UNIVERSITAT JAUME·I

TRABAJO DE FIN DE MÁSTER

MÁSTER EN SISTEMAS INTELIGENTES. CURSO 2017-18

“Generación y estudio de modelos de clasificación de perfiles de usuario de Twitter”

Autor:

Javier Tauste Martí

Tutor académico:

Rafael Berlanga Llavori

Fecha de lectura: septiembre de 2018

Índice de contenidos

Índice de figuras

Índice de tablas

1. Introducción.

La comunicación es imprescindible en el día a día de las personas para la transmisión útil de conocimiento. De forma resumida, un emisor genera un mensaje por un canal determinado para que sea consumido por otro agente. El contenido generado no solo proporciona información de un supuesto referente, sino que también, de forma indirecta y muchas veces involuntaria, de la autoría del contenido. A veces de forma tan sutil como en forma de prosodia, énfasis o incluso el canal elegido por el emisor.

Por otra parte, la influencia del no tan joven mundo interconectado ha propiciado un auge de usuarios creadores de todo tipo de contenido de forma directa en la red. Este contenido, o información, muchas veces se alberga en lugares de la red de fácil acceso donde su recopilación puede llevarse a cabo a gran escala. Tal procedimiento es muy favorable para las recientes disciplinas de extracción de conocimiento útil a partir de grandes volúmenes de información para su aprovechamiento y aumento de la eficiencia en otras áreas como: análisis del mercado, seguridad, detección de fraude o automatización y mejora de procesos de carácter humano. Por tanto, y en relación con lo antes expuesto, es posible averiguar información, muchas veces imperceptible a primera vista, de un usuario concreto a partir de datos generados en forma de multimedia o no.

Por otra parte, haciendo referencia a la literatura, la determinación de forma cuantitativa de las diferencias entre textos de varios autores es un campo antiguo denominado “Automatic Authorship Identification”. Este amplio campo se subdivide en otros 3 más específicos: “Authorship Attribution”, “Author Identification” y “Author Profiling”. En los dos primero el objetivo es determinar el autor de un corpus, el último determina características del autor, siempre en base a propiedades cuantitativas del texto [1][2]

Este trabajo se enfoca en “Author Profiling” donde se pretende analizar el contenido en formato texto producido por un usuario, tratando sus textos como datos no estructurados. Suelen ser generados en un lugar virtual, como, por ejemplo: emails, valoraciones de un servicio, conversaciones de mensajería instantánea, etc. Este enfoque se presenta como una tarea compleja ya que la pertenencia a una categoría es difusa por tratarse de categorías inherentemente subjetivas, por eso, se deben emplear métodos de clasificación supervisados, es decir, los datos empleados deben presentarse previamente clasificados [3]

Más concretamente, para lograr el objetivo suelen emplearse técnicas de “Text categorization” [4], sub-área de “Information Retrieval”, encargada de clasificar y recuperar archivos de una colección [5] y técnicas de carácter tecnológico del campo “Natural Language Processing” encargado del procesamiento y análisis por parte de los ordenadores del lenguaje humano [6].

Las aplicaciones, restringidas al mundo virtual como origen de datos, son numerosas por tratar del medio natural de comunicación humana y aun a pesar de no ser un área longeva o popularmente conocida. Aportan nuevas interfaces más integrales, facilitan la accesibilidad de muchos sistemas y automatizan procesos aportando más eficiencia en ciertas tareas. Los resultados al aplicar y desarrollar las funcionalidades asociadas son notables, por ello crecen los intereses comerciales.

Principalmente, se extrae la información no estructurada disponible, en este caso texto, de la web usando técnicas de “web scraping”. El análisis del texto obtenido permite examinar con detalle las relaciones subyacentes entre información demográfica, como edad, género o región [7], el uso del lenguaje e interacciones sociales entre usuarios de lugares web como: Twitter, YouTube, Facebook, blogs o foros de discusión.

De este modo, la clasificación de textos permite la extracción de relaciones e información que, a primera vista no parecen evidentes. Una de las utilidades más popularmente extendidas, y casi eclipsando otras utilidades, es “Sentiment analysis” o extracción de opinión [8] [9], la clasificación de textos masiva donde identifican la polaridad negativa o positiva del mensaje respecto un referente. Existen multitud de técnicas empleadas para tal fin, pero en su gran mayoría se basan en relaciones estadísticas y de asociación, no en análisis lingüístico o semántico [10]. El comercio electrónico es el gran demandante, normalmente para sistemas de recomendación automáticos, marketing masivo e imagen de marca comercial. Un caso claro fue el descontento de los usuarios de Uber en 2017, cuando, los trabajadores, ante la huelga de taxis, acudieron al aeropuerto JFK aprovechando la situación. Los usuarios respondieron de forma negativa en las redes con el hashtag “#deleteUber” [11]. Otra funcionalidad de esta aplicación es clasificar e identificar actitud de abuso o “bullying” para su tratamiento o prevención.

Dentro de las aplicaciones más famosas también se encuentran las respuestas inteligentes o “Smart replies”, la generación automática de diversas respuestas cortas al contenido de un mensaje. El sistema de clasificación de textos detecta la categoría en la que se engloba el cuerpo del mensaje y en función de ello propone un conjunto de respuestas predefinidas. Claramente este proceso mejora la eficiencia en el tiempo de latencia en las comunicaciones, ya que gran parte de las respuestas en un dialogo suelen ser poco variadas [12].

Una de las funciones más implementadas y perfectamente integradas en cualquier sistema de email es la detección de mensajes maliciosos o “Spam detection”. Se trata de un sistema de clasificación de mensajes binario que detecta si el mensaje es o no malicioso. Este es un claro ejemplo de categorización de textos y gracias al interés, la literatura y a su simpleza los resultados en la precisión son altos. Otra de las razones de su rápida integración ha sido la poca criticidad de los falsos positivos y negativos [13]

Otras aplicaciones tratadas, aunque no tan popularmente conocidas son, por ejemplo: análisis y clasificación de reseñas de productos por clientes o blogs, usado para la predicción de precios o adaptación de campañas publicitarias. Lo mismo para el mundo de la política, se persigue adaptar la campaña del candidato a la imagen que se percibe en la red. También, y relacionado con la función anterior, mediante la interacción en las redes sociales, se puede clasificar e identificar indicadores de poder e influencia, grupos sociales étnicos o tendencias populares [14]. Sin olvidar la identificación de la autoría, género, edad, región o el idioma del autor son funcionalidades básicas que pueden formar parte del desarrollo de aplicaciones más complejas.

2. Fundamentos

Antes de comenzar, y para poder aplicar adecuadamente los algoritmos de aprendizaje, se necesita un tratamiento de los datos previo, como, por ejemplo: limpieza de valores corruptos y ruido, formateo correspondiente, trasformaciones, y modificación de las dimensiones, reduciéndolas o aumentándolas. Esta última tarea es de importancia crítica, pues la extracción de características es clave para que los algoritmos de aprendizaje tengan recursos suficientes para poder generalizar y clasificar, como es este caso, de forma correcta.

2.1. Representación muestral de documentos

En la búsqueda de nuevas dimensiones se consideran dos formas de proceder, aunque en la literatura se presentan difusas y en muchos casos no diferenciadas; “Feature engineering” [15] [16] y “Feature learning” [17]. Ambos procesos se encargan de descubrir nuevos atributos del texto, en el primero de ellos, las nuevas dimensiones son elegidas, diseñadas y creadas de forma manual, normalmente a través de extracción de parámetros estadísticos, este es un proceso que muchos consideran “poco ortodoxo” ya que depende de la imaginación y es más crítico que el segundo, “Feature learning”. En este segundo las dimensiones son descubiertas de forma automática, el sistema se encarga de extraer las características directamente del texto, la extracción de estas nuevas dimensiones supone una representación del elemento más completa y por tanto es preferida ante el primer proceso.

En “Feature learning” enfocada a textos, destaca el paradigma “Word embedding”, un conjunto de lenguajes de modelado y técnicas de aprendizaje en dónde las palabras, frases o documentos enteros son vinculados a vectores de números reales, teniendo así una representación matemática en un espació n-dimensional, donde la similitud de los textos se mide por distancias como, ejemplo: euclídea, Manhattan, Hamming, Mahalanobis, la distancia del Coseno, etc. Siempre demostrado con el ejemplo “king − queen = man – woman” [18] [19].

A su vez, las técnicas en “Word embedings” pueden agruparse en dos subgrupos bien definidos. Técnicas de factorización global de matrices (FGM) y técnicas de ventanas locales contextuales (VLC). Estas técnicas presentan desventajas complementarias entre sí. FGM prioriza aspectos estadísticos frente a las analogías semánticas puras, y viceversa con VLC. FGM analiza la semántica distribucional bajo la premisa que los elementos lingüísticos de significado similar tienen distribuciones similares, para ello usa métodos de aproximación de rango bajo para descomponer grandes matrices que contienen la información estadística del corpus. En cambio, VLC se centra más en aspectos como la co-ocurrencia o la cercanía de elementos lingüísticos a otros de significado similar.

Bajo FGM se encuentran técnicas como Hyperspace Analogue to Language(HAL) [20], COALS [21] LSA o LSI [22] y bajo VLC técnicas probabilisticas condicionales como skip-grams [23] o Bag of words [24]. Pero, a partir del trabajo de Bengio et al. [25], que introdujo un pequeño modelo neuronal como herramienta de codificación de palabras a vectores numéricos, se desarrollaron nuevas técnicas más sofisticadas similares que producen resultados más útiles.

En tal representación se reflejan las relaciones semánticas como sinonimia, antonimia o analogías. Las dos técnicas más populares son: word2Vec [26] encargado de representar palabras y doc2Vec [27], una extensión del anterior pero encargado de representar documentos independientemente de su longitud. Y, aunque estos métodos son bastante eficientes en relación a su literatura anterior, existen variaciones adaptadas a entornos específicos que prometen mejoran aún más tales representaciones; es el caso de GloVe [28] que no solo aprovecha eficientemente la información estadística de los elementos distintos de cero en una matriz de ocurrencia de palabra-palabra, sino también en toda la matriz dispersa del contexto individual del corpus mejorando en tareas como identificación de similaridades o reconocimiento de entidades.

Mención especial para “Latent Dirichlet Allocation” (LDA) [29] un modelo VLC probabilístico y generativo. LDA interpreta cada muestra como un conjunto finito de probabilidades de pertenecer a ciertos tópicos. Esta técnica puede emplearse en clasificación de textos comparando los tópicos y las categorías.

Sense2Vec [30] es otra adaptación especializada en la desambiguación de todos los posibles sentidos que puede adquirir una misma palabra o frase, incluso los más sutiles como el sarcasmo. Existe Tweet2Vec [31] una versión más adecuada para este trabajo, especializada en tweets, se centra en los elementos naturales de ese ámbito como los ‘hastags’, palabras coloquiales o desconocidas, emoticonos o caracteres poco comunes.

Muchas más técnicas han sido aplicadas a lo largo de trabajos en la literatura y solo se han tratado unas cuantas de toda la totalidad. Las técnicas no mostradas aquí son de menos popularidad, de menor desempeño y eficiencia. Los procesos y técnicas aplicadas han sido los más populares, por su rendimiento o por estar especializadas en el objetivo de este trabajo. Por ello, las técnicas elegidas para desarrollar este trabajo serán elegidas en función de su rendimiento o popularidad.

2.2. Aprendizaje automático

Esto con respecto al proceso de cuantización del texto o como extraer características lo más representativas y útiles para el proceso de aprendizaje. Este proceso de aprendizaje también presenta multitud de variedades de técnicas empleadas en este ámbito en concreto. Las técnicas más frecuentes empleadas son, por lo general, elegidas por heurística, experiencia o por las características del problema ya que no hay unos patrones establecidos; conocer otros casos previos y las técnicas empleadas aporta una visión completa al inicio del trabajo que permitirá un desarrollo más eficiente.

En [32] se hace un recorrido por la literatura en el área de clasificación de textos, se recopilan y estudian 177 documentos relevantes en este campo y analiza al detalle datos de interés. Una de esas variables estudiadas son las técnicas empleadas. Como muestra la tabla 1, en los 177 documentos se identifican 13 técnicas y de cada una se describen 3 porcentajes de uso: el primero el primero si la técnica ha sido utilizada directamente en alguna etapa del proceso, el segundo parámetro si la técnica fue utilizada con el objetivo de comparar el desempeño de otra técnica principal y el tercer valor si la técnica fue mencionada en la revisión de literatura de otros documentos analizados. Como puede observase, SVM es la técnica más popular con casi un cuarto de uso directo del total. Se destaca el poco uso de las redes neuronales en relación a otras técnicas ya que su popularidad en todo tipo de estudios es muy alta. Por último, se echa en falta otras técnicas

bastante famosas, posiblemente por la selección de documentos analizados, como, por ejemplo “Random Forest” que suele tener resultados notables en otros ámbitos.

