajo de Argamon et al. [3] demostró una interdependencia entre las variables edad y género: la detección del género afectaba a la detección de la edad.

Recientemente, se ha ampliado el ámbito de estudio del AP tratando de identificar también las características psicológicas de los/as usuarios/as de redes sociales como Facebook y Twitter. En estos casos[15, 24, 35] se explotan, además de las características del texto, características propias de la red social -número de amigos, número de imágenes subidas, eventos a los que se ha asistido, etc.- para modelar la personalidad del autor/a.
Estos trabajos buscan identificar cinco rasgos de personalidad siguiendo la teoría de los Cinco Factores, *Big Five*[17], que más acuerdo despierta entre los expertos en psicología. Estos cinco rasgos son:

- Extroversión.

- Estabilidad emocional.
- Apertura a la experiencia.
- Afabilidad.
- Consciencia.

Vinciarelli and Mohammadi [76] elaboraron un resumen exhaustivo de los trabajos realizados sobre la identificación de rasgos de personalidad.

Con el objetivo de impulsar la investigación en AP desde 2013 [63, 64] se celebra un concurso de caracterización de perfiles de usuario dentro del PAN, como ya habíamos introducido. Los participantes han presentado sistemas basados en características estilísticas y de contenido, como los reportados en la literatura del área.
Cuando observamos la precisión que consiguieron los participantes de años anteriores advertimos la importancia de la naturaleza del corpus. Se consiguió una precisión combinada de género y edad en torno a un 40 % en el mejor de los casos en Twitter, lo cual podría indicar una identificación basada en el tema porque esta precisión desciende hasta un 25 % para el caso de textos extraídos de revisiones de servicios hoteleros donde el tema es fijo.
En la edición del 2015 del PAN [65], se ha ampliado la tarea; además de identificar el género y la edad debemos cuantificar los cinco rasgos de personalidad previamente descritos. En nuestro caso se propuso un sistema que utiliza técnicas de NLP y aprendizaje automático.

5.3. Descripción de la tarea

Como ya habíamos adelantado en el apartado anterior, en esta tarea abordaremos la caracterización de perfiles de usuario. Para la tarea propuesta en el PAN 2015 se consideraron los siguientes aspectos: género, edad y los cinco rasgos de personalidad (abierto, agradable, consciente, estable y extrovertido).

Una de las características de esta tarea es que se solicita a los participantes que envíen sus sistemas, no únicamente las predicciones. La organización creó un entorno virtual con las mismas características técnicas para todos participantes y ejecutó los sistemas. Por lo tanto, todos los modelos que desarrollaran debían ser competitivos en este entorno.

A diferencia de ediciones anteriores, los corpora que se proporcionaron a los participantes fueron extraídos de una única fuente: Twitter.

Se facilitaron a los participantes un conjunto de corpora multilingüe, compuesto por cuatro corpora en inglés, español, italiano y holandés, respectivamente. Estos corpora se dividieron en tres particiones: una para entrenamiento, otra para evaluación precoz (*early bird*) y una última para evaluar los sistemas entrenados. La distribución de tweets en el corpus de entrenamiento puede verse en la tabla 5.1, mientras que en la tabla 5.2 puede verse la distribución de los autores en el corpus de evaluación[3].

Corpus	# Tweets	# Autores
Inglés	14.166	152
Español	9.879	100
Italiano	3.687	38
Holandés	3.350	34

Tabla 5.1: Distribución del número de tweets y autores en el conjunto de entrenamiento.

Corpus	# Autores
Inglés	152
Español	100
Italiano	38
Holandés	34

Tabla 5.2: Distribución del número de autores en el conjunto de evaluación

Los tweets se distribuyen de modo balanceado por género y desbalanceado por edad, despuntando el rango de edad entre 25-34 que es la edad mayoritaria en Twitter, por lo tanto podemos asumir que la distribución de los tweets por autores es representativa de la realidad.
En la figura 5.2 podemos ver la distribución de las muestras del corpus por edad, en la figura 5.1 la distribución por genero y finalmente, en la figura 5.3 el rasgo de personalidad "afable", como muestra de uno de los rasgos de personalidad estudiados.

El etiquetado de la edad y el género se realizó según la información que los propios usuarios proporcionan en su perfil. Mientras que el etiquetado de los rasgos de personalidad se realizó de forma automática utilizando el test BFI-10 [62] disponible *on-line* y después se normalizó entre $[-0{,}5 \ldots 0{,}5]$.
Todos los corpus estaban etiquetados utilizando el género y los cinco rasgos de personalidad, pero únicamente los corpora en inglés y en español contenían información acerca

[3]Nuestro sistema no pudo ser evaluado en el *early bird*. Para más información al respecto consultar el resumen de la tarea [65].

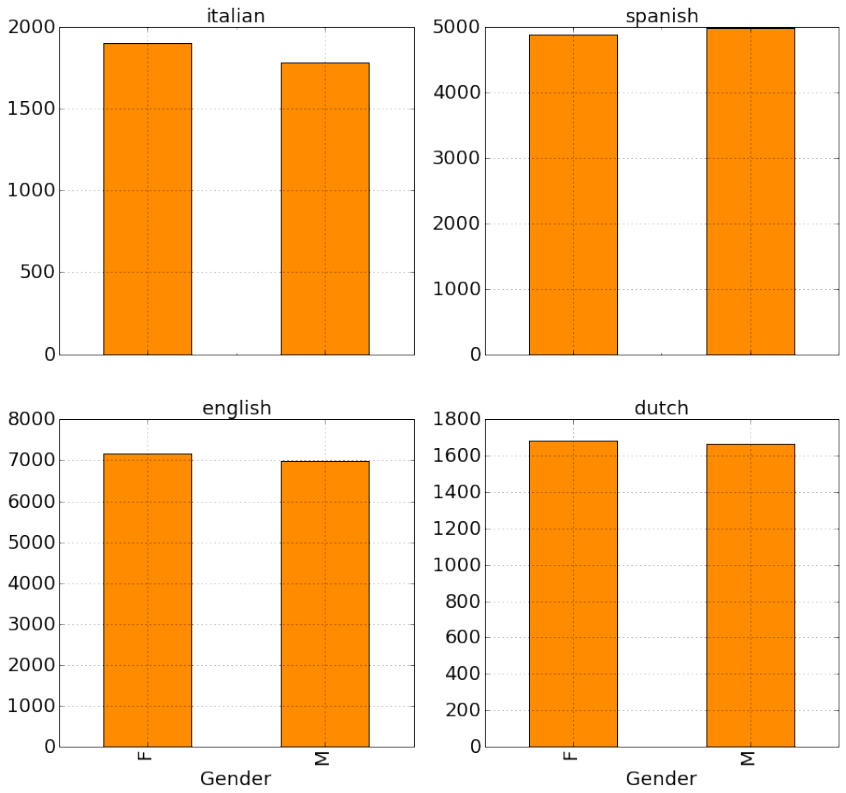


Figura 5.1: Distribución del género en el corpus de entrenamiento.

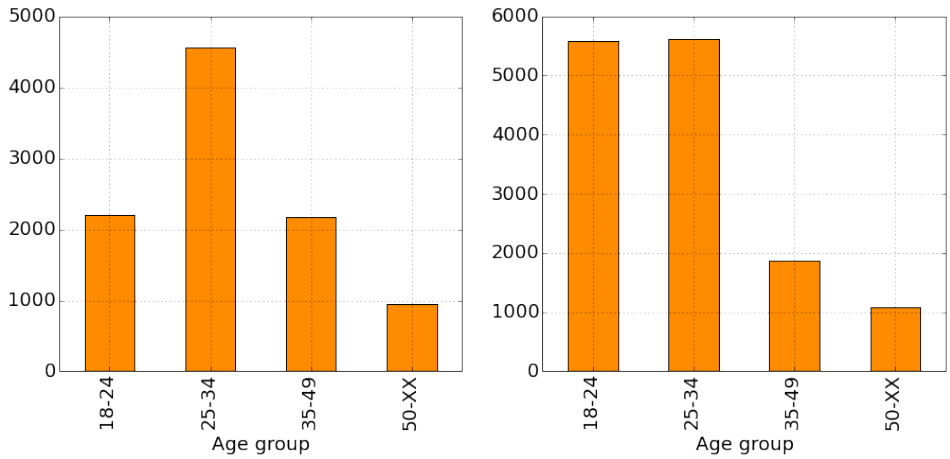


Figura 5.2: Distribución por edad en el corpus de entrenamiento.