Técnica	Frecuencia de uso como método principal	Frecuencia de uso como método de comparación	Frecuencia de mención en la literatura por documentos ajenos	TOTAL
Support Vector Machine	15 (22.72%)	7 (20.59%)	13 (16.88%)	35
K-means	9 (13.63%)	5 (14.71%)	8 (10.39%)	22
K-nearest neightbors	8 (12.12%)	5 (14.71%)	11 (14.29%)	24
Naive Bayes	8 (12.12%)	5 (14.71%)	10 (12.99%)	23
Self-Organizing maps	6 (9.09%)	1 (2.94%)	3 (3.90%)	10
Latent Semantic Indexing	4 (6.06%)	1 (2.94%)	7 (9.09%)	12
Hierarchial Agglomerative Clustering	3 (4.55%)	3 (8.82%)	6 (7.80%)	12
Decision Trees	3 (4.55%)	3 (8.82%)	3 (3.90%)	9
Artificial Neuronal Network	3 (4.55%)	2 (5.88%)	2 (2.60%)	7
Association Rules	3 (4.55%)	0	5 (6.49%)	8
Case-Based Reasoning	2 (3.03%)	0	4 (5.19%)	6
Maximum Entropy Classifier	2 (3.03%)	0	4 (5.19%)	6
Multinomial Naive Bayes	0	2 (5.88%)	1 (1.29%)	3
TOTAL	66 (100.0%)	34 (100.0%)	77 (100.0%)	177

Tabla 1. Resumen estadístico de las técnicas empleadas en clasificación de textos [32]

En [33] exponen las razones de que SVM sea tan popular; **SVM es estable tratando datos de muy alta dimensionalidad**, característica de este tipo de problemas. Otra característica de este paradigma es que cada dimensión es relevante, pero **de naturaleza dispersa**, SVM también es robusto, sus resultados, comparados con el de otros algoritmos de aprendizaje, son mejores en este tipo de escenario.

2.3. Clasificación de textos

Los trabajos que emplean clasificación de textos son abundantes, aunque no tanto los relacionados con 'text classification' y mucho menos aquellos relacionados con datos procedentes de una red social masiva como Twitter, el cual plantea desafíos debido a su heterogeneidad y escala.

La clasificación de textos presenta muchas aplicaciones, una de tantas es descubrir los aspectos sociales subyacentes en el texto, o *social text analysis.* Esta área, a su vez, alberga áreas como el análisis de sentimientos y la categorización de la autoría o *author profiling.* En la primera, sentiment analysis, se evalúan aspectos como la subjetividad, la polaridad o el *aspect* del texto mientras que en la segunda, *author profiling*, se suelen averiguar características de autoría como: el sexo y la edad, tratadas en el trabajo [34], o categorías más concretas. En este trabajo se trata el último camino, la clasificación de los textos para averiguar aspectos de *authority*. En la tabla 2 se muestran las áreas comentadas, resaltada la linea seguida por este trabajo.

Social text analysis	Sentiment analysis	Subjectivity	[0, 1]
		Polarity	[0, +, -]
		Aspect	[sonido, robustez...]
	Author profiling	Sexo	
		Edad	
		Authority	**influencer**
			categories

Tabla 2.Contextualización de las áreas de acción en la clasificación de textos. Resaltada la línea seguida en este trabajo

En los trabajos [35], [36] y [37] alcanzan un una precisión (accuracy) del 80% aplicando etiquetado morfosintáctico o POS-tagging (Part Of Speech tagging) a cada una de las palabras en textos de más de 250 palabras [38]. Aunque la complejidad en el aprendizaje automático aumenta con la disminución del tamaño de los textos estudiados; por ejemplo, Zhang and Zhang [39] clasificó el sexo de los autores en textos de 15 palabras, alcanzando un 72.1% de precisión. Nguyen et al. [40] clasificaron la edad de los autores en un espectro continuo con una regresión logística alcanzando una correlación del 0.75 y un error medio aproximado de 5 años. Los *datasets* de entrenamiento clasifican los textos en tres clases de rango de edad: menores de 20 años(-20), entre 20 y 40 años (20-40) y mayores de 40 años (40-), aunque la regresión genera un espectro continuo en un rango de entre 10 a 70 años de edad. La dificultad de este último reside en la brevedad de los textos, ya que usaron la red social Twitter donde la media de palabras por mensaje no supera las 10 palabras.

En la literatura también se encuentran ejemplos poco convencionales, como en [41] donde se emplean redes neuronales convoluciones (CNN) para la clasificación directa de un documento dado. El modelo usado emplea convoluciones unidimensionales por cada frase del documento para más tarde poder hacer una suma ponderada de cada vector obtenido. El resultado final es la categoría del documento. Alcanzan precisiones del 76.14% en comparación al 74% que consiguen métodos SVM o al 80% que consiguen los humanos en la misma tarea.

Cabe destaca la capacidad y los resultados notables que se obtienen al extraer dimensiones elaboradas a partir del texto mediante '*feature learning*', muchas de ellas inherentes al campo de minería de textos aquí tratado. Una de las estrategias son los modelos de representación de textos estadísticos y probabilísticos como los usados en los trabajos [42] y [43] que usaron, junto con técnicas antes comentadas: la frecuencia de aparición de los signos de puntuación, mayúsculas, citas, o características intrínsecas a una cadena simple de texto. Es decir, trabajos con características que representan la muestra y sean discriminativas ante un proceso de clasificación.

Por otra parte, no hay que olvidar el objetivo por el que se emplean estas técnicas, poder clasificar la categoría a la que pertenece un autor concreto. En este trabajo, y a diferencia de la mayoría de trabajos antes nombrados, no se centra en categorías como edad o sexo, sino en categorías de rol; es decir, qué función desempeña el autor al elaborar el texto, o conjunto de textos analizados. Dicho de otra forma, qué tipo de autor es el que hay detrás de un conjunto de textos.

El texto de los mensajes puede ser suficiente para una clasificación de calidad, por el estilo del usuario reflejado y por el contenido del mensaje como, por ejemplo, la opinión. Pero, adelantándose al punto "Metodología", se pretende completar la representación de la muestra con datos adicionales no textuales, cualquier cuantización del perfil del usuario como, por ejemplo: la frecuencia de mensajes, tiempo invertido en la red, número de interacciones con otros usuarios, etc. Todo con el objetivo de encontrar esas características representativas que ayuden al modelo de aprendizaje a una discriminación de mayor calidad.

3. Escenario de aplicación

3.1. Twitter como base para clasificación de textos

Twitter es uno de esos lugares *on-line* donde se expone contenido de forma masiva y abierta de fácil obtención, el cual constituye un servicio popular en forma de red social basada en *microblogging*. Creado en 2006 y con 332 millones de usuarios registrados en menos de 10 años, permite la interacción de usuarios a través la emisión pública de mensajes de texto cortos, restringidos a los 280 caracteres.

El origen de esta plataforma tiene controversia por no haber consenso en el lugar o empresa en la que se gestó. A pesar de ello, sus creadores Evan Williams, Biz Stone, Jack Dorsey, Evan Henshaw-Plath y Noah Glass, gestaron la idea en la empresa Odeo en paralelo al desarrollo de un proyecto interno. Con la fama que más tarde alcanzaría este nuevo producto se creó Obvius Corporation, compradora de Odeo, aunque en 2007 nace Twitter Inc. independiente, sin gestora.

Su nombre no fue tampoco claro desde los inicios. Sus creadores querían captar la idea de los SMS, un nombre que indicara una actualización constante. Comenzaron con “Stat.us” y tras otros nombres y lluvia de ideas nació el nombre de Twitter haciendo alusión al pio de un pájaro, queriendo transmitir "una ráfaga corta de información intrascendente". Esta red social hace su primera aparición el 15 de julio del 2006, y el primer tweet fue realizado por Jack Dorsey a las 12:50 pm, que decía “just setting my twtrr” (ajustando mi twtrr) [44] [45].

La influencia social de este servicio es alta, existe una fuerte presencia de instituciones oficiales como forma de comunicación informal al público en general. No solo actúa como una red social, en los últimos años Twitter también ha aportado gran valor como medio de comunicación y herramienta de marketing digital. La creación de etiquetas compartidas (hastag), indicadas al inicio con el símbolo almohadilla (#), su uso masivo en los mensajes y el carácter publico de los textos hacen que una etiqueta pueda ser tratada por muchos usuarios. No solo las etiquetas, los contenidos multimedia también son clave en la retrasmisión de información a nivel global. Estos dos factores han propiciado una retransmisión casi en directo de cualquier evento de gran importancia. Los casos más impactantes son los desastres naturales, se han cubierto con gran rapidez por los usuarios, llegando a contrarrestar la lenta reacción de los servicios competentes [46] [47].

La ilustración 1 expone una infografía [48] con estadísticas generales que reflejan mejor la importancia de esta infraestructura. Con estos datos es fácil hacerse la idea de que se puede encontrar: la mitad de los mensajes tiene alguna fotografía como contenido multimedia, solo el 26% son mensajes originales o 33 idiomas diferentes.

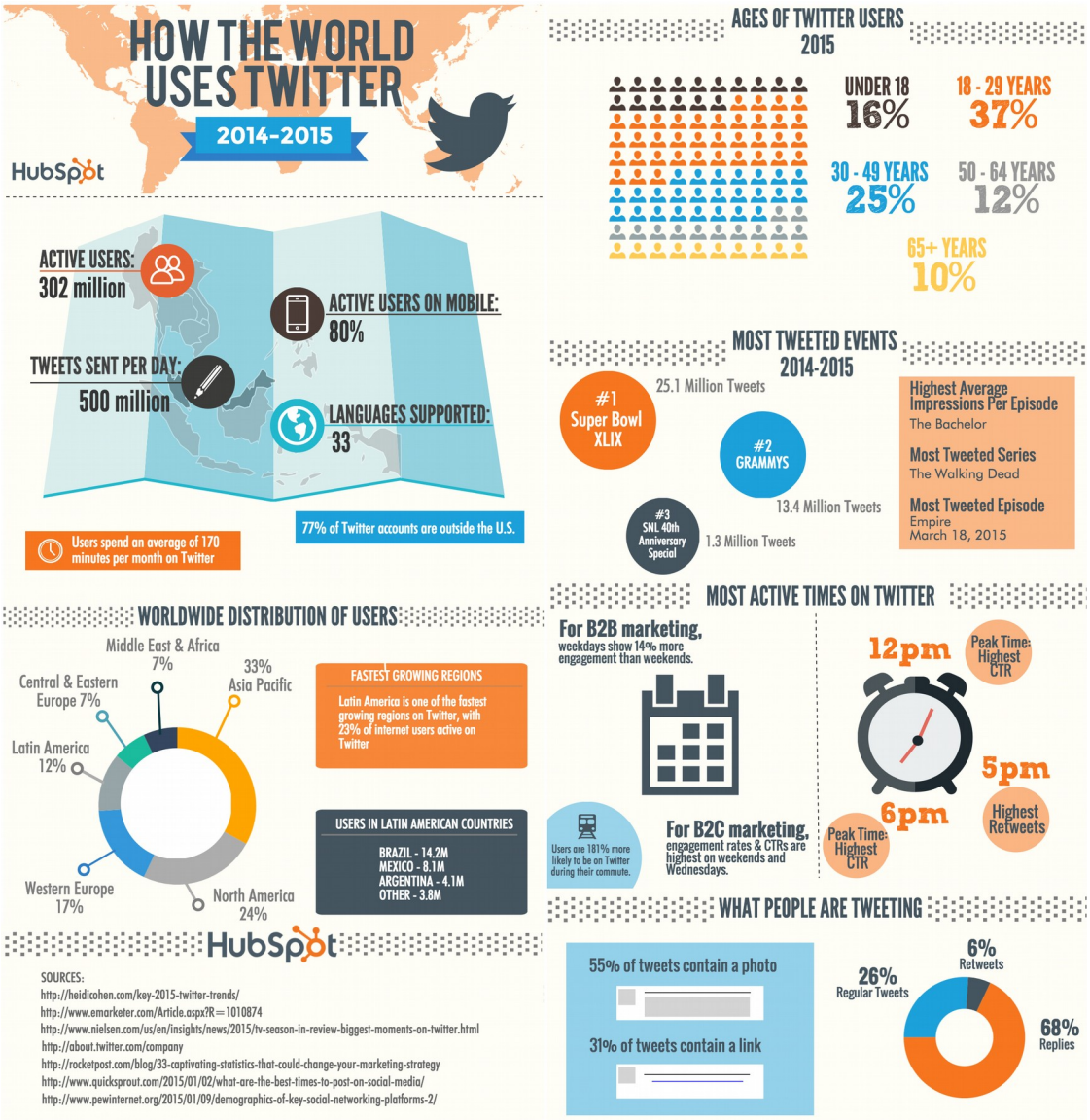


Ilustración 1. Infografía estadística mundial sobre Twitter

Tanta es la huella social que muchos de los conceptos y características han sido aceptados por la RAE por su uso extendido. En la tabla 3 se muestra una lista de los conceptos más importantes, muchos de ellos se usan más adelante para describir el funcionamiento de la plataforma.

Concepto	Descripción
tweet	Mensaje de longitud máxima 280 caracteres. Puede contener todo tipo de símbolos: letras, signos, enlaces, etc. Su significado hace referencia al pio del pájaro como alusión a la brevedad del mensaje.
follow	Acción de seguir o estar suscrito para poder observar la acción de un usuario o evento concreto. Hay importantes derivados como 'follower' (usuario suscrito) o 'unfollow' (acción de dejar de estar suscrito)
retweet	Acción de mencionar un mensaje de otro usuario. Normalmente por mostrar acuerdo o simpatía con el mensaje.
mención	Acción de mencionar un usuario en un texto de un mensaje determinado. Se indica con el símbolo @ seguido del nombre del usuario, ejemplo @Javier_Taus
trending topic	Mención de un evento que por ser mencionado por un número elevado de usuarios se considera de gran importancia. Los eventos se mencionan con el símbolo # seguido del nombre del evento, ejemplo #spanishrevolution
influencer	Tipo de usuario caracterizado por su gran 'influencia' sobre otros usuarios, ya sea difundiendo información o una opinión. Suelen tener, en relación con el resto de usuarios, un número elevado de 'followers'
spammer	Tipo de usuario que suele generar gran cantidad de contenido no deseado por otros usuarios. Intención maliciosa o poco adecuada para tal entorno. Comúnmente de finalidad comercial.

Tabla 3. Conceptos más usados en Twitter

El funcionamiento de la red social es simple, la parte principal es el muro, donde se muestra todo el contenido que los usuarios, a los que sigues, han creado en orden cronológico en forma de lista vertical. Los usuarios pueden seguir otros usuarios, quedarán suscritos y sus mensajes aparecerán en su muro. El sistema de seguimiento es asimétrico, a comparación de Facebook, donde los usuarios pueden acceder a aceptar a otra persona y ambos convertirse en amigos, en Twitter un usuario puede seguir a otro, pero eso no significa que el otro usuario lo siga. No se requiere el consentimiento mutuo en ambos usuarios. Cualquier mensaje tiene tres posibilidades públicas, puede ser respondido, en forma de hilo secuencial, puede ser 'retuiteado' o señalado como favorito. Esta metodología presenta un factor diferenciador, un usuario cualquiera puede ponerse en contacto con un personaje público de forma inmediata. Los mensajes, como se ha indicado antes, no solo pueden albergar texto sino contenido variado como fotografías, video, emoticonos o enlaces a otros lugares web.

En Twitter la información queda de forma pública, no se requieren permisos, el contenido está limitado y organizado de forma cronológica. Los eventos o etiquetas también son organizados, pero de forma territorial, y su carácter es menos personal que otras redes sociales y más enfocado a la comunicación pública. Estas características hacen de Twitter una plataforma idónea para el análisis social y económico de muchas empresas. Por otro lado, esta plataforma presenta información no estructurada pero sí bien organizada.

3.2. Análisis de datos sociales en Twitter

La clasificación de autores es una tarea que no solo requiere de la clasificación de los textos, puede apoyarse en otro tipo de datos para alcanzar su objetivo. Todos estos datos deben tener origen común para ser consistentes y coherentes, deben presentarse con un gran número de muestras para que los algoritmos de aprendizaje puedan llegar al punto óptimo entre la generalización y la concreción. Como es lógico, los datos deben proceder de Twitter. Pero no son extraídos directamente de la plataforma web como un usuario corriente, ese procedimiento no es el más adecuado, se tardaría demasiado tiempo en llegar a recolectar una colección de muestras suficiente.

Twitter dispone de una API [49] o interfaz de programación de aplicaciones, que permite la extracción de esa información de forma automática. El uso de la API ahorra tiempo y elimina la carga gráfica de la aplicación web nativa. Se puede extraer no solo información de los mensajes y sus repercusiones, sino incluso de metadatos útiles como: número de seguidores, número de usuarios que sigue, descripción propia del usuario, región, fecha de creación del perfil, etc. El uso de la API suele requerir complejidad y tiempo para recolectar datos, por eso, se tratarán datos ya recolectados por terceros, más concretamente de dos orígenes distintos: RepLab2014 y la infraestructura SLOD-BI. Los datos de ambas fuentes serán al final integrados en un mismo volumen de datos.

RepLab [50] es una competición apoyada por el proyecto europeo LiMoSINe [51]. Su objetivo es fomentar la investigación y desarrollo del tratamiento de la reputación online y proveer una base de colaboración entre instituciones académicas y profesionales mediante campañas de laboratorios donde el diseño y evaluación de las tareas son elaboradas de forma conjunta por ambos grupos.

En las ediciones previas a RepLab 2014 los objetivos han sido: resolución de entidades, detección de tópicos, polaridad de la reputación de los usuarios y detección de tópicos dañinos para la reputación. RepLab siempre se ha enfocado en el contenido proporcionado por Twitter por las razones antes descritas.

RepLab 2014 proponía 2 objetivos principales, clasificación de las publicaciones por dimensiones de reputación y clasificación de los perfiles de usuarios (Author profiling). El primer objetivo contribuye a un mejor entendimiento del tópico subyacente de un tweet o grupo de estos, mientas que el segundo objetivo aporta información importante para crear un ranquin de tweets en función del tipo de usuario que hay detrás del contenido. El primer objetivo no se desarrolla más por no tener relación con la meta de este trabajo. El segundo objetivo, a su vez, está dividido en otros dos sub-objetivos, la creación de un sistema de generación de ranquing de autores en función de su poder de influencia y la creación de un sistema de clasificación de los usuarios. Las categorías de clasificación por tipo de autor establecidas en esta segunda sub-tarea han sido: ‘Company’, ‘Professional’, ‘Celebrity’, ‘Employee’, ‘Stockholder’, ‘Investor’, ‘Journalist’, ‘Sportsman’, ‘Public institution’ y ‘Non-Governmental Organizations’. Este trabajo se enfoca a cumplir el segundo objetivo principal de RepLab 2014, como comentaremos de forma más precisa en el siguiente punto.

La API de Twitter es excelente pero no perfecta, presenta restricciones que pueden ralentizar la extracción masiva de datos. Dependiendo del tipo de API, la persistencia de los datos y el tipo de información disponible varía [52]. Para poder elaborar un gran conjunto de datos y que la persistencia no sea un problema, como es el caso de este trabajo, se debe usar ‘Streaming API’ que proporciona un flujo de *tweets* en casi tiempo real al establecer una conexión permanente con los servidores. En la ilustración 2 se observa fácilmente cuales son los datos facilitados por este proceso.

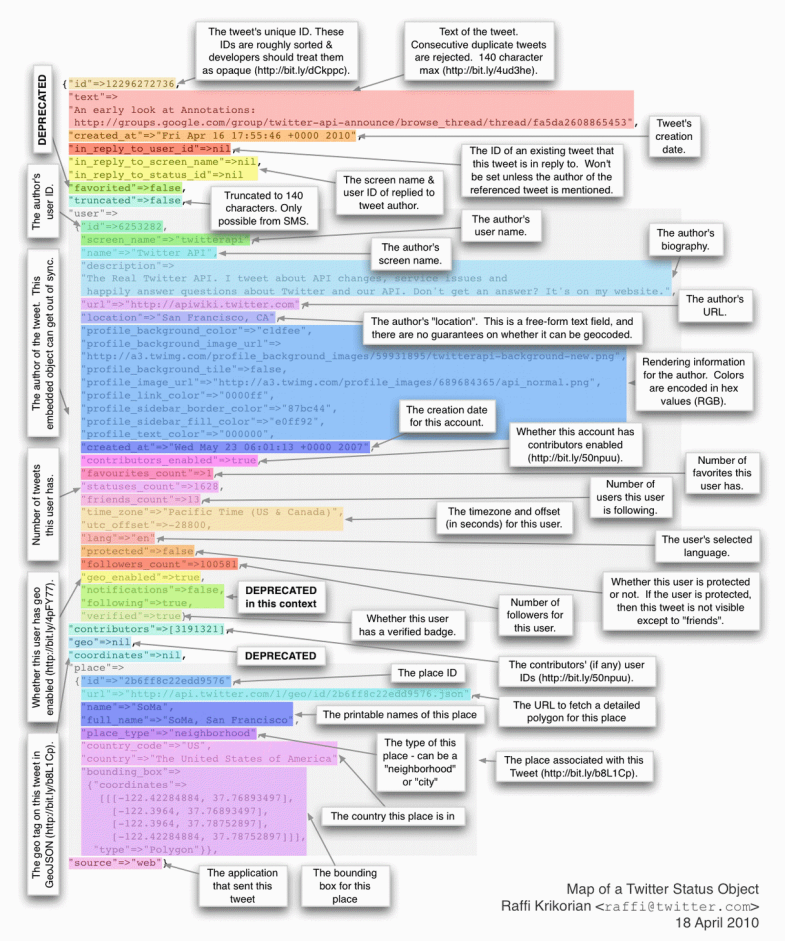


Ilustración 2. Datos disponibles para extracción en la API

SLOD-BI (Social Linked Open Data for Business Intelligence) es una infraestructura de datos abiertos que permite tareas de extracción, carga, manipulación y análisis de grandes cantidades de datos en la web 3.0, datos de opinión y sentimientos procedentes de foros y redes sociales. Es una infraestructura enfocada a la inteligencia de negocios (BI), presenta los principales patrones de dicha área de forma integrada. Este sistema es innovador por poder llevar a cabo las tareas principales integrando datos de diferente naturaleza, datos corporativos generados en la misma empresa y datos sociales públicos. Para ello, SLOD-BI sigue los principios del paradigma de datos abiertos enlazados (LOD) mediante tecnologías de web semántica, como RDF, y de naturaleza gratuita y de libre utilización. La principal ventaja es la integración de datos propios junto con opiniones externas relacionadas con el negocio, una innovación frente a las tradicionales técnicas de BI [53].

Para capturar los datos que contextualicen y enriquezcan los corporativos, la infraestructura integra las principales aplicaciones de comercio electrónico: Voice of the Customer (VoC) y Voice of the Market (VoM). Prestar atención a VoM aporta una importante dirección estratégica de negocio basada en la visión global de los consumidores principales, mientras que prestar atención a VoC ayuda a identificar y afiliar mejor los clientes, como señala [54] ambas perspectivas son una importante ventaja competitiva a largo plazo.

La estructura interna de esta infraestructura está construida mediante módulos funcionales y preserva los valores antes nombrados y, también, de forma integral. Cada módulo ejecuta

una capacidad de la infraestructura. Conceptualmente se puede representar como en la ilustración 3, dos capas concéntricas, la primera de almacenamiento de datos y la segunda con los vocabularios externos. La primera capa o núcleo alberga conjunto de datos mantenidos y situados de forma independiente pudiendo estar en maquinas separadas, cada conjunto tiene una finalidad distinta. Las uniones entre elementos representan el tipo de operación posible, la línea en negrita, uniones entre conjuntos y la delgada, posibles conexiones entre entidades. En este trabajo se emplea el módulo 'Social facts' del núcleo del sistema.