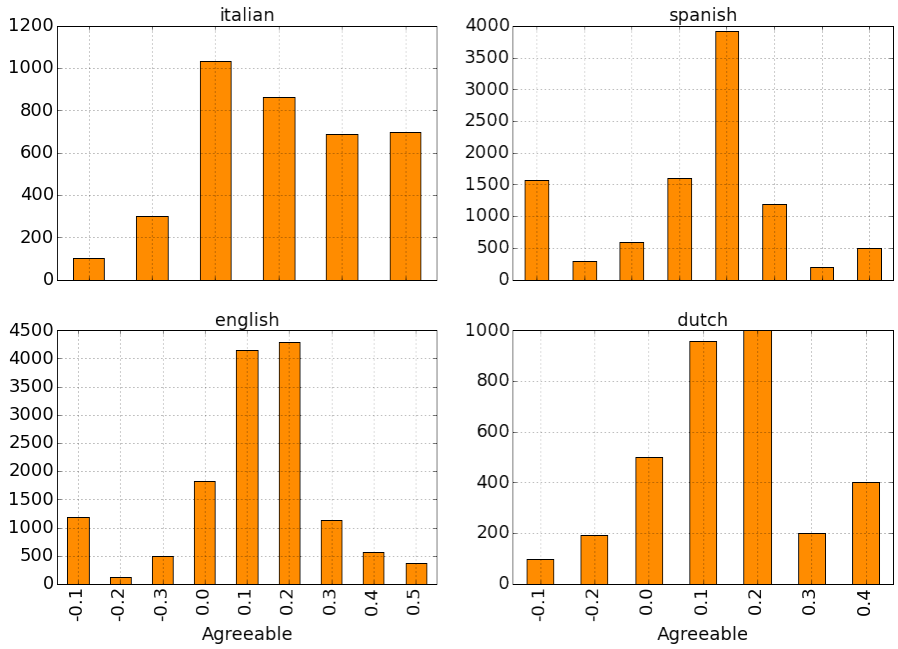


Figura 5.3: Distribución del rasgo de personalidad *afable* en el corpus de entrenamiento.

de la edad del autor/a.

Para el género se debían entrenar sistemas de clasificación binaria: femenino y masculino. Mientras que en el caso de la edad se definieron cuatro rangos de edad: 18-24,25-34,35-49, 50+.
Tanto para la edad como para el género se empleó la exactitud (*accuracy*).
En el caso de los rasgos de personalidad se debía cuantificar cada uno con un valor real. La métrica que se empleó fue la raíz cuadrada del error cuadrático medio, *Root Mean Square Error (RMSE)* definida según la ecuación 5.1.

$$RMSE = \sqrt[2]{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{5.1}$$

Antes de proceder a proponer un modelo para la tarea, se estudió el vocabulario de cada idioma. Para lo cual se eliminaron todos los signos de puntuación y las *stopwords* de los textos y se *tokenizaron* los textos remanentes para obtener el vocabulario. Se encontró

que las palabras más frecuentes eran invariantemente las relacionadas con la red social: RT, HTTP, username y vía, así como abreviaturas. Se estudió también la distribución de palabras más frecuentes por género y edad, pero no se encontraron diferencias acusadas en el vocabulario de cada grupo. En la figura 5.3 se puede ver la distribución de las palabras más frecuentes por rango de edad mientras que en la figura 5.4 se presenta esta distribución por género. Se realizó este estudio para todos los idiomas y rasgos de personalidad obteniendo resultados análogos en todas las categorías.

Español	18-24	username, HTTP, si, día, quiero, ser, 3, mejor, bien, vida, hoy, voy, ver.
	25-34	username, HTTP, q, si, vía, RT, d, Gracias, ser, ver, bien, día, va, hacer.
	35-49	username, HTTP, si, q, vía, RT, México, ser, hoy, Si, d, jajaja, Gracias, 1.
	50-XX	username, HTTP, q, RT, si, i, els, l, 2, 0, 1, Mas, d, amb, és, tasa, per.
Inglés	18-24	username, HTTP, m, like, know, love, want, get, RT, 3, one, people, time.
	25-34	HTTP, username, via, m, w, NowPlaying, like, others, 2, Photo, new, pic.
	35-49	HTTP, username, via, new, Data, RT, New, Big, Life, m, data, Facebook.
	50-XX	username, HTTP, RT, via, know, 2, like, m, good, day, love, 3, time, new.

Tabla 5.3: Distribución de las palabras más frecuentes del vocabulario por edad.

Español	Mujeres	username, HTTP, q, si, vía, ser, d, RT, vida, Gracias, ver, mejor, día
	Hombres	HTTP, si, RT, ser, ver, q, d, hoy, día, xD, 1 va, bien
Inglés	Mujeres	username, HTTP, via, m, like, love, know, RT, 3, get, want, one
	Hombres	username, HTTP, m, via, like, RT, 2, new, w, NowPlaying, know

Tabla 5.4: Distribución de las palabras más frecuentes del vocabulario por género.

Terminamos este estudio preliminar del corpus investigando los *hashtags*. El porcentaje de *hashtags* que aparecen en cada idioma es: un 37.9 % en inglés, un 26.7 % en español, un 59.9 % en italiano y un 27.3 % en holandés. Los *hashtags* nos proporcionan información del contenido de los tweets. Es el propio usuario quien etiqueta los tweets y en general, estos *hashtags* contienen el significado global del mismo, por lo tanto, intuitivamente la información que contienen puede ser relevante para clasificar el perfil del usuario.

En la figura 5.4 encontramos la distribución de *hashtags* en español. Observamos una contaminación de vocabulario en inglés en el corpus en español. Este mismo fenómeno

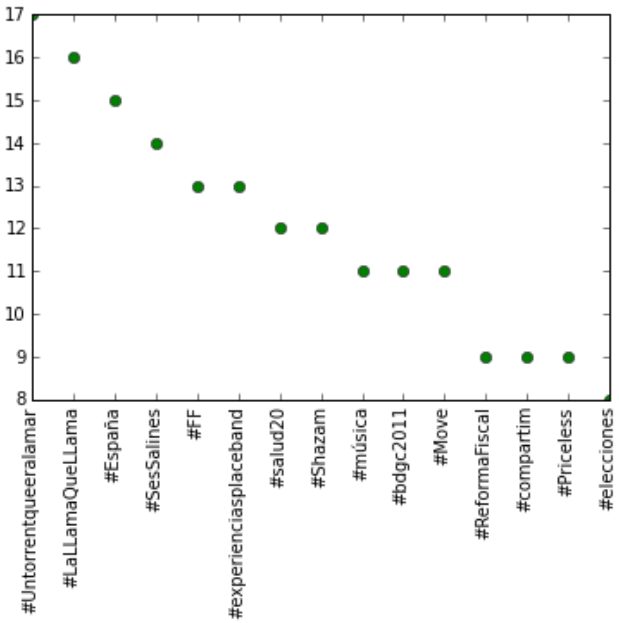


Figura 5.4: Distribución de los quince hashtags más frecuentes en castellano.

también se observó en el corpus en italiano y holandés. La contaminación entre idiomas dificultará el uso eficaz de diccionarios de polaridad, puesto que son recursos dependientes de la lengua utilizada.