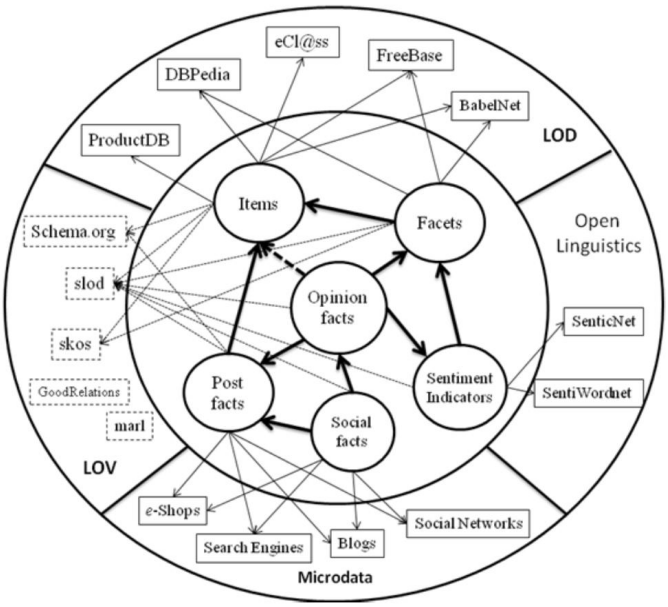


Ilustración 3. Vista estructural de SLOD-BI

Retomando la primera parte de este punto, SLOD-BI se encarga también de la extracción de los datos de esos datos abiertos, usa la API de Twitter. Los datos extraídos serán los encargados de, como establece una de sus objetivos principales, enriquecer y contextualizar aquellos datos que se tienen previamente. 'Social facts' es el módulo de almacenamiento encargado de albergar tales datos.

4. Objetivos

En este punto se exponen los objetivos de este trabajo con más detalle. Gracias a los puntos anteriores se conoce el objetivo principal, 'Author profiling', poder clasificar los autores de Twitter por categorías preestablecidas a partir de datos de su perfil. Es necesario resaltar que este no es un objetivo original, es heredado de 'RepLab 2014' como se ha señalado anteriormente y, por tanto, las categorías de la clasificación también son las mismas. También recordar que los datos con los que trabajamos en un inicio son los proporcionados por este evento y restringidos al ámbito automovilístico.

Pero esta meta no es suficiente, no basta con la eficacia, se requiere también eficiencia. Se pretende superar otras marcas, mejorar los resultados de otros trabajos previos que hayan intentado este mismo objetivo. En el documento de RepLab 2014 se presentan los diferentes grupos, sus técnicas empleadas y los resultados de cada uno. Cada grupo emplea una metodología diferente y por tanto existe cierta liberta de acción. Este segundo objetivo pretende mejorar los resultados mostrados en este artículo [50].

Uno de los trabajos de referencia a superar no está presente en ese artículo. Se trata de [55], a partir de ahora "Twitter sólo texto" por brevedad. Es el documento referencia principal, a estos resultados se les presta más atención ya que sus marcas que definen la calidad de sus resultados están por encima de los trabajos presentados en RepLab 2014. En este trabajo sólo se procesa información textual procedente de los usuarios, no meta-datos.

Para ello, y porque la distribución de frecuencias de las categorías es bastante desbalanceada a favor de 'Professional' y 'Journalist', siguen una metodología particular. Sólo clasifican los usuarios con las categorías más comunes, 'journalist' y 'pofessional', si están comprendidos en la lista de 'influencer'. 'Journalist' y 'pofessional' los definen como grupos de autoridad, es decir, aquellos que influencian con sus opiniones y aquellos que esparcen información con facilidad respectivamente. La metodología seguida se muestra en la ilustración 4. Clasifican, por una parte, el resto de categorías del conjunto 'influencer' y, por otra parte, y de forma paralela, identifican tal conjunto de influencia para más tarde clasificar el resto de categorías, resolviendo así el problema del desbalance.

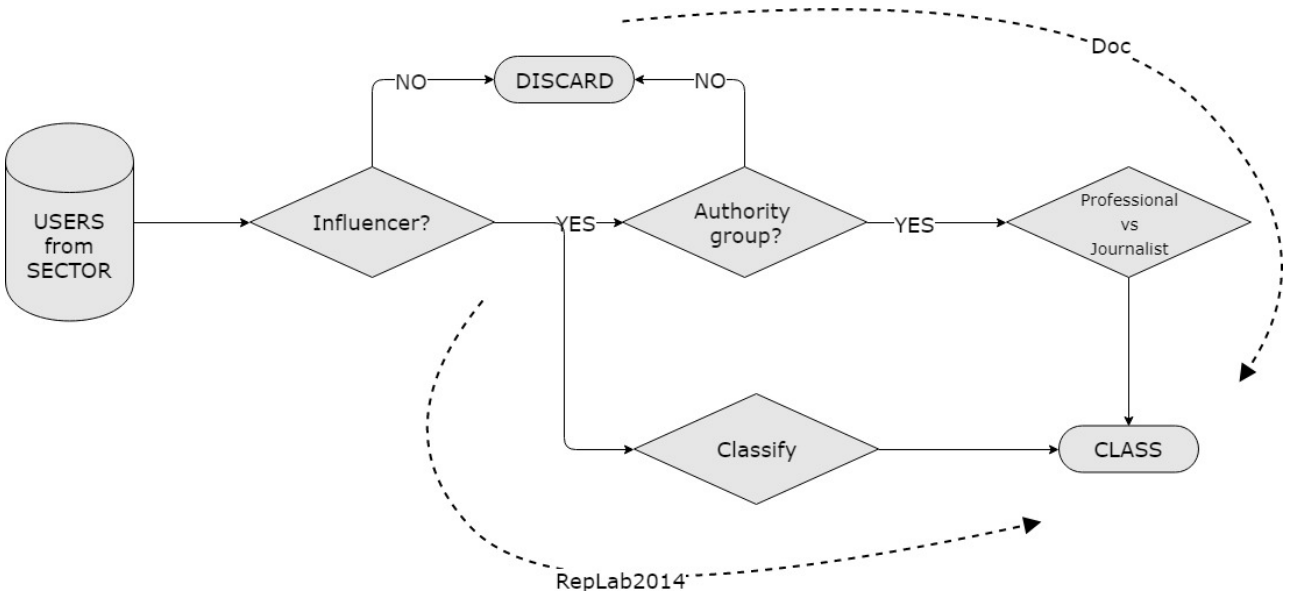
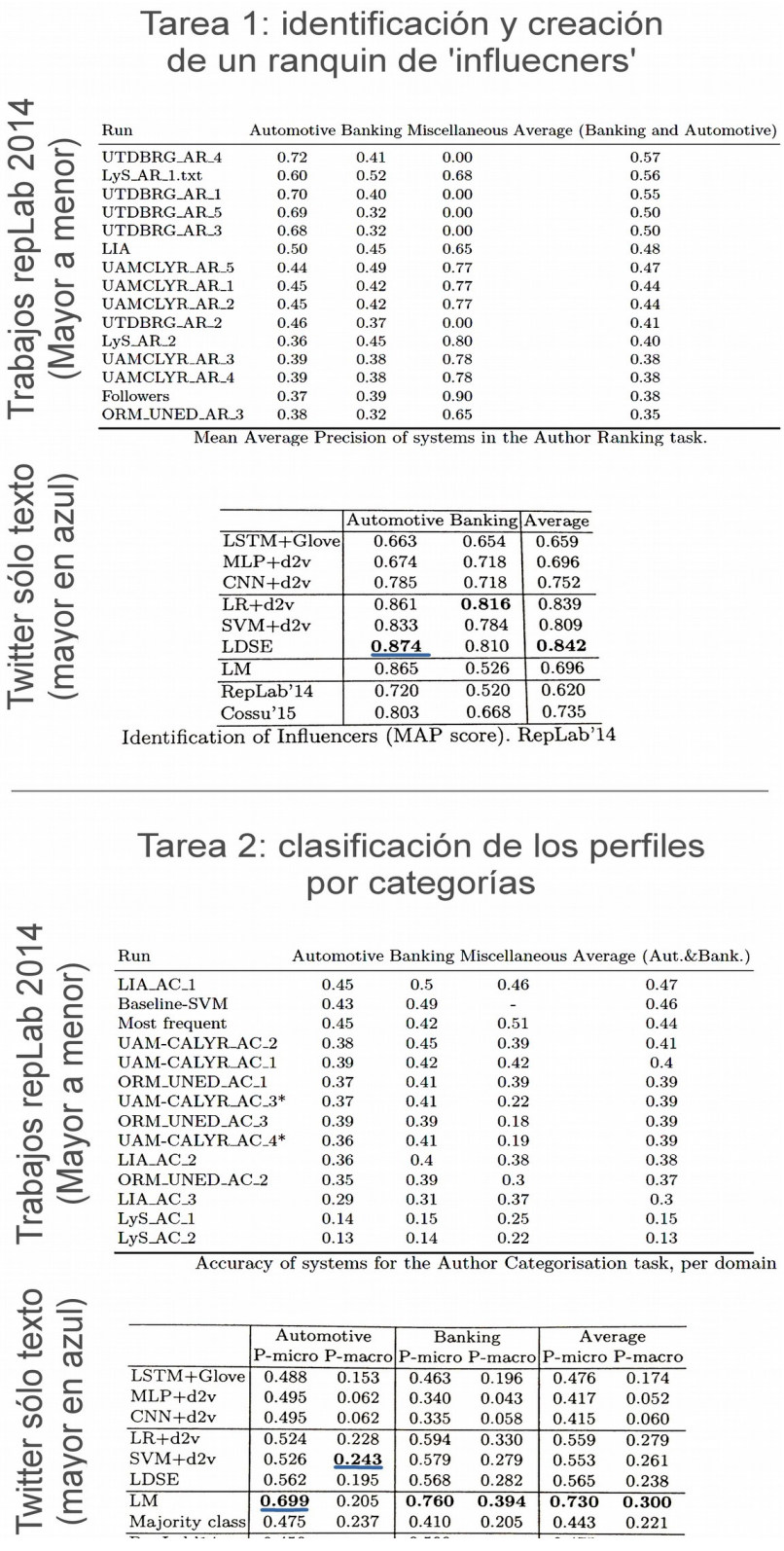


Ilustración 4. Taxonomía de la metodología del documento objetivo [55]

Los tres mejores resultados expuestos en el documento RepLab 2014 han sido para los grupos denominados UTDBRG, LyS y LIA. El primero [56] sustenta la hipótesis de que los autores influyentes comunican gran cantidad de mensajes sobre tópicos populares. El segundo [57] sólo usa meta-datos del perfil de usuario como la imagen de usuario, etiqueta del nombre, enlaces relacionados, etc. Y el último [58], LIA, solo alcanza usa el texto asociado a los mensajes del usuario y los meta-datos relacionados con cada mensaje. En la ilustración 5 se muestran los resultados de estos 3 grupos y el trabajo “Twitter sólo texto” en las tareas de clasificación y reputación (identificación y creación de un ranquin de ‘influencers’). En cada tarea usan una métrica distinta pero común entre los trabajos, para la clasificación multi-etiqueta emplean ‘accuracy’ y para el ranquin usan ‘mean average precision’.



Run	Automotive	Banking	Miscellaneous	Average (Banking and Automotive)
UTDBRG_AR_4	0.72	0.41	0.00	0.57
LyS_AR_1.txt	0.60	0.52	0.68	0.56
UTDBRG_AR_1	0.70	0.40	0.00	0.55
UTDBRG_AR_5	0.69	0.32	0.00	0.50
UTDBRG_AR_3	0.68	0.32	0.00	0.50
LIA	0.50	0.45	0.65	0.48
UAMCLYR_AR_5	0.44	0.49	0.77	0.47
UAMCLYR_AR_1	0.45	0.42	0.77	0.44
UAMCLYR_AR_2	0.45	0.42	0.77	0.44
UTDBRG_AR_2	0.46	0.37	0.00	0.41
LyS_AR_2	0.36	0.45	0.80	0.40
UAMCLYR_AR_3	0.39	0.38	0.78	0.38
UAMCLYR_AR_4	0.39	0.38	0.78	0.38
Followers	0.37	0.39	0.90	0.38
ORM_UNED_AR_3	0.38	0.32	0.65	0.35

Mean Average Precision of systems in the Author Ranking task.

	Automotive	Banking	Average
LSTM+Glove	0.663	0.654	0.659
MLP+d2v	0.674	0.718	0.696
CNN+d2v	0.785	0.718	0.752
LR+d2v	0.861	**0.816**	0.839
SVM+d2v	0.833	0.784	0.809
LDSE	**0.874**	0.810	**0.842**
LM	0.865	0.526	0.696
RepLab'14	0.720	0.520	0.620
Cossu'15	0.803	0.668	0.735

Identification of Influencers (MAP score). RepLab'14

Run	Automotive	Banking	Miscellaneous	Average (Aut.&Bank.)
LIA_AC_1	0.45	0.5	0.46	0.47
Baseline-SVM	0.43	0.49	-	0.46
Most frequent	0.45	0.42	0.51	0.44
UAM-CALYR_AC_2	0.38	0.45	0.39	0.41
UAM-CALYR_AC_1	0.39	0.42	0.42	0.4
ORM_UNED_AC_1	0.37	0.41	0.39	0.39
UAM-CALYR_AC_3*	0.37	0.41	0.22	0.39
ORM_UNED_AC_3	0.39	0.39	0.18	0.39
UAM-CALYR_AC_4*	0.36	0.41	0.19	0.39
LIA_AC_2	0.36	0.4	0.38	0.38
ORM_UNED_AC_2	0.35	0.39	0.3	0.37
LIA_AC_3	0.29	0.31	0.37	0.3
LyS_AC_1	0.14	0.15	0.25	0.15
LyS_AC_2	0.13	0.14	0.22	0.13

Accuracy of systems for the Author Categorisation task, per domain

	Automotive		Banking		Average	
	P-micro	P-macro	P-micro	P-macro	P-micro	P-macro
LSTM+Glove	0.488	0.153	0.463	0.196	0.476	0.174
MLP+d2v	0.495	0.062	0.340	0.043	0.417	0.052
CNN+d2v	0.495	0.062	0.335	0.058	0.415	0.060
LR+d2v	0.524	0.228	0.594	0.330	0.559	0.279
SVM+d2v	0.526	**0.243**	0.579	0.279	0.553	0.261
LDSE	0.562	0.195	0.568	0.282	0.565	0.238
LM	**0.699**	0.205	**0.760**	**0.394**	**0.730**	**0.300**
Majority class	0.475	0.237	0.410	0.205	0.443	0.221

Ilustración 5. Ranquin de resultados de los trabajos referencia previos

Se observa fácilmente los resultados más altos en los cuatro documentos referencia, “Twitter sólo texto” supera los tres grupos con mejores resultados de RepLab2014, aunque, bajo un método diferente en cada tarea. Para cumplir este segundo objetivo se plantea seguir la misma filosofía que ellos; encontrar un factor diferenciador que altere las condiciones iniciales, ya sea una metodología diferente o un conjunto de datos que permita al modelo poder aprender los valores más representativos.

Al final del trabajo se emplean modelos de aprendizaje para poder obtener las nuevas categorías mediante inferencia. Se usan distintos modelos de aprendizaje para poder observar y analizar su desempeño ya que no se sabe con total certeza como de adecuado es un modelo en función del contexto. Para ello, y como tercer objetivo se pretende el uso de modelos tradicionales y modelos” novedad”; modelos como ‘Naive Bayes’, Arboles, máquinas de soporte vectorial (SVC) y modelos de redes neuronales artificiales. Se espera que este último genere un mayor rendimiento.

A modo resumen y para facilitar la lectura los tres objetivos son:

- Predecir el perfil de un usuario dado a partir de los datos obtenidos de su cuenta de Twitter.

- Superar la calidad de las predicciones de trabajos previos mediante un factor diferenciador.

- Emplear, analizar y comparar modelos de carácter tradicional y de reciente popularidad.

5. Metodología

En este punto se exponen aquellas técnicas, métodos, tecnologías y formas en las que se lleva a cabo el trabajo. Primero se indican los pasos se siguen y en cuales de ellos se prestará más atención. Más tarde, las técnicas y algoritmos que se probarán en el desarrollo del experimento. Y por último las tecnologías elegidas para llevar a cabo el proceso de desarrollo. Por último, es necesario destacar la ambigüedad de este punto, poco ortodoxo, no se siguen unas pautas más definidas a favor de cumplir los objetivos marcados a costa de una metodología concreta. Por ese motivo no se desarrolla un punto de planificación temporal, se desconocen los resultados, el tiempo de aprendizaje y prueba de los experimentos.

5.1. Técnicas y algoritmos

Para poder llevar a cabo los objetivos antes comentados se ejecutan pasos comúnmente conocidos: Recolección de los datos, exploración de estos, preparación, construcción de los modelos de aprendizaje y ajustar hiper-parámetros. El despliegue del sistema no entra en los límites de este trabajo y la recolección de los datos se ha explicado en el punto “Origen de los datos”. La exploración de los datos ya ha sido brevemente introducida con las estadísticas de la distribución de clases ya mostrada, aunque se analizará con más detalle. Este trabajo se centra principalmente en la preparación de los datos y la construcción de modelos de clasificación que cumplan con los objetivos ya comentados.

En la introducción se han repasado las técnicas más populares que se han empleado en este ámbito, tanto las técnicas de extracción de dimensiones o representación muestral como las técnicas de aprendizaje máquina más adecuadas. Respecto la representación muestral, las técnicas más empleadas en la mayoría de trabajaos previos han sido “Word embeddings” aunque existen trabajos que complementan los datos obtenidos por esta técnica con la extracción de características manual obteniendo mejores resultados. ‘Support Vector Machine’ (SVM) es el modelo más usado en la mayoría de trabajos previos estudiados, el resto de modelos le siguen desde lejos.

Por ende, se pretende seguir los mismos pasos. Se pretende representar el cuerpo de los mensajes mediante la representación numérica de los “Word embeddings”, también, se consideraría la posibilidad de crear nuevas características en el caso de que los resultados no superaran el segundo objetivo. No solo se representarán los textos de los usuarios sino todas aquellas variables del perfil que puedan ayudar a la clasificación. El conjunto de datos principal, como se ha comentando anteriormente, se completará con los datos aportados por SLOD-BI; variables adicionales como:

- 'favourites_count' : Núm. de favoritos que ha dedicado el usuario de la cuenta
- 'followers_count' : Núm. de seguidores de la cuenta
- 'friends_count' : Núm. de amigos de la cuenta
- 'statuses_count' : Núm. de estados de la cuenta
- 'onDomain_tweets' : Núm. de mensajes relacionados con el ámbito automovilístico

Se empleará SVM como una de los algoritmos para el aprendizaje. Se empleará también 'Random Forest' por proveer buenos resultados en otros ámbitos y no aparecer en el estudio de la tabla 1. Como tercera técnica de aprendizaje principal, se usarán redes neuronales artificiales por tratarse de un método que goza de gran popularidad y buenos resultados.

5.2. Tecnologías

Para llevar a cabo el trabajo se han empleado herramientas concretas que cumplen funciones específicas, desde aportar un entorno de desarrollo hasta el análisis específico de datos. La elección de estas tecnologías ha sido por: familiaridad, facilidad de aprendizaje, gran documentación en la red y, por supuesto, su uso extendido en este tipo de trabajos. A continuación, se describen brevemente:

Python: es un versátil y sencillo lenguaje de programación con potentes y de gran eficiencia estructuras de datos, pero orientado a objetos. Interpretado y con semántica dinámica. El atractivo de este lenguaje reside en su legibilidad y baja curva de aprendizaje, tiene una sintaxis sencilla y permite el desarrollo de aplicaciones a más velocidad que otros lenguajes de programación. Este lenguaje y sus librerías son gratuitas y de libre distribución, por ello tiene una gran comunidad [59]. Se usará la última versión, la 3.6.

Las siguientes herramientas son, o pueden ser, usadas en Python:

- Jupyter Notebook: es un novedoso entorno de trabajo open source orientado a científicos que soporta los lenguajes R y Python. Ofrece una shell interactiva vía web, a la que se puede acceder desde un navegador. La shell está organizada en pequeños bloques, cada bloque puede contener texto arbitrario formateado en Markdown o comentarios, fórmulas matemáticas en LaTeX, código en multitud de lenguajes, resultados, gráficos, vídeos, widgets o cualquier elemento multimedia [60].

- Pandas: es una biblioteca de software escrita como extensión de NumPy para manipulación y análisis de datos para el lenguaje de programación Python. En particular, ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales y herramientas para su manipulación. Es un software libre distribuido bajo la licencia BSD [61].

- Keras con Tensorflow backend: Keras es una API de redes neuronales de alto nivel, escrita en Python y capaz de ejecutarse sobre TensorFlow, CNTK o Theano, en este caso, el primero. Fue desarrollado con un enfoque en permitir la experimentación rápida. Poder pasar de la idea al resultado con la menor demora posible es la clave para hacer una buena investigación. Admite gran variedad de topologías avanzadas y permite su ejecución tanto en CPU como en GPU [62].

- Gensim: es un API robusta de extracción de tópicos y modelado matemático open-source implementado en Python. Se apoya en NumPy, SciPy y Cython. Especializado en el tratamiento de grandes colecciones de datos usando 'streaming' de datos y algoritmos incrementales altamente eficientes. Incluye implementaciones como: tf-idf, random projections, algoritmos word2vec y document2vec, hierarchical Dirichlet processes (HDP), latent semantic analysis (LSA, LSI, SVD) y latent Dirichlet allocation (LDA), incluyendo distributed parallel versions [63].

- NLTK: es un conjunto de bibliotecas y programas para el procesamiento del lenguaje natural (PLN) simbólico y estadísticos para el lenguaje de programación Python. NLTK incluye demostraciones gráficas y datos de muestra. Cumple casi con las mismas funcionalidades que Gensim. Alberga librerías para clasificación, tokenization, stemming, etiquetado, parsing, y razonamiento semántico, así como soporte para librerías de carácter industrial para NLP [64].

- Scikit Learn: es un conjunto de librerías para aprendizaje automático, presenta algoritmos para clasificación, regression y clustering. También algoritmos de reducción de la dimensionalidad, selección de modelos y preprocesador de datos. Técnicas como SVM, random forest, gradient boosting, k-means y DBSCAN. Diseñado para interoperar junto con librerías como NumPy. Es una de las librerías con esta funcionalidad de uso más extendido [65].

- Numpy: es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices. NumPy es open source [66].

- MatplotLib: es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy [67].

El equipo con el que se desarrolla y se llevan a cabo la mayoría de experimentos es un ordenador portátil MSI GE60 2PC con una CPU i7-4710HQ, GPU GTX850m y 12Gb de RAM. Aunque cabe señalar la ayuda de una máquina más potente para la parte de aprendizaje y ajuste de hiperparámetros.

6. Desarrollo

El desarrollo del experimento se ha dividido en 2 partes principales: el tratamiento de los conjuntos de datos provenientes del punto “Origen de los datos” para creación del conjunto de datos principal, que contendrá toda la información a procesar, y la aplicación de aprendizaje máquina sobre ese conjunto de datos final.

6.1. Minería y tratamiento de los datos

6.1.1. Descripción de los conjuntos

Los datos proporcionados para el concurso RepLab 2014 son usados para este trabajo como grueso principal o base de partida. Estos, están divididos en 2 ámbitos, bancario y automovilístico, aunque también hay un tercer ámbito misceláneo; por acotar el trabajo se reduce el ámbito del trabajo al automovilístico. A pesar de esta delimitación, se describe todo el conjunto de datos al detalle como en el documento. Los datos contienen alrededor de 7000 perfiles de usuario, cada perfil de usuario contiene el nombre del usuario, sus últimos 600 tweets y el lenguaje de cada uno de ellos. Además, expertos en reputación y categorización han analizado y completado los datos de forma manual indicando que perfil de usuario es ‘influencer’ y a que categoría pertenece.

Estos conjuntos de datos principal se presentan divididos con 2500 perfiles para el conjunto de entrenamiento y 4991 perfiles para el conjunto de validación, ello implica el conjunto de test interno al conjunto de entrenamiento. En el ámbito automovilístico, los datos contienen 1313 perfiles de usuario en el conjunto de entrenamiento y 2493 perfiles el conjunto de evaluación.

En la tabla 4 se muestran las estadísticas de los datos totales, se observa la distribución de los tipos de perfiles con algunos de muy escasos. Centrando la atención en el ámbito automovilístico, se observa el desbalance, la mayoría de muestras se localiza en 4 clases de 9 totales. La clase ‘Undecidable’ con la mayoría de 38.7%, seguido de ‘Professional’ con un 26.6%, el 17.2% de ‘journalist’ y el 8% de ‘company’. Las 5 clases restantes acumulan solo un 9.5%. Una posible explicación podría ser su poca actividad en la plataforma en un periodo de extracción desafortunado. Otro punto a observar, sólo con el análisis que este documento proporciona, es el alto número de usuarios influyentes, más de lo esperado. Este aspecto recae en la categorización manual de los expertos.