5.4. Modelo propuesto

Tras el estudio del vocabulario y el análisis de la tarea, se propuso un modelo basado en características estilísticas y de contenido similar al propuesto en la sección 4.4, ampliando el sistema para el tratamiento multilingüe de los tweets. Asimismo, se seleccionaron los métodos de aprendizaje automático más adecuados para inferir los distintos aspectos del perfil de usuario del autor/a de un tweet.

De nuevo, se empleó el *toolkit scikit-learn* [58] para el desarrollo del sistema. Se desarrolló un ampliación del *toolkit* que permitiera entrenar utilizando los lexicones y las características estilísticas de forma nativa. Como únicamente se disponía de lexicones en inglés, se tradujeron de modo automático al español, italiano y holandés. Para ello se utilizó la

API de *Google Translate*[4]. En caso de que una palabra tuviera más de una acepción, se seleccionó la primera palabra que devuelve el traductor que corresponde con la acepción más usual. A la palabra traducida se le asignó la polaridad que tenía la palabra en el idioma original del lexicón.

Se decidió entrenar siete clasificadores distintos, uno para identificar género, otro para la edad y cinco más por cada rasgo de personalidad. En una fase preliminar se experimentó combinando clasificador de genero con el de edad, clasificando entre ocho clases diferentes, pero los resultados no fueron satisfactorios por lo que se descartó esta aproximación. Entrenar un clasificador por cada característica del autor nos permitió seleccionar un clasificador que se adaptara mejor a la tarea concreta.

En líneas generales, se exploró el comportamiento de distintos clasificadores extrayendo un conjunto de características del corpus de entrenamiento.
Tal y como se recoge en la literatura de AP, se prescindió de la fase de preproceso. Esta etapa puede eliminar característica estilísticas del autor/a del texto y por lo tanto dificultar la resolución de la tarea.

Las características que se extrajeron de los corpora y que se emplearon para entrenar fueron:

- Características del texto: en este grupo encontramos los n-gramas extraídos del texto. Como únicamente nos interesa en esta categoría la frecuencia de aparición de los n-gramas, se convirtió todo el texto a minúsculas para extraer estas características. Se experimentó con los siguientes conjuntos de características:
  - Los coeficientes tf-idf de n-gramas de palabras. Se consideraron los siguientes rangos de n-gramas: 1-3, 1-4, 1-6, 1-9, 3-6 y 3-9 en todos los casos que listamos a continuación.
  - Los coeficientes tf-idf de n-gramas de caracteres inter-palabra.
  - Los coeficientes tf-idf de n-gramas de caracteres intra-palabra.
  - Bolsa de palabras.
- Características de estilo:
  - Número de palabras con caracteres repetidos.
  - Número de palabras con todos sus caracteres en mayúsculas.

[4] https://cloud.google.com/translate/docs

- Sintaxis de Twitter: número de retweets, número de hashtags, número de menciones y el número de URLs.

- Diccionarios de polaridad: para este tipo de característica se eliminaron las *stopwords* y se convirtieron todas las palabras a minúsculas. Para cada tweet y para cada diccionario de polaridad considerado se obtuvo una puntuación, la cual se calculó sumando la polaridad de cada palabra y normalizando por la longitud del tweet, empleando la siguiente fórmula: $\frac{1}{|W|}\sum_{w \in W} lexicon(w)$.
  1. Afinn-111 [29].
  2. NRC [48]
  3. NRC hashtags [48].
  4. Jeffrey [33].

  Tanto el cálculo del *score* como los diccionarios de polaridad fueron definidos previamente en la sección 4.4.1.

Los clasificadores que se consideraron para resolver la tarea fueron:

- Máquinas de soporte vectorial lineal. (Las dos implementaciones que podemos encontrar en el toolkit)[5].
- Máquinas de soporte vectorial con kernel polinómico.
- Naïve Bayes.
- Regresión logística.
- *Random forests*.

5.5. Entrenamiento

Nos planteamos dos opciones para experimentar con los corpora de entrenamiento: unir todos los tweets de un autor para tener una única muestra de aprendizaje por autor de longitud similar a la de los textos normativos; o bien, que cada uno de los tweets se empleara como una muestra de aprendizaje independiente. Ambas opciones generan un vector del tamaño del número de características, pero en el primer caso este vector será menos disperso y con menor número de muestras (una por autor) lo cual puede se útil al aplicar algoritmos como Naïve Bayes, porque simplifica la distribución de probabilidad. Sin embargo, la segunda opción nos generará más muestras (una por tweet del conjunto

[5] SVC http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html y LinearSVC http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

de datos) y el vector que se genere será más disperso lo que favorece el entrenamiento de la máquina de soporte vectorial.

Preliminarmente, experimentamos con todos los clasificadores y todas las combinaciones de características descritas en el apartado 5.4.
Se segmentó el corpus en diez grupos aleatorios, entrenando con nueve de ellos y evaluando con el que no se vio durante el entrenamiento, siguiendo una aproximación clásica de validación cruzada con diez particiones (*10 fold cross validation*). Se ajustaron los modelos utilizando como métrica la exactitud, definida en el apartado 2.3.1.
En las figuras 5.5, 5.6 puede verse la precisión que obtuvimos con los mejores modelos entrenados para italiano y holandés respectivamente, utilizando los lexicones en inlglés así como las versiones traducidas de los mismos que se desarrollaron automáticamente para esta tarea.

Sin embargo, estas gráficas son engañosas, porque aunque la exactitud media presenta buenos resultados, al observar la desviación típica de las particiones realizadas durante el entrenamiento, vemos que estamos ante modelos muy variables. Como muestra incluimos en las gráficas 5.7 y 5.8 los diagramas de cajas (*box & whiskers*) de los modelos entrenados con el conjunto de datos en italiano para identificar el género y el aspecto de personalidad "abierto". En estas figuras se puede ver como varía el comportamiento del sistema en función del conjunto de datos de la partición empleada para entrenar.

Tras esta fase de experimentación, se optó por entrenar máquinas de soporte vectorial para identificar el género y la edad del autor/a del texto, y se emplearon clasificadores basados en regresión logística para cuantificar los cinco rasgos de personalidad.
Tras seleccionar los mejores modelos se repitió la experimentación únicamente para ajustar los parámetros siguiendo de nuevo una aproximación basada en validación cruzada.
Finalmente, las características que se tuvieron en cuenta fueron: los coeficientes tf-idf de n-gramas de caracteres (en un rango entre 1-3 y 1-6 dependiendo del modelo), todos los diccionarios de polaridad traducidos (excepto en el caso del inglés, que no se necesitó traducir el recurso) y todas las características estilísticas.