	Automotive			Banking			
Class	**Training**	**Test**	**Total**	**Training**	**Test**	**Total**	**Total**
Undecidable	454	0	454	556	0	556	1010
Professional	312	358	670	279	286	565	1235
Journalist	202	171	373	258	231	489	862
Company	94	119	213	51	33	84	297
Sportsmen	49	36	85	8	4	12	97
Celebrity	27	24	51	33	7	40	91
NGO	21	5	26	78	83	161	187
Public Inst.	12	3	15	27	30	57	72
Employee	3	8	11	1	6	7	18
Total	1174	724	1898	1291	680	1971	3869

Tabla 4. Estadísticas distribución usuarios en RepLab 2014

En la tabla 5 se observa una pequeña muestra de 5 muestras 'tipo' variadas. Se observan dimensiones no explicadas antes, pero fáciles de distinguir por el nombre. Las dimensiones se comentarán en detalle más en adelante.

class	influencer	id_message	name	id_domain	lang.	message
und.	0	421839135872409600	Ruddy_edson	RL2014D02	en	When this woman comes on your telly that's when you know it's late ðŸ˜ pic.twitter.com/V70K5cnuG1
journalist	1	430808256358133760	MailOnline	RL2014D03	en	He could have murdered us in our sleep' http://dailym.ai/1fF1KqpÂ
company	1	423053741642620928	unitetheunion	RL2014D04	en	Press release: Weymouth bus drivers to strike on Monday in pay dispute - http://shar.es/9LWmPÂ
prof.	1	427587474836434944	Themoneygame	RL2014D05	en	Gold is rallying http://read.bi/1d453CBÂ
ngo	0	425336654379745280	GloriaMitchell6	RL2014D02	en	Yes, exactly. Thanks @DennisMLynch for this pic.twitter.com/wUnZeFyzUh

Tabla 5. Muestras 'tipo' de los datos de RepLab 2014

El objetivo de esta parte es obtener un único conjunto de datos que represente cada usuario del que se tienen datos. Para ello se pretende extraer, transformar, manipular y agrupar los conjuntos de datos originales. Se parte de los conjuntos de datos que muestra la tabla 6, donde se aprecian los diferentes tamaños. También, dependiendo del origen, las muestras representan o un usuario o un mensaje de un usuario. El objetivo de los conjuntos de datos procedentes de SLOD-BI es cumplir el segundo objetivo propuesto, poder mejorar los resultados con un elemento diferenciador. Estos datos deben aportar ese factor diferenciador.

Origen	Nombre	N muestras	Descripción
RepLab 2014	Train automóviles	788.693	Mensajes y datos asociados, usuario diferente cada 600 mensajes
	Test automóviles	1.501.307	Mensajes y datos asociados, usuario diferente cada 600 mensajes
SLOD-BI	Métricas adicionales	629.143	Muestra por usuario, variables que reflejan su actividad
	Descripción	863.187	Muestra por usuario, texto de descripción propia del perfil

Tabla 6. Conjuntos de datos de partida

6.1.2. Exploración y peculiaridades

Antes de desarrollar es necesario conocer los datos al detalle, las exploraciones permiten sacar conclusiones y características que definen el proceso a seguir en el trabajo, como, por ejemplo:

Como curiosidad, en la primera exploración, y enfocado al ámbito automovilístico, se descubren anomalías naturales de los conjuntos de datos. Referencias a variaciones espaciales o culturales como pueda ser los diferentes nombres que recibe la marca ‘Opel’ en el Reino Unido (Vauxhall) y el resto de Europa. Incluso referencias temporales. Los conjuntos de SLOD-BI han sido recolectados previos a una fecha, pero a pesar de ello aparecen modelos de coche más recientes a esta fecha, por ejemplo, el caso del ‘Fiat toro’, esto es debido a las referencias de los mensajes que contienen comparativas y enfrentan modelos más antiguos con estos más recientes.

Por las diferencias entre los conjuntos, ya comentadas antes, existe el problema evidente que no puede haber una intersección completa entre los conjuntos, aun a pesar de agrupar 600 muestras de mensajes en un solo usuario habrá muestras de usuario sin la información del conjunto más pequeño. En el estudio de las intersecciones de esos conjuntos se descubre efectivamente que de 3806 usuarios totales de RepLab 2014, solo coinciden 560 con métricas procedentes de SLOD-BI. Con las descripciones de los perfiles o ‘profiles’ ocurre algo similar, la intersección de conjuntos también es mínima. Además, existe la libertad de no completar dicho campo, muchos perfiles quedan también incompletos. Estos hechos presentarán problemas más en adelante. La función principal de los conjuntos de métricas SLOD-BI se ve diluida por este hecho.

Por su naturaleza y origen se considera el conjunto de muestras de RepLab 2014 como datos de ‘calidad’. Se da por cierto que cada muestra ha sido seleccionada y extraída bajo criterios minuciosos por expertos para que: sean coherentes con el ámbito relacionado, aunque por el contrario parece no ser así en una exploración rápida, los usuarios no pertenezcan a grupos maliciosos o no se presenten dispersos sino, al menos, casi completos.

En otra breve exploración de los datos de SLOD-BI, no en RepLab2014, se descubren perfiles de usuario con comportamientos anómalos; en especial en sus mensajes, altamente repetidos, casi siempre con intenciones comerciales, el contenido multimedia, como enlaces, redirige a sitios maliciosos, poco adecuados o están obsoletos.

Una vez detectados estos, en una exploración ya más detallada se descubren ciertos patrones en común: sus descripciones de usuario suelen emplear un conjunto de palabras concretas como: 'auction', 'highly ', 'item', 'offer', 'deal', 'product', 'extremely', 'opulent',

'stunning', 'grand', 'shop', 'buy', 'bargain', 'offer', 'sale', 'discount', 'exquisite', 'exclusive', 'magnificent' y 'review'. De estos, el patrón clave para su identificación se hallan al inicio de sus mensajes, usuarios que a simple vista no deberían tener nada en común, el inicio comienza con pares de números como “06 07 08 09 10 11 12”. Estos usuarios son tratados como ‘Spammers’, usuarios con la descripción de antes cuyas intenciones son malignas o de beneficio poco ético.

Como se ha explicado antes, en RepLab 2014 no existen ‘Spammers’ por ser considerado un conjunto de calidad y si unos pocos en SLOD-BI. Por tanto, se extrae mediante selección de la infraestructura un tercer conjunto de datos con 1611 perfiles identificados como ‘Spammers’. La finalidad de este conjunto es, no la de mejorar los resultados de trabajos previos como propone el segundo objetivo sino la de poder adecuarse a datos más complejos y sobre todo reales. La intención es insertar este conjunto para aprender a detectarlos.

Se experimenta también con la detección de usuarios ‘spammer’ dentro del conjunto de RepLab2014. Solamente con la ayuda de las métricas aportadas por SLOD-BI se ejecuta una clasificación binaria simple mediante ‘random forest’ y SVM sobre el nuevo conjunto. Se pretende averiguar la calidad que aportan las métricas añadidas por SLOD-BI, razón por el uso exclusivo de estas dimensiones. El conjunto de datos lo integran los 1611 usuarios ‘spammer’ detectados previamente y 560 usuarios de RepLab2014 que coinciden con el conjunto de SLOD-BI y por tanto tienen métricas. Los resultados son los esperados, la precisión supera el 98% incluso mediante el uso de cada métrica/dimensión por separado. Se concluye y se corrobora la hipótesis que el conjunto de RepLab2014 no posee ningún usuario ‘spammer’.

Haciendo referencia a las tres peculiaridades antes expuestas se debe abordar otro breve análisis. El objetivo es averiguar si las métricas que aporta SLOD-BI coincidentes con RepLab 2014, teniendo este como referencia de datos de calidad, y el resto de métricas muestran patrones diferenciados, lo suficiente como para, en un trabajo futuro de mejora, la extracción de los datos de SLOD-BI sean más semejantes a los datos de RepLab 2014. De ese modo se asegurarían usuarios con comportamientos estándar y sus datos y métricas favorecerían un aprendizaje con menos. La ilustración 6 muestra los histogramas de las distribuciones normalizadas de las 5 métricas. Como se enfrentan conjuntos de tamaño desbalanceado se observa esa falta de muestras, como por ejemplo en la gráfica de ‘OnDomain_count’. Excepto en ‘favourites_count’ y ‘OnDomain_count’, en el resto de gráficas sí que ser aprecia claramente una tendencia diferenciadora al alza del contenido de estas variables. La frecuencia de valores altos es mayor en el conjunto “de calidad”. Por tanto, se reafirma por una parte la calidad, o al menos la diferencia, entre los conjuntos de SLOD-BI y RepLab, y por otra parte la necesidad de aumentar la calidad del trabajo en una línea futura en la extracción de muestras de enriquecimiento de la infraestructura.

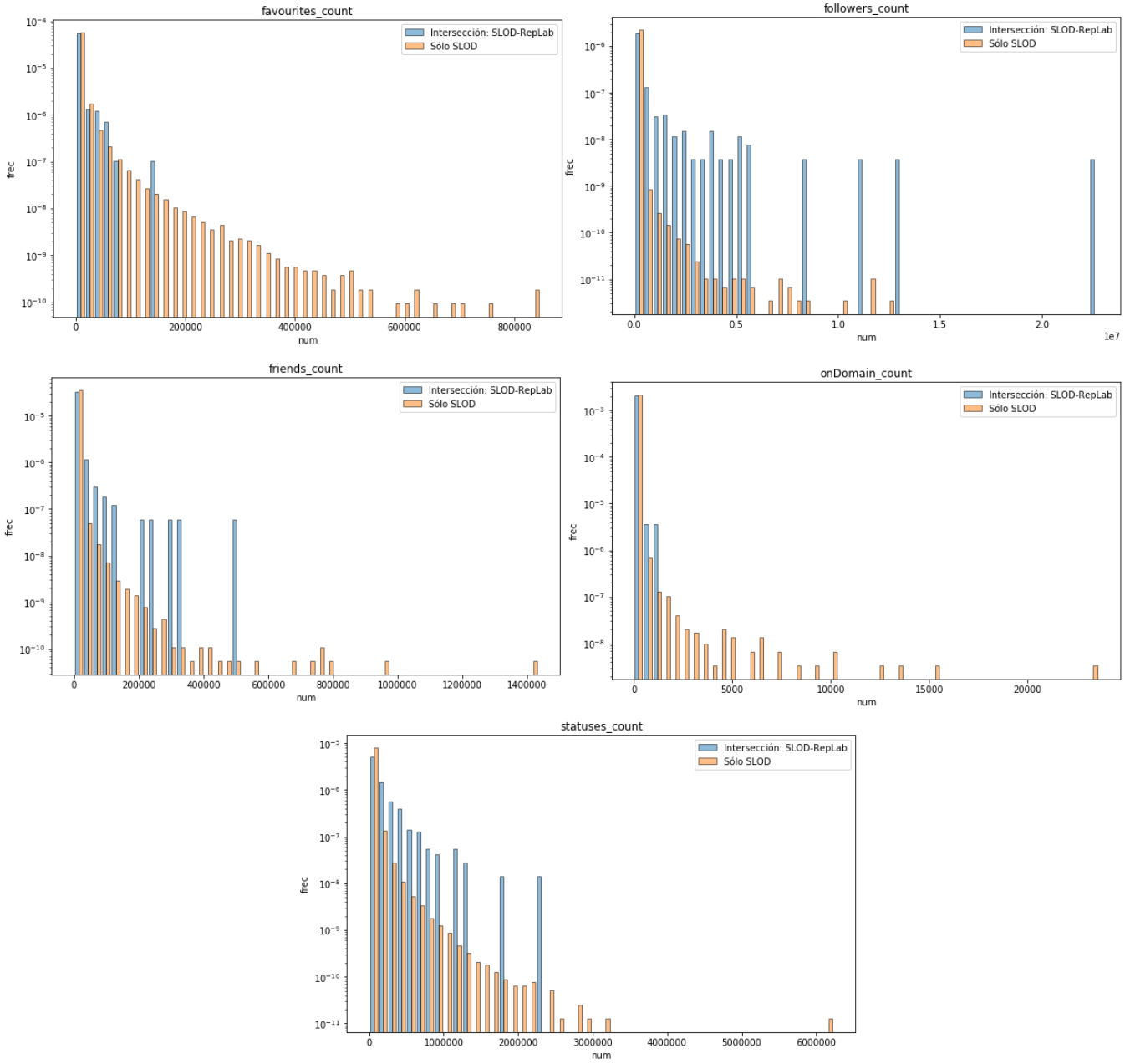


Ilustración 6. Comparativa de las distribuciones de las métricas entre el pequeño conjunto con métricas de RepLab2014 y el resto del conjunto con métricas no presentes en RepLab2014.