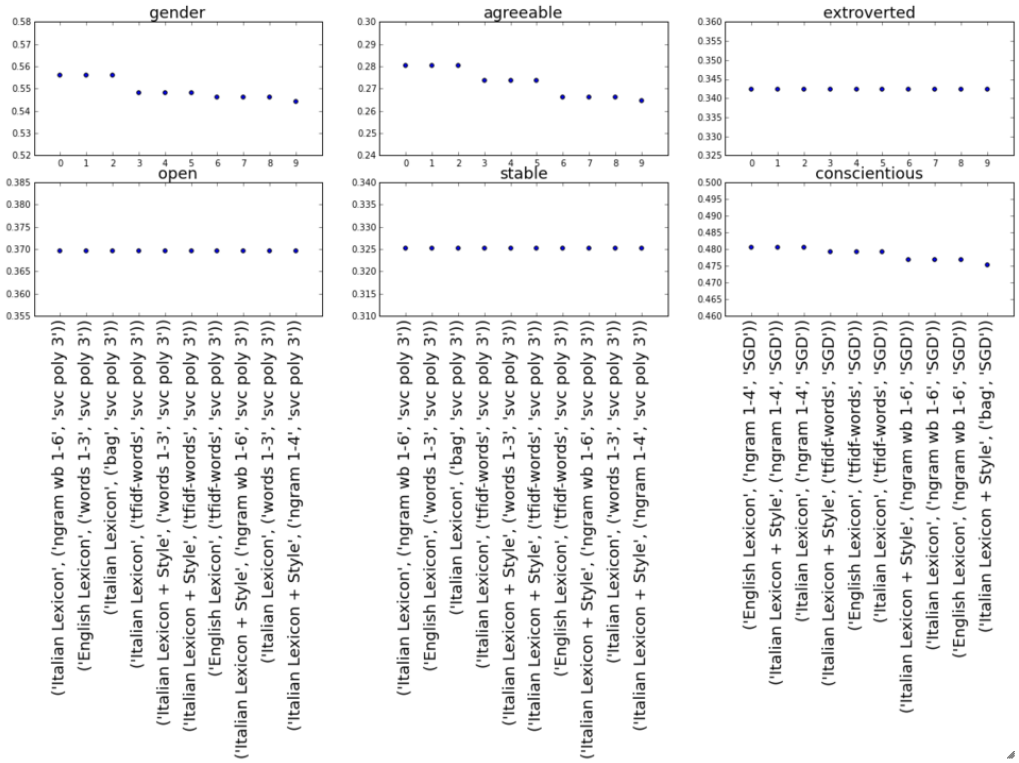


Figura 5.5: Mejores modelos obtenidos durante la fase de entrenamiento para el corpus en italiano. La etiqueta de cada clase define los tres componentes del modelo: si se ha empleado un lexicón en inglés o traducido, el tipo de vectorización del texto y el algoritmo de aprendizaje empleado.

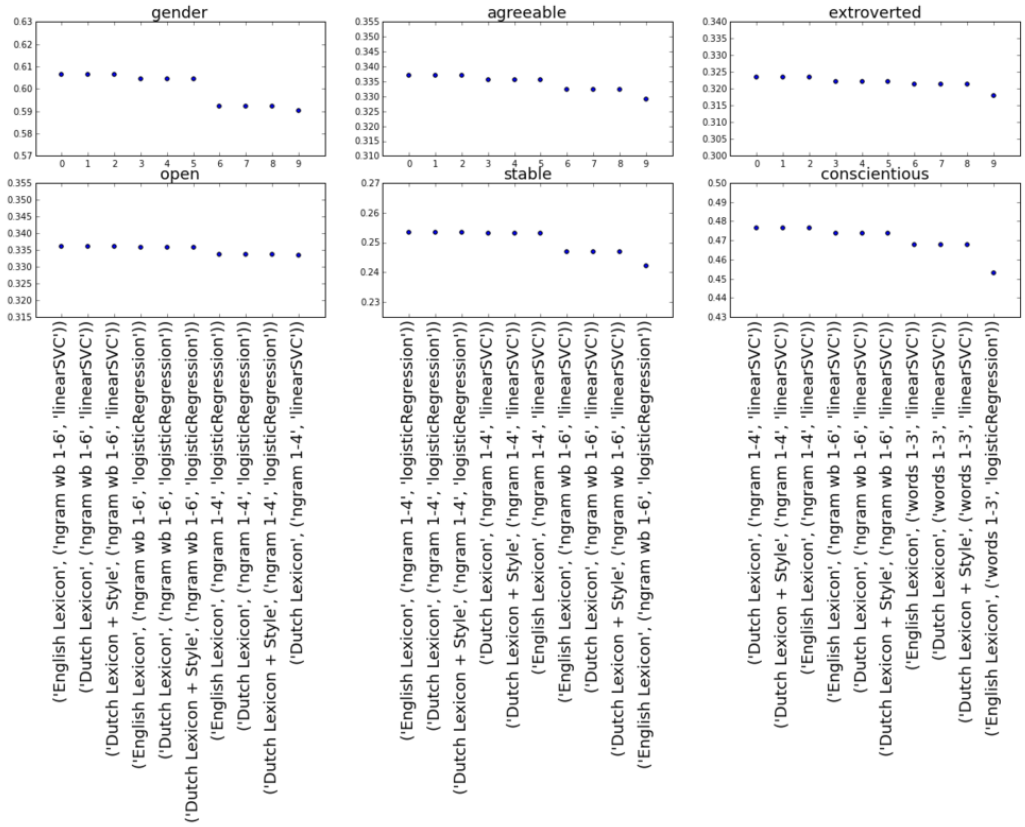


Figura 5.6: Mejores modelos obtenidos durante la fase de entrenamiento para el corpus en holandés. Cada sistema está etiquetado análogamente a la gráfica 5.5.

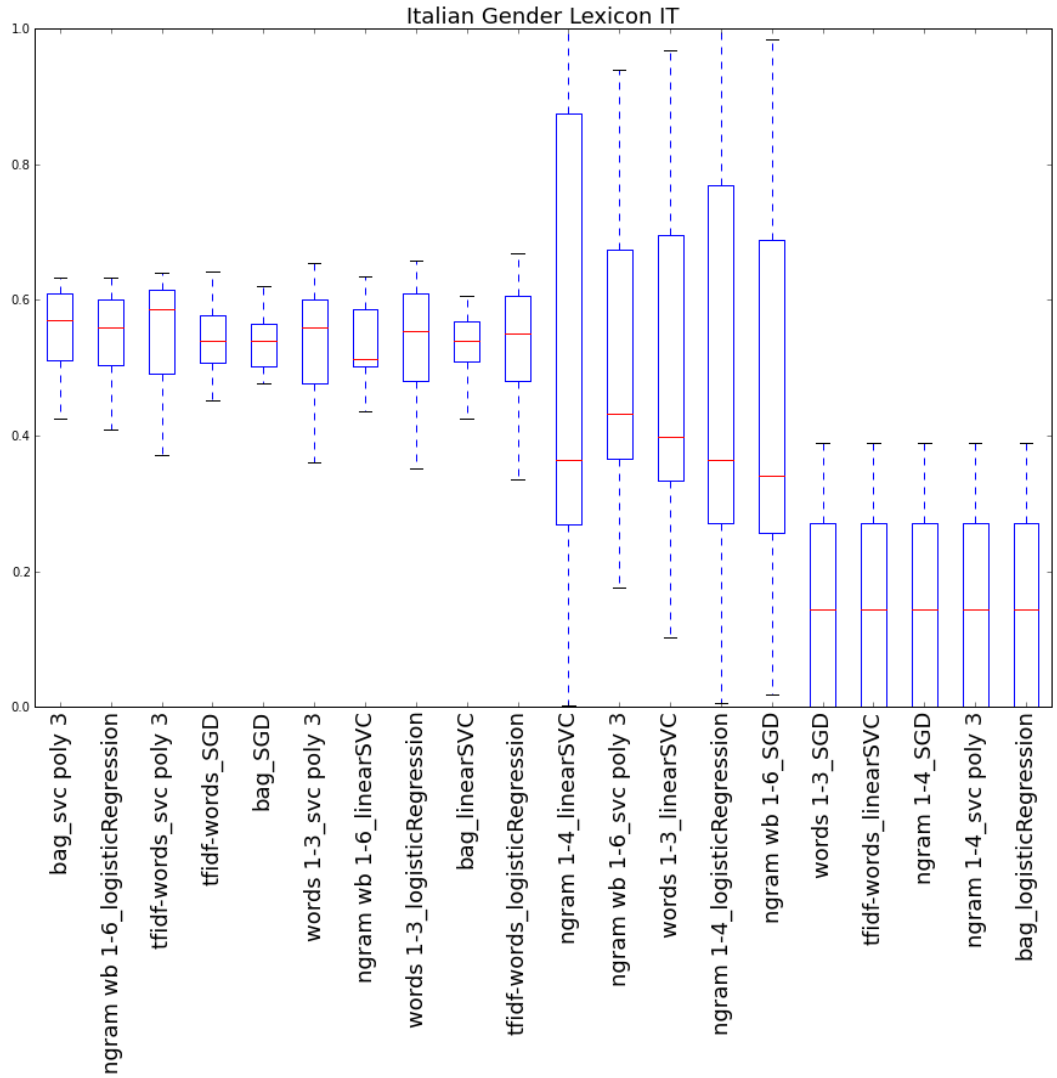


Figura 5.7: Diagrama de cuartiles para el género en Italiano.

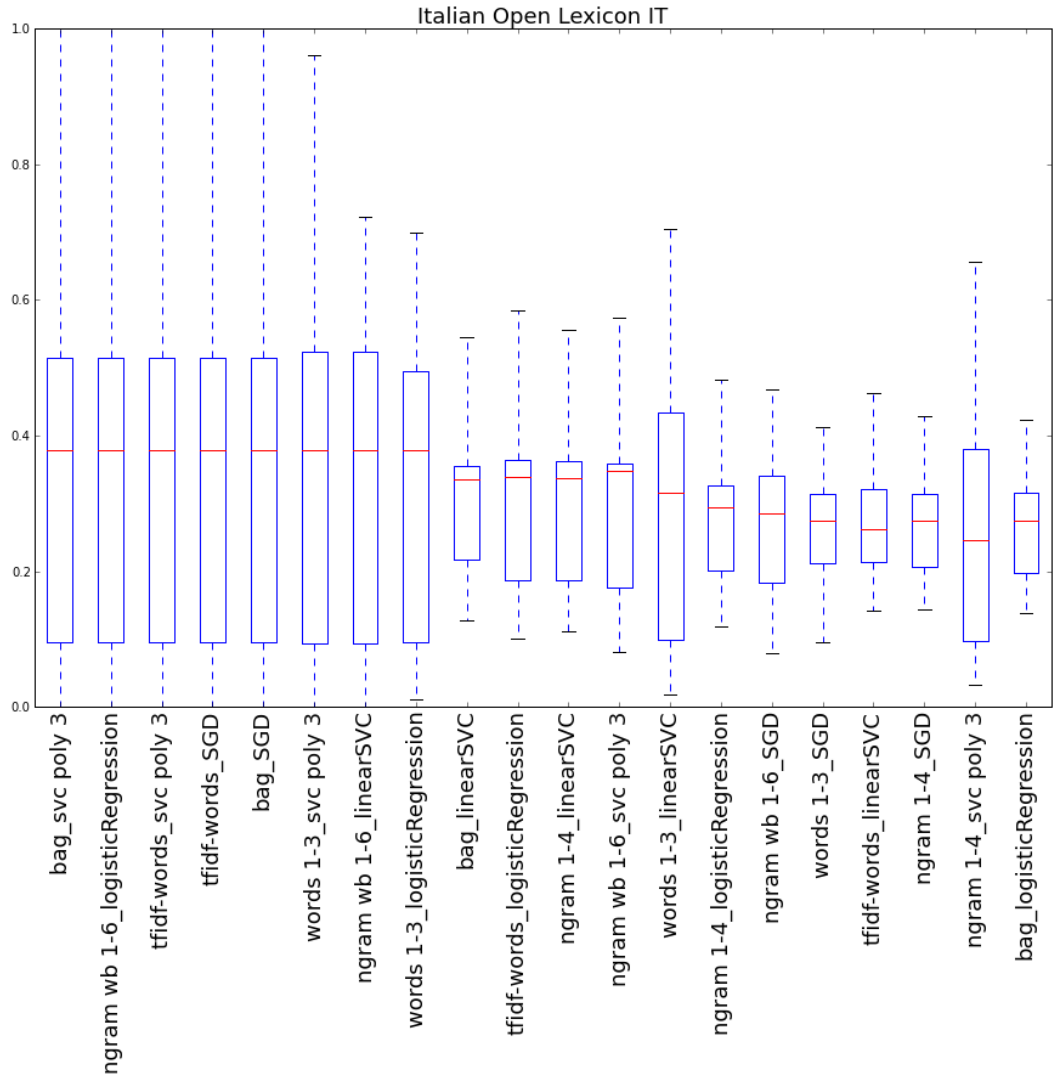


Figura 5.8: Diagrama de cuartiles para el rasgo de personalidad ``abierto'' en Italiano.

En la tabla 5.5 se muestra la exactitud media en el caso del género y la edad, y RMSE para el caso de los rasgos de personalidad de las 10 particiones de nuestros mejores sistemas durante la fase de entrenamiento:

		Exactitud/RMSE
Español	Género	56.9 % †
	Edad	46.58 % †
	Abierto/a	39.55 % *
	Afable	40.44 % *
	Consciente	32.84 % *
	Estable	29.05 % *
	Extrovertido/a	36.98 % *
Inglés	Género	53.49 % †
	Edad	55.29 % †
	Abierto/a	24.78 % *
	Afable	23.7 % *
	Consciente	20.8 % *
	Estable	17.81 % *
	Extrovertido/a	20.85 % *
Holandés	Género	57.49 % †
	Abierto/a	43.69 % *
	Afable	42.33 % *
	Consciente	49.82 % *
	Estable	38 % *
	Extrovertido/a	46.37 % *
Italiano	Género	61.63 % †
	Abierto/a	42.20 % *
	Afable	43.28 % *
	Consciente	52.67 % *
	Estable	46.15 % *
	Extrovertido/a	45.65 % *

† Rasgos evaluados mediante la exactitud (*accuracy*).

* Rasgos evaluados mediante la RMSE.

Tabla 5.5: Exactitud media obtenida mediante validación cruzada durante la fase de entrenamiento del PAN.

Tras terminar esta fase de entrenamiento, nuestra labor consistió en poner en producción en la máquina del concurso nuestros mejores sistemas.
El concurso requería disponer de dos *scripts* que implementaran dos etapas: una de entrenamiento y otra de test para cada característica de personalidad e idioma, de modo que los organizadores pudieran evaluar los sistemas.
Por lo tanto, se desarrolló el software necesario para que dado un corpus de entrada, entrenara y serializara los modelos; análogamente, se desarrolló un software que ante un corpus de evaluación y la ruta donde se encuentran los modelos serializados los cargara en memoria y realizara las predicciones correspondientes. Fuimos especialmente cuidadosos en la producción de este software para que fuera lo más eficiente posible.

5.6. Evaluación

A continuación se presentan los resultados que obtuvimos en la evaluación oficial del PAN 2015.
Un total de 22 sistemas participaron en la evaluación de la tarea. La mayor parte de ellos realizaron algún tipo de preprocesado de los tweets como por ejemplo: eliminar URLs, sustituir *hashtags* o URLs por tokens, pasar a minúsculas todo el texto, etc. Las característica empleadas para entrenar los sistemas fueron muy similares a los de años anteriores, incluyendo combinaciones de características de estilo con características de contenido como las ya descritas previamente. En cuanto a los algoritmos de aprendizaje automático, se utilizaron fundamentalmente SVM, aunque también se emplearon: árboles de decisión implementados siguiendo el algoritmo J48, análisis lineal discriminante, etc. [6]

[6] Consultar el artículo de Rangel et al. [65] para ver en detalle las diferentes aproximaciones que se emplearon en la tarea.