6.1.3. Uso exclusivo de métricas

Llegados a este punto y con las observaciones previas, es necesario añadir un nuevo objetivo al trabajo. Este es la identificación de los grupos maliciosos 'Spammer' de forma previa a cualquier otro proceso. Así, descartando este conjunto de usuarios, se clasificarán, en las categorías correspondientes, solo los usuarios 'no spammer'. La razón de ello es evidente, por ejemplo, no podría existir, o no existe de normal, un usuario caracterizado por su gran influencia y que fuera malicioso. Las siguientes líneas van dedicadas a detectar este tipo de usuarios.

En un primer experimento, se quiso comprobar como de relevantes y/o representativas son las 5 métricas. Para ello se pretende una clasificación binaria simple usando solo esas 5 métricas. Los conjuntos son los usuarios identificados claramente 'spammers', gracias al nuevo conjunto de 1611 usuarios extraídos deliberadamente, y 'no spammers', de aquellos 560 usuarios coincidentes entre ambos conjuntos. Se emplea un conjunto de datos con la etiqueta 'spammer' y las 5 métricas. Los resultados son curiosos, se alcanza un 'accuracy' superior al 95%. Incluso cada dimensión por separado, es decir, empleando un conjunto con la etiqueta y una de las 5 métricas, es capaz de obtener un resultado no inferior al 94%. Posiblemente, y como se ha visto en la ilustración 6, cada métrica por separado es crucial para determinar la clase spammer. Se refuerza la hipótesis que los usuarios de RepLab2014 son de buena calidad, no existen 'spammers'.

Este hecho es sospechoso y, por ese motivo se ejecuta, con el modelo de clasificación binaria aprendido, una predicción sobre el conjunto de datos de SLOD-BI excluyendo los coincidentes con RepLab. De esta predicción se obtiene un 65% de los usuarios son de naturaleza 'Spammer', algo nada raro teniendo en cuenta que los datos obtenidos por SLOD-BI no son de 'buena calidad'. Sin embargo, al explorar los resultados y los datos, al examinar detenidamente los usuarios clasificados como 'spammer', se considera que la calidad de la clasificación es bastante baja, usuarios con comportamientos nada maliciosos y con contenido común son considerados 'spammer'. Una posible explicación de ello se deba a que este grupo, los falsos positivos, no hablen suficiente respecto el ámbito y que, por ello, la dimensión 'onDomain_count' adquieran valores bajos. La categoría 'spammer' si presenta esta dimensión con valores pequeños. Esta baja calidad se puede perfeccionar mediante la clasificación manual de expertos. Recolectar muestras del conjunto predicho que se observen claramente, y bajo el criterio del experto, como usuarios 'no spammer' e incrustar dichas muestras en el conjunto de entrenamiento, pudiendo así disminuir el numero de falsos positivos.

A pesar de las adversidades y la poca calidad de los resultados del experimento anterior, se continua, aunque no a partir de sus resultados. El problema de un exceso de falsos positivos no es un impedimento cuando se deben tratar los verdaderos negativos, simplemente reduce el conjunto más de lo que debiera. Aunque no se trabajará sobre la predicción sino sobre los 'no spammers' de RepLab que tienen métricas, los 560 que intersecan con SLOD-BI. Se persigue ahora la clasificación binaria de 'influencers' y 'no influencers' solamente sobre esas 560 muestras. No es un conjunto representativo por su tamaño y posiblemente esté sesgado. De la clasificación se obtiene un rendimiento del 60% de 'accuracy', para nada aceptable.

Las clasificaciones binarias simples han sido implementadas de la misma forma. Se han empleado modelos de árbol como un árbol de decisión y 'random forest' con un ajuste automático de hiper-parámetros para optimizar el aprendizaje. Al ser ambos aprendizajes de muestras desbalanceadas se ha muestreado de forma estratificada para compensar la desventaja.

Los dos procesos comentados han sido ejecutados con éxito, pero con excesivos inconvenientes. Los conjuntos de datos no son adecuados y los resultados obtenidos no son buenos. No se comprueba la relevancia de las 5 métricas por no haber suficientes muestras para un aprendizaje de calidad. Se deja de lado el uso exclusivo de las métricas como línea de acción.

6.1.4. Uso exclusivo de texto: LDA

LDA (Latent Dirichlet Allocation), ya señalada al inicio de este trabajo, es un modelo generativo que, entre otras funcionalidades, permite la extracción de tópicos en documentos de forma generativa. Este modelo asume a priori que cada documento es una combinación de probabilidades de categorías o tópicos. LDA sigue una filosofía de "Bolsa de palabras", el orden no importa, que el uso de una palabra es ser parte de un tema y que comunica la misma información sin importar dónde se encuentra en el documento. Esta hipótesis dice que Juan contrató a Pedro es lo mismo que Pedro contrató a Juan. En ambos casos, el conjunto de palabras es la misma junto con la frecuencia de cada palabra. Este supuesto es necesario para que las probabilidades sean intercambiables y que permitan una mayor aplicación de métodos matemáticos. Muy empleado en la minería de textos ya que encuentra aquellos tópicos que mejor representan un conjunto de textos. Puede resumir grandes cantidades de texto de forma automática o descubrir patrones latentes. Cada tópico es, a su vez, un conjunto de probabilidades de palabras. Un ejemplo claro es, teniendo estos 5 documentos:

- Doc 1: He desayunado un sándwich de crema de cacahuete.
- Doc 2: Me encantan las almendras, nueces y cacahuetes.
- Doc 3: Mi vecino adoptó ayer un perro.
- Doc 4: Los perros y gatos son enemigos naturales.
- Doc 5: No deberías darle cacahuetes a tu perro.

El modelo LDA se encarga de descubrir los tópicos latentes en este conjunto de documentos y cuales de ellos representan mejor cada documento. Así, puede encontrar las siguientes estructuras:

- Tópico 1: 30% cacahuete, 15% almendra, 10% desayuno… (posiblemente referido a la comida)
- Tópico 2: 20% perros, 10% gatos, 5% cacahuete… (posiblemente referido a mascotas o animales)

Documento 1 y 2 representados por: 100% tópico 1.
Documento 3 y 4 representados por: 100% tópico 2.
Documento 5 representado por: 70% tópico 1 y 30% tópico 2.

En este segundo experimento, a diferencia del anterior, se busca emplear el máximo número de muestras usando solamente los mensajes de los usuarios. Pero, el modelo LDA solo permite la extracción de información, no es un modelo de clasificación. La solución es crear tópicos artificiales. En los mensajes de todos aquellos usuarios establecidos en una categoría determinada se encola una palabra claramente identificativa y distinguible. Se propone el sistema "@categoría@", de este modo, los tópicos generados presentan esta palabra junto con otras relacionadas a la categoría. De ese modo, para clasificar un usuario sin categorizar se extraen los tópicos de sus mensajes y se comparan con los tópicos relacionados con cada categoría. El resultado es un ranquin de porcentajes de acercamiento o similitud de lo que suelen hablar los usuarios con esas mismas categorías. Por ejemplo, para detectar aquellos usuarios 'spammer' se encolan en los mensajes de estos el token "@Spammer@" y en el resto el token "@NoSpammer@". Ese token esta en todos los mensajes y por eso aparece en todos los tópicos extraídos. En esta línea de acción no solo se tratan los mensajes, sino también, los profiles o descripciones de usuario de forma paralela ya que en las exploraciones se observa claramente la diferencia entre el discurso entre clases de usuario, en especial los 'spammers'.

El desarrollo y creación de este sistema es simple gracias a la librería Gensim, basada en Python y antes comentada. También se desarrolla la capacidad de calcular la perplejidad del modelo. El primero es una métrica común para medir y comparar el poder predictivo de que tan seguro está el modelo de que la predicción sobre el vocabulario es correcta o que tan perplejo está. Por ejemplo, en un dado sería de 6 ya que ante un número anterior la probabilidad del resto es la misma, en este caso, cada iteración estaría igual de perplejo. A menor perplejidad más seguridad de predicción.

El tiempo de ejecución del sistema LDA creado es excesivamente alto, sólo con el conjunto de 'profiles' de usuario se interpreta una velocidad de cálculo de 100.000 muestras por hora. SLOD-BI aporta 863.187 muestras o 'profiles', lo que supone casi 9 horas. A pesar del tiempo, el resultado obtenido tampoco es esperanzador por varias razones, los caracteres no latinos como: cirílicos, japoneses, árabes, griegos, hebreos... son interpretados como palabras independientes y los tópicos extraídos están compuestos por este tipo de palabras, siendo en realidad caracteres y desvirtuando la interpretabilidad de los tópicos. Incluir estos como 'stopwords' es inviable. En la ilustración 7 se muestra un ejemplo de los 11 primero tópicos. Se aprecia también la distribución poco homogénea de la influencia de los tokens en algunos tópicos, se considera pues un camino fallido.

```
topic #0 (0.020): 1.000*"м" + 0.000*"0.000 + "ↄ*"ы" + 0.000*"á" + 0.000*"л" + 0.000*"0.000 + "1"*0.000 + "ᵢ"*0.000 + "ᵧ"*0.0
topic #1 (0.020): 0.020*"ↇ" + 0.020*"ы" + 0.020*"л" + 0.020*"0.020 + "1"*0.020 + "ᵢ"*0.020 + "0"*0.020 + "ↄ"*0.020 + "ᵎ*"ñ"
topic #2 (0.716*"3" + 0.284*"a" + 0.000*"ы" + 0.000*"л" + 0.000*"0.000 + "0"*0.000 + "ↄ"*0.000 + "ᵎ*"ↇ" + 0.000*"1"*
topic #3 (0.020): 1.000*"0.000 + "ↄ"*0.000 + "ᵧ*"á" + 0.000*"л" + 0.000*"0.000 + "ᵎ*"ы" + 0.000*"ↇ" + 0.000*"0.000 + "1"*0.0
topic #4 (0.020): 0.020*"ↇ" + 0.020*"ы" + 0.020*"л" + 0.020*"0.020 + "1"*0.020 + "ᵢ"*0.020 + "0"*0.020 + "ↄ"*0.020 + "ᵎ*"ñ"
topic #5 (0.020): 1.000*"e" + 0.000*"0.000 + "ↄ*"ы" + 0.000*"л" + 0.000*"0.000 + "0"*0.000 + "ᵎ*"ↇ" + 0.000*"ᵧ"*0.000 + "1"*
topic #6 (0.020): 1.000*"6" + 0.000*"ы" + 0.000*"л" + 0.000*"0.000 + "0"*0.000 + "ↄ"*0.000 + "ᵎ*"ↇ" + 0.000*"1" + 0.000*"é"
topic #7 (0.020): 0.020*"ↇ" + 0.020*"ы" + 0.020*"л" + 0.020*"0.020 + "1"*0.020 + "ᵢ"*0.020 + "0"*0.020 + "ↄ"*0.020 + "ᵎ*"ñ"
topic #8 (0.020): 1.000*"к" + 0.000*"ↇ" + 0.000*"é" + 0.000*"л" + 0.000*"0.000 + "1"*0.000 + "ᵢ"*0.000 + "0"*0.000 + "ↄ"*0.0
topic #9 (0.020): 1.000*"é" + 0.000*"0.000 + "ↄ*"ы" + 0.000*"л" + 0.000*"0.000 + "0"*0.000 + "ᵎ*"ↇ" + 0.000*"1" + 0.000*"ñ"
topic #10 (0.020): 0.686*"ñ" + 0.314*"0.000 + "ↄ"*0.000 + "ᵾ*"ы" + 0.000*"л" + 0.000*"0.000 + "1"*0.000 + "ᵢ"*0.000 + "0"*0.
```

Ilustración 7. Ejemplo de tópicos extraídos por LDA

Otra consideración, como reflexión a posteriori, es la imposibilidad de este sistema de extraer los tópicos de real interés al ámbito de acción estudiado. Son, por ejemplo, tópicos referidos al automovilismo como: presentaciones o reseñas de coches o modelos con 12.000 ejemplos en los mensajes, avisos de policía o robos de coche con la misma frecuencia de aparición que la anterior, compra venta y alquiler de estos con 16.000 ejemplos de muestras en los mensajes, la nueva tendencia de los automóviles eléctricos con también 16.000 ejemplos en el corpus, la retirada del nuevo modelo Alfa Romeo o tópicos tan mediáticos como el caso de fraude de emisiones de Volkswagen.

El uso del modelo LDA es una línea de acción que aporta valor y diferenciación respecto otras líneas de acción de los trabajos a los que se compara este. No solo eso, sino que promete una base sólidamente razonada a priori de la representación de los datos con la totalidad de las muestras y solo usando datos textuales, línea de acción empleada en [68]. Por desgracia, y por los inconvenientes comentados esta forma no ha resultado ser conveniente. Se abandona esta línea para investigar otras formas de representación más rigurosas.