		Exactitud/RMSE
Español	Género	62.5 % †
	Edad	56.82 % †
	Abierto/a	16.17 % *
	Afable	17.29 % *
	Consciente	18.53 % *
	Estable	24.40 % *
	Extrovertido/a	20.97 % *
Inglés	Género	63.38 % †
	Edad	59.86 % †
	Abierto/a	20.23 % *
	Afable	17.54 % *
	Consciente	18.19 % *
	Estable	27.81 % *
	Extrovertido/a	17.70 % *
Holandés	Género	71.88 % †
	Abierto/a	13.23 % *
	Afable	17.05 % *
	Consciente	13.92 % *
	Estable	17.85 % *
	Extrovertido/a	18.29 % *
Italiano	Género	69.44 % †
	Abierto/a	20.21 % *
	Afable	16.24 % *
	Consciente	12.47 % *
	Estable	25.33 % *
	Extrovertido/a	13.94 % *

† Rasgos evaluados mediante la exactitud (*accuracy*).

* Rasgos evaluados mediante la RMSE.

Tabla 5.6: Precisión media obtenida en la evaluación oficial del PAN.

En la tabla 5.6 se muestran los resultados que nuestro sistema obtuvo en la evaluación de la tarea. Como se ha apuntado previamente la métrica empleada para el género y la edad

fue la precisión, mientras que se empleó el RMSE para evaluar los rasgos de personalidad.

El sistema que se presentó obtuvo una precisión global de 68.57 % quedando en 13ª posición de 19 sistemas que se presentaron. La precisión de los sistemas osciló entre el 50.49 % y el 82.15 % y los sistemas se concentraban por debajo de la media (65.47 %). Los resultados más dispersos se pueden encontrar para el español. Mientras que los mejores resultados fueron los obtenidos por los sistemas para italiano y el holandés a pesar de que se trate de los corpora de los que disponíamos de menos datos.
Respecto a los resultados de ediciones anteriores, los resultados mejoraron a pesar de que en esta edición se utilizaron datos de Twitter, por lo que los organizadores afirman que a pesar de la longitud de los tweets, si el número de tweets es suficientemente grande, se pueden entrenar sistemas automáticos para caracterizar el perfil de un usuario competentemente. Si se desea ampliar la información, el trabajo de Rangel et al. [65] recoge un análisis detallado de los resultados de los participantes.

5.7. Conclusiones y trabajo futuro

En este capítulo se ha presentado la participación al PAN, cuyo objetivo es identificar características demográficas acerca del autor/a de un texto. La edición de 2015 propuso la identificación del género, la edad y cinco rasgos de personalidad para datos extraídos de Twitter en cuatro idiomas: español, inglés, italiano y holandés.

Presentamos una aproximación para resolver el problema basada en aprendizaje automático, empleando como características para entrenar el sistema los coeficientes tf-idf de n-gramas de caracteres, diccionarios de polaridad -traducidos automáticamente para aquellos idiomas en los que no existe el recurso- y características estilísticas. Se empleó una SVM para clasificar la edad y el género, y un algoritmo de regresión logística para cuantificar los rasgos de personalidad.
Nuestro sistema presentó un comportamiento aceptable en todas las categorías. La aproximación aquí presentada de traducción automática de los recursos, mantiene los resultados del sistema independientemente del idioma, a pesar de que la traducción pueda tener errores y que la polaridad de una palabra no tiene porque ser la misma entre diferentes idiomas. Sin embargo, esta aproximación nos ha permitido generalizar los sistemas para todos los idiomas considerados en la tarea.

Mejorar la aproximación empleada para la traducción de los recursos podría suponer mejorar el comportamiento del sistema multilingüe aquí presentado. Aunque lo ideal sería el desarrollo de sistemas adaptados por idioma, lo cual capturaría la variabilidad del vocabulario en el idioma del texto.
Además, sería interesante ampliar el número de características estilísticas consideradas en el entrenamiento de los sistemas, así como tratar el argot, muy presente en redes sociales e incluir nuevos recursos.

Uno de los problemas que queda fuera del ámbito de esta tarea, pero que debería considerarse para poder ejecutar un sistema automático de AP en condiciones de trabajo reales, es la gestión del gran volumen de datos que producen los usuarios de Twitter en tiempo real., así como el desarrollo de un sistema capaz de aprender de los nuevos datos aplicando técnicas de *on-line learning*.

Finalmente, sería interesante aplicar el sistema propuesto a un volumen de datos mayor, donde aparecieran muchos más autores y donde las características demográficas no se obtuvieran utilizando los datos de su perfil ya que los usuarios pueden mentir; como tampoco automáticamente utilizando una herramienta *on-line*. Así, haciendo uso de datos etiquetados por expertos, por ejemplo, sería interesante comprobar si los resultados obtenidos por los sistemas aquí presentados se mantienen estables.

6
Conclusiones

En esta memoria del trabajo final del "Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital" se han presentado diversas tareas en el área de investigación del Procesamiento del Lenguaje Natural como son: la identificación de idioma, el análisis de sentimientos y la caracterización de perfiles de usuario.
Todas estas tareas tienen en común que buscan extraer información relevante a partir de textos de redes sociales. El crecimiento de estos medios apunta que estas tareas cada vez serán más relevantes y por lo tanto, el desarrollo de sistemas automáticos capaces de extraer conocimiento en tiempo real será más empleado en distintas aplicaciones fuera del ámbito académico.

Utilizando los conocimientos adquiridos a lo largo del máster hemos sido capaces de abordar estas tareas y presentar sistemas competitivos en tres concursos internacionales. Los sistemas aquí presentados tienen en común el uso de algoritmos de aprendizaje automático entrenados a partir de textos extraídos de redes sociales. Se optó por máquinas de soporte vectorial en el caso de tareas en las que se debiera identificar un número finito de clases y algoritmos de regresión para aquellas tareas donde se debía proponer un valor entre un rango de valores continuo.

Los sistemas propuestos se entrenaron empleando características superficiales, como las bolsas de n-gramas, que no son capaces de capturar el contexto más allá de n palabras

en las sentencias. Por lo tanto, como trabajo futuro se debería experimentar con métodos de representación de texto continuos (*word embeddings*) como el propuesto por Bengio et al. [10] y que ha impulsado nuevas aproximaciones a los sistemas propuestos para resolver problemas de NLP.

Además, la mayor parte de recursos de los que disponemos para capturar conocimiento semántico, como pueden ser: los diccionarios de polaridad, los diccionarios de emociones, las bases de conocimiento, etc, se han generado para textos en inglés, por lo tanto, será necesario adaptar o desarrollar nuevos recursos para poder resolver las tareas de NLP para todos, o al menos, una mayoría de los idiomas empleados en Internet.

Análogamente, la mayor parte de los recursos se desarrollan con texto normativo. Estos recursos no llegan a recoger el uso del lenguaje que hacen los usuarios en redes sociales, lo que queda demostrado por la baja cobertura que presentan. Desarrollar recursos que recojan la polaridad del argot empleado en redes sociales podría ayudar a mejorar el comportamiento de los sistemas aquí presentados.

Durante el desarrollo de este trabajo se han liberado nuevos recursos y sería interesante incluirlos en nuestros sistemas y evaluar el comportamiento de los mismos empleando estos nuevos recursos.
En esta misma línea se debería estudiar el impacto de las características en los modelos presentados, ¿Son todas las características igual de significativas?, ¿Son redundantes?, ¿La interacción entre las características provoca errores en los algoritmos de aprendizaje?, etc.

En este trabajo se han presentado tres tareas independientes, pero en un caso real se debería desarrollar un sistema que integrara los tres sistemas aquí presentados, tras identificar el idioma, podríamos realizar un análisis de sentimientos y crear un perfil del usuario que emitió la opinión.
Los módulos de análisis de sentimientos y el de creación de perfil de usuario deberían poder comunicarse para reforzar lo aprendido en cada uno de ellos, puesto que parece sensato trabajar con la hipótesis de que las características demográficas puedan afectar a la expresividad de sentimientos, al evaluar productos, servicios, etc. Por lo tanto, la detección de rasgos demográficos puede ser relevantes para el correcto análisis de sentimientos.

Como se ha apuntado previamente, los sistemas aquí presentados se han desarrollado utilizando un conjunto de datos limitado que proporcionaron los organizadores de las

respectivas tareas. Sin embargo, para conseguir sistemas útiles en condiciones de trabajo reales, será necesario adaptar los sistemas aquí presentados para que puedan gestionar datos en tiempo real.

Todos nuestros sistemas se basaban en aproximaciones supervisadas. Pero en las redes sociales podemos encontrar una cantidad ingente de datos no etiquetados, y el porcentaje de datos que etiquetemos, con mucho esfuerzo económico y temporal, será insignificante en relación con los datos no etiquetados.
En trabajos futuros se podría estudiar las siguientes dos técnicas, entre otras, para mejorar los sistemas aquí desarrollados: en primer lugar empleando algoritmos de agrupamiento (*clustering*) que permitan clasificar datos no etiquetados a partir de un pequeño conjunto de datos etiquetados y agrupando el texto de los datos no etiquetados en función de la similitud de las palabras empleadas en cada clase por ejemplo, y en segundo lugar mediante el uso de técnicas para entrenar algoritmos de aprendizaje automático que permitan incorporar nuevo conocimiento como puede ser técnicas de *on-line learning*. El objetivo debería ser desarrollar sistemas semi-supervisados, que a partir del conocimiento etiquetado mejore el comportamiento de los sistemas propuestos a partir de datos no etiquetados.

A

Publicaciones

En este apartado se incluyen las publicaciones que se han realizado como resultado del trabajo realizado. Además de las publicaciones citadas préviamente:

- Sanchis, E., Giménez, M., Hurtado, L.-F. (2014). Language identification with limited resources. V Jornadas TIMM, (pp. 7–10)[71]
- Hurtado, L.-F., Pla, F., Giménez, M., Sanchis, E. (2014). Elirf-upv tweetlid: Identificación del idioma en twitter. En *TweetLID@ SEPLN*.[34]
- Giménez, M., Pla, F., Hurtado, L.-F. (2015). Elirf: A svm approach for sa tasks in twitter at semeval-2015. En *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*(pp. 574–581). Denver, Colorado: Association for Computational Linguistics. [23]
- Giménez, M., Hernández, D. I., Pla, F. (2015). Segmenting target audiences: Automatic author profiling using tweets. En *Procedings of the CLEF Conference and Labs of the Evaluation Forum* [22]

Bibliografía

[1] C. Alarcón-del Amo, M-C.and Lorenzo-Romero and M-Á. Gómez-Borja. Classifying and profiling social networking site users: A latent segmentation approach. *Cyberpsychology, behavior, and social networking*, 14(9):547–553, 2011.

[2] P. Analytics. Twitter study–august 2009. 2009.

[3] S. Argamon, M. Koppel, J. W. Pennebaker, and J. Schler. Automatically profiling the author of an anonymous text. *Communications of the ACM*, 52(2):119–123, 2009.

[4] S. Baccianella, A. Esuli, and Sebastiani F. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *in Proc. of Language Resources Evaluation Conference*, 2010.

[5] S. Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Language Resources Evaluation Conference*, volume 10, pages 2200–2204, 2010.

[6] T. Baldwin and M. Lui. Language identification: The long and the short of the matter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 229–237. Association for Computational Linguistics, 2010.

[7] F. Barbieri, F. Ronzano, and H. Saggion. Upf-taln: Semeval 2015 tasks 10 and 11. sentiment analysis of literal and figurative language in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 704–708, Denver, Colorado, June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S15-2119.

[8] L. Barbosa and J. Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 36–44. Association for Computational Linguistics, 2010.

[9] K. R. Beesley. Language identifier: A computer program for automatic natural-language identification of on-line text. In *Proceedings of the 29th Annual Conference of the American Translators Association*, volume 47, page 54. Citeseer, 1988.

[10] Y. Bengio, H. Schwenk, J-S. Senécal, F. Morin, and J-L. Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.

[11] B. J. Boe, J. M. Hamrick, and M. L. Aarant. System and method for profiling customers for targeted marketing, May 22 2001. US Patent 6,236,975.

[12] S. Carter, W. Weerkamp, and M. Tsagkias. Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation*, 47(1):195–215, 2013.

[13] P. Carvalho, L. Sarmento, M. J. Silva, and E. De Oliveira. Clues for detecting irony in user-generated contents: oh...!! it's so easy;-). In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56. ACM, 2009.

[14] W. B Cavnar and J. M Trenkle. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175, 1994.

[15] F. Celli and L. Polonio. Relationships between personality and interactions in facebook. *Social Networking: Recent Trends, Emerging Issues and Future Outlook*, pages 41–54, 2013.

[16] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995.

[17] P. T. Costa and R. R. MacCrae. *Revised NEO personality inventory (NEO PI-R) and NEO five-factor inventory (NEO FFI): Professional manual.* Psychological Assessment Resources, 1992.

[18] T. De Smedt and W. Daelemans. "vreselijk mooi!"(terribly beautiful): A subjectivity lexicon for dutch adjectives. In *Language Resources Evaluation Conference*, pages 3568–3572, 2012.

[19] S. K. Divvala, A. Farhadi, and C. Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3270–3277. IEEE, 2014.

[20] M. Dragoni. Shellfbk: An information retrieval-based system for multi-domain sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 502–509, Denver, Colorado, June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S15-2084.

[21] A. Ghosh, G. Li, T. Veale, P. Rosso, E. Shutova, J. Barnden, and A. Reyes. Semeval-2015 task 11: Sentiment analysis of figurative language in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 470–478, 2015.

[22] M. Giménez, D. I. Hernández, and F. Pla. Segmenting target audiences: Automatic author profiling using tweets. In *Proceedings of the Conference and Labs of the Evaluation Forum*, 2015.

[23] M. Giménez, F. Pla, and L-F. Hurtado. Elirf: A svm approach for sa tasks in twitter at semeval-2015. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 574–581, Denver, Colorado, June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S15-2096.

[24] J. Golbeck, C. Robles, and K. Turner. Predicting personality with social media. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 253–262. ACM, 2011.

[25] E. M. Gold. Language identification in the limit. *Information and control*, 10(5): 447–474, 1967.

[26] M. Goldszmidt, M. Najork, and S. Paparizos. Boot-strapping language identifiers for short colloquial postings. In *Machine Learning and Knowledge Discovery in Databases*, pages 95–111. Springer, 2013.

[27] I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. Book in preparation for MIT Press, 2016. URL http://goodfeli.github.io/dlbook/.

[28] X. Han, B. Li, J. Ma, Y. Zhang, G. Ou, T. Wang, and K. Wong. Uir-pku: Twitter-opinminer system for sentiment analysis in twitter at semeval 2015. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 664–668, Denver, Colorado, June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S15-2111.

[29] L. K. Hansen, A. Arvidsson, F. Å. Nielsen, E. Colleoni, and M. Etter. Good friends, bad news-affect and virality in Twitter. In *Future information technology*, pages 34–43. 2011.

[30] Irazú Hernández-Farías, José-Miguel Benedí, and Paolo Rosso. *Pattern Recognition and Image Analysis: 7th Iberian Conference, IbPRIA 2015, Santiago de Compostela, Spain, June 17-19, 2015, Proceedings*, chapter Applying Basic Features from Sentiment Analysis for Automatic Irony Detection, pages 337–344. Springer International Publishing, Cham, 2015. ISBN 978-3-319-19390-8. doi: 10.1007/978-3-319-19390-8_38. URL http://dx.doi.org/10.1007/978-3-319-19390-8_38.