6.1.5. Creación del ‘dataframe’ principal

El uso exclusivo de métricas no es adecuado por muchas razones, principalmente por la falta de datos. Tampoco es adecuado el uso exclusivo del texto mediante LDA por las muchas razones antes señaladas. La línea de acción que ahora se busca en la extracción de todas las características de todas las formas posibles, automática y manual. Se intenta pues extraer en máximo numero de dimensiones que representen fielmente una muestra dada a partir de datos no dispersos, mayormente completos en todo el espectro de muestras. El objetivo de este punto es crear un conjunto de datos único que albergue todos los usuarios, tanto los de entrenamiento como los de test y los inyectados como ‘spammer’ y que albergue las características lo suficientemente adecuadas como para que el sistema de aprendizaje generalice e infiera correctamente las categorías. Para ello, las nuevas características son extraídas de forma manual y mediante ‘word embeddings’ (automática)

Como revisión se debe tener en mente las 3 clases objetivo: influencer (binaria), spammer (binaria) y categoría (multiclase). Sin embargo, la clase objetivo ‘spammer’ ha sido introducida en este trabajo y su naturaleza es incompatible con las clases ya definidas. Por ejemplo, una ‘Celebrity’ o un ‘Stockholder’ normalmente no suelen ser perfiles ‘spammers’ que obedezcan a la definición aquí tenida en cuenta; no suelen tener un carácter malintencionado. Esta es la razón de poder considerar, a partir de este momento, la categoría ‘spammer’ como una categoría más de las tantas dentro de la clase objetivo. Ahora se tienen sólo 2 clases objetivo, una binaria y otra multiclase.

A partir de este momento se explican los cuatro frentes diferentes enfocados a la extracción de características como ya antes se ha comentado. Las dos primeras se encargan de la extracción estadística más simple, formar variables a partir de los elementos presentes en los mensajes (estadística intra-muestral). La tercera vía también es fruto de la estadística, aunque relacionada con la comparativa entre otro conjunto de datos (estadística inter-muestral). La última es la simple obtención de variables mediante ‘word embedding’ de los textos generados por los usuarios.

6.1.5.1. Dimensiones relacionadas con los enlaces de los mensajes

La primera línea de acción trata los enlaces, *links*, presentes en los mensajes como simples cadenas de texto. Para ello, y antes de poder tratarlos, es necesario extraerlos de los mensajes que los contienen. La gran variedad de métodos existentes de extracción de enlaces no da resultados del todo adecuados, pueden dejar enlaces incompletos o no detectar algunos. El mejor método ha resultado ser la consulta mediante expresión regular, en este caso se emplea la ayuda de una expresión regular ya definida por la comunidad [69] que contempla casi cualquier tipo de enlace. Se evalúa cada enlace extraído y se extraen las dimensiones de más interés para el caso. Cada dimensión es binaria, un 1 representa la existencia de la característica buscada. La tabla 7 siguiente muestra la lista del nombre de cada característica y una breve descripción de cada una.

Esta línea requiere de un paso previo a las extracciones. La gran mayoría de los enlaces son enlaces acortados, es decir, no se presentan como largas cadenas de texto común como: “https://blog.exploratory.io/sentiment-analysis-with-trump-clinton-sanders-twitter-data-cc978e91960f” sino como cadenas codificadas por servidores externos que ayudan a acortar la dirección como “https://bit.ly/2ANkM1w”. Esta técnica permite la indexación del enlace y por ello la carga computacional es menor, consume menos datos de transmisión, no afecta al SEO y, sobre todo, permiten la incrustación de sistemas de recolección y análisis de datos de su acceso. Twitter suele acortar muchos de los enlaces por defecto por motivos de seguridad

y análisis con el aspecto “t.co”. Incluso muchos de los lugares con renombre y confianza tienen su propio dominio acortado por las mismas razones. Este aspecto es una molestia, no se pueden observar “a simple vista” los componentes del enlace y es necesario consultar la red para poder decodificar y obtener el enlace original. Para ello se detecta primero si el enlace alberga alguno de los dominios de una lista con los servidores de acortamiento de uso más extendido. En caso positivo se intenta, con la ayuda del módulo Python “requests”, la petición y decodificación del enlace. El proceso no es nada simple, muchos de los enlaces se encuentran caídos, son maliciosos, el tiempo de respuesta del servidor es excesivo o, incluso, presentan una ocultación doble, enlaces acortados que se decodifican en otro corto, necesitan de dos o más consultas. La latencia del tratamiento es excesiva, la incrustación de esta tarea al proceso aumenta su tiempo de cálculo a meses, teniendo en cuenta que casi la totalidad de mensajes presentan enlace y se cuenta con casi dos millones y medio de enlaces. Después de semanas de intentos fallidos de conversión, la tarea completa de decodificación se abandona para líneas futuras de mejora. Se tratan pues, los enlaces que se encuentren sin acortar. Otro aspecto a tener en cuenta, solo el 1.7% de los mensajes presenta más de un enlace, por consiguiente se considera solo un enlace por tweet.

Nombre	Descripción
link_campaign	Si contiene el código de seguimiento estadístico en texto “utm”
link_entity	Si contiene alguna de las entidades de una lista predefinida relacionada con el ámbito automovilístico, especialmente modelos y marcas de coche, como: “hilux”, “vivaro” o “sandero”.
link_mediaNews	Si contiene alguna de las entidades de una lista predefinida contenedora de una serie de nombres de medios de comunicación, como: “foxnews”, “spiegel” o “huffingtonpost”.
link_socialMedia	Si contiene alguna de las entidades de una lista predefinida contenedora de una serie de nombres de redes sociales o lugares sociales, como: “instagram”, “reddit” o “foursquare”.
link_photo	Si contiene la cadena "pic.twitter.com/"

Tabla 7. Dimensiones referidas a los enlaces de cada mensaje

6.1.5.2. Dimensiones relacionadas con los mensajes como cadenas de carácteres

La segunda línea de creación de dimensiones es incluso más simple que la anterior. Con los enlaces ya excluidos de los mensajes, se tratan estos como cadenas de caracteres. Se extraen cuatro características relacionadas con la naturaleza estadística de las palabras y dos con elementos propios de Twitter. La siguiente tabla 8 es similar a la anterior. Además de las dimensiones de la tabla, también se implementan las mismas dimensiones, pero no tratando cada mensaje sino el profile o descripción del perfil de usuario, estas son 'prof_long', 'prof_countWords', 'prof_longWords', 'prof_ratioWords'; todas ellas con nombres fácilmente interpretable.

Nombre	Descripción
msg_long	Longitud del mensaje, número de caracteres totales.
msg_countWords	Número de palabras existentes en el mensaje.
msg_longWords	Media del número de caracteres de cada palabra del mensaje. Calculado como “msg_long / msg_longWords”
msg_ratioWords	Relación entre palabras únicas en el mensaje y la cantidad de palabras totales. Los mensajes “Hola Juan” y “El tribunal supremo a vetado la ley más polémica del año pasado” obtendrían ambos ‘1’
msg_retweet	Si el mensaje es producto de un ‘retweet’ o contiene alguno. Consultado con la expresión “^(RT|rt)(@\w*)?[:]”
msg_mention	Si el mensaje contiene menciones. Consultado con la expresión “\s([@#][\w_-]+)”

Tabla 8. Dimensiones referidas al texto del mensaje

6.1.5.3. Dimensiones relacionadas con las palabras usadas por cada clase

La tercera línea de actuación se encarga de la búsqueda de características más enfocada al objetivo principal. Se comparan los diccionarios de la misma longitud, en este caso las 75 palabras con mayor frecuencia, de todos los mensajes de cada usuario con los diccionarios propios de todos los mensajes de los usuarios categorizados con una clase determinada. Es decir, primero se agrupan todos los mensajes de los usuarios dentro de una misma categoría, 'celebrity' por ejemplo; se extrae el diccionario de ese corpus y se almacena con el resto de diccionarios. Para generar las nuevas dimensiones se genera otro diccionario con el corpus de sus mensajes, y se comprueba la intersección de este diccionario con todos y cada uno de los diccionarios de cada clase antes generados. Por tanto, el rango de las variables está en [0,75] aunque se normaliza entre cero y uno. Las dimensiones obtenidas por esta línea son: 'ratioDicc_celebrity', 'ratioDicc_company', 'ratioDicc_employee', 'ratioDicc_investor', 'ratioDicc_journalist', 'ratioDicc_ngo', 'ratioDicc_professional', 'ratioDicc_public_institutions', 'ratioDicc_spammer', 'ratioDicc_sportsmen' 'ratioDicc_stockholder', 'ratioDicc_undecidable'. Estas son las doce características que describen lo parecidos que son los diccionarios respecto las otras clases. Se destaca que, como se ha indicado previamente, ahora spammer es una categoría más de clasificación.

6.1.5.4. Dimensiones relacionadas con 'word embedings'

Por último, en la cuarta y última línea se extraen las dimensiones que representan los textos de los mensajes de los usuarios. Mensajes a los que previamente se les ha extraído los enlaces existentes. La extracción de estas dimensiones se gestiona de forma automática, empleando 'word embeddings'. Los sistemas barajados para esta tarea son los siguientes:

El uso de Word2Vec, codifica cada palabra a un espacio matemático de R dimensiones mediante el uso de una veloz red neuronal. Para esta técnica se necesitaría agrupar los resultados de cada palabra para representar un mensaje o tweet, cosa que no da buenos resultados en la literatura en relación con otras técnicas. Es un método potente, captura las vinculaciones internas entre las palabras mediante relaciones espaciales.

En base a Word2Vec existen diferentes variaciones especializadas que reducen pequeñas desventajas de casos específicos. Es el caso de 'GloVe' Sense2Vec' y Tweet2Vec'. GloVe, a diferencia de Doc2Vec, no es un modelo predictivo sino basado en recuento. Es decir, no se enfoca en predecir una o varias palabras sino de construir la representación a partir de una matriz de palabra-frecuencia y la reducción de esta. Presenta pues ventajas de velocidad, paralelización y tolerancia a grandes conjuntos de datos. Sense2Vec, en cambio, aporta capacidades de desambiguación entre palabras o incluso ironía; por desgracia es un modelo que requiere meta-información adicional como etiquetas de la función gramatical por cada palabra y al ser relativamente reciente se hayan muy pocos ejemplos en la literatura.

Doc2Vec es una variación del anterior modelo que codifica de forma íntegra cada mensaje al espacio matemático. De ámbito general, de uso extendido y popularidad por su facilidad de adaptación e integración en las plataformas más famosas de tratamiento de datos. 'Tweet2Vec' es otra de las variantes especializadas, en este caso de Doc2Vec. Este curioso método se encarga de codificar los tweets, aunque con adaptaciones en su estructura interna para ser robusta a palabras o caracteres especiales no vistos en el conjunto de test.

De entre todos los sistemas destacados, el sistema elegido entre todos es 'Doc2Vec' por ser el más simple de aplicar al trabajo ya que es de fácil integración con el ecosistema de herramientas elegido, gracias al módulo Python Gensim. También su comunidad y la documentación disponible en la red son factores clave. El uso de sistemas basados en 'word2Vec' necesitarían de una aglomeración final de dimensiones que desvirtuaría la representación más fiel.

Estos sistemas transforman los textos a representaciones en un espacio matemático de dimensiones elegidas previamente. Los datos se presentan por mensaje y no por usuario pero se necesita una representación que englobe todos los mensajes por cada usuario. Por ese motivo se contemplan dos posibilidades:

- La concatenación de todos los mensajes de cada usuario en un solo texto de gran tamaño para su transformación posterior

- La transformación de cada mensaje de cada usuario y la posterior aplicación de técnicas de agregación de cada dimensión por cada mensaje representado.

Se emplea la segunda opción ya que el uso de '*word embedings*' en textos de gran tamaño no suele favorecer las pequeñas particularidades que pueden ser decisivas en la clasificación y que sí se reflejan mejor en los textos de menor tamaño.

El número de dimensiones elegidas para la representación de los textos es 75. Así pues, por cada mensaje de cada usuario se obtiene un vector de 75 elementos que se agregan mediante el mínimo, el máximo, la media y la mediana. De este proceso resultan cuatro vectores de 75 elementos, 300 en total, que representan de formas distintas el conjunto de mensajes por cada usuario. Se usa también 'word embeding' en el texto de la descripción del perfil