[31] J. Holmes and M. Meyerhoff. *The handbook of language and gender*, volume 25. John Wiley & Sons, 2008.

[32] K. Hornik, P. Mair, J. Rauch, W. Geiger, C. Buchta, and I. Feinerer. The textcat package for *n*-gram based text categorization in R. *Journal of Statistical Software*, 52(6):1–17, 2013. doi: 10.18637/jss.v052.i06.

[33] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[34] L-F. Hurtado, F. Pla, M. Giménez, and E. Sanchis. Elirf-upv en tweetlid: Identificación del idioma en twitter. *TweetLID@ SEPLN*, 2014.

[35] M. Kosinski, Y. Bachrach, P. Kohli, D. Stillwell, and T. Graepel. Manifestations of user personality in website choice and behaviour on online social networks. *Machine learning*, 95(3):357–380, 2014.

[36] N. Krishnamoorthy, G. Malkarnenkar, R. Mooney, K. Saenko, and S. Guadarrama. Generating natural-language video descriptions using text-mined knowledge. *NAACL HLT 2013*, page 10, 2013.

[37] G. Li and T. Ghosh, A. Veale. Constructing a corpus of figurative language for a tweet classification and retrieval task. In *Proceedings of the Forum for Information Retrieval Evaluation*, pages 130–133. ACM, 2014.

[38] Y. Lin, J. Michel, E. L. Aiden, J. Orwant, W. Brockman, and S. Petrov. Syntactic annotations for the google books ngram corpus. In *Proceedings of the ACL 2012*

system demonstrations, pages 169–174. Association for Computational Linguistics, 2012.

[39] B. Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[40] B. Liu and L. Zhang. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer, 2012.

[41] M. Lui and T. Baldwin. Accurate language identification of twitter messages. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)@ EACL*, pages 17–25, 2014.

[42] C. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[43] J.H. Martin and D. Jurafsky. Speech and language processing. *International Edition*, 2000.

[44] D. Maynard and M. A. Greenwood. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Language Resources Evaluation Conference*, pages 4238–4243, 2014.

[45] T. M. Mitchell. *Machine learning*. McGraw-Hill Boston, MA:, 1997.

[46] S. Mohammad, C. Dunne, and B. Dorr. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 599–608. Association for Computational Linguistics, 2009.

[47] S. M. Mohammad, S. Kiritchenko, and X. Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*, 2013.

[48] S.M. Mohammad, S. Kiritchenko, and X Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, June 2013.

[49] P. Nakov, Z. Kozareva, A. Ritter, S. Rosenthal, V. Stoyanov, and T. Wilson. Semeval-2013 task 2: Sentiment analysis in twitter. 2013.

[50] P. Nakov, Torsten Zesch, Daniel Cer, and David Jurgens, editors. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. As-

sociation for Computational Linguistics, Denver, Colorado, June 2015. URL http://www.aclweb.org/anthology/S15-2.

[51] D. Nguyen, N. A. Smith, and C. P. Rosé. Author age prediction from text using linear regression. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 115–123. Association for Computational Linguistics, 2011.

[52] D-P. Nguyen, R. Gravel, R.B. Trieschnigg, and T. Meder. "how old do you think i am?.ª study of language and age in twitter. 2013.

[53] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. *International Conference on Web and Social Media (ICWSM)*, 11(122-129):1–2, 2010.

[54] A. Orebaugh and J. Allnutt. Classification of instant messaging communications for forensics analysis. *Social Networks*, pages 22–28, 2009.

[55] C. Özdemir and S. Bergler. Clac-sentipipe: Semeval2015 subtasks 10 b,e, and task 11. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 479–485, Denver, Colorado, June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S15-2081.

[56] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[57] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[58] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[59] J. W. Pennebaker, M. R. Mehl, and K. G. Niederhoffer. Psychological aspects of natural language use: Our words, our selves. *Annual review of psychology*, 54(1): 547–577, 2003.

[60] V. Perez-Rosas, C. Banea, and R. Mihalcea. Learning sentiment lexicons in spanish. In *Language Resources Evaluation Conference*, volume 12, page 73, 2012.

[61] S. Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21(5):1112–1130, 2014.

[62] B. Rammstedt and O. P. John. Measuring personality in one minute or less: A 10-item short version of the big five inventory in english and german. *Journal of research in Personality*, 41(1):203–212, 2007.

[63] F. Rangel, E. Stamatatos, M. Koppel, G. Inches, and P. Rosso. Overview of the author profiling task at pan 2013. In *Conference and Labs of the Evaluation Forum Conference on Multilingual and Multimodal Information Access Evaluation*, pages 352–365. CELCT, 2013.

[64] F. Rangel, P. Rosso, I. Chugur, M. Potthast, M. Trenkmann, B. Stein, B. Verhoeven, and W. Daelemans. Overview of the 2nd author profiling task at pan 2014. In *Conference and Labs of the Evaluation Forum Evaluation Labs and Workshop*, 2014.

[65] F. Rangel, P. Rosso, M. Potthast, B. Stein, and W. Daelemans. Overview of the 3rd author profiling task at pan 2015. In *Conference and Labs of the Evaluation Forum*, 2015.

[66] A. Reyes, P. Rosso, and T. Veale. A multidimensional approach for detecting irony in twitter. *Language resources and evaluation*, 47(1):239–268, 2013.

[67] . Ritter, A. Mausam, O. Etzioni, and S. Clark. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112. ACM, 2012.

[68] A. Ritter, S. Clark, and O. Etzioni. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics, 2011.

[69] S. Rosenthal, A. Ritter, P. Nakov, and V. Stoyanov. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80, 2014.

[70] S. Rosenthal, P. Nakov, S. Kiritchenko, S. M. Mohammad, A. Ritter, and V. Stoyanov. Semeval-2015 task 10: Sentiment analysis in twitter. *Proceedings of SemEval-2015*, 2015.

[71] E. Sanchis, M. Giménez, and L-F. Hurtado. Language identification with limited resources. *V Jornadas de la Red Temática en Tratamiento de la Información Multilingüe y Multimodal (TIMM)*, pages 7–10, 2014.

[72] P. Sibun and J. C. Reynar. Language identification: Examining the issues. *Proc. Symp. Document Analysis and Information Retrieval*, pages 135–145, 1996.

[73] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer, 2013.

[74] A. Stolcke. Srilm-an extensible language modeling toolkit. In *INTERSPEECH*, volume 2002, page 2002, 2002.

[75] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[76] A. Vinciarelli and G. Mohammadi. A survey of personality computing. *Affective Computing, IEEE Transactions on*, 5(3):273–291, 2014.

[77] G. Vinodhini and R. M. Chandrasekaran. Sentiment analysis and opinion mining: a survey. *International Journal*, 2(6), 2012.

[78] U. Waltinger. Germanpolarityclues: A lexical resource for german sentiment analysis. In *Language Resources Evaluation Conference*. Citeseer, 2010.

[79] C. Whitelaw, N. Garg, and S. Argamon. Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 625–631. ACM, 2005.

[80] J. Wiebe, T. Wilson, and C. Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210, 2005.

[81] C. Zhang and P. Zhang. Predicting gender from blog posts. Technical report, Technical Report. University of Massachusetts Amherst, USA, 2010.

[82] X. Zhu, S. Kiritchenko, and S. M. Mohammad. Nrc-canada-2014: Recent improvements in the sentiment analysis of tweets. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 443–447, 2014.

[83] G. K. Zipf. Human behavior and the principle of least effort. 1949.

[84] A. Zubiaga, I. San Vicente, P. Gamallo, J. Ramom Pichel, I. Alegria, N. Aranberri, A. Ezeiza, and V. Fresno. Tweetlid: a benchmark for tweet language identification. *Language Resources and Evaluation*, pages 1–38, 2015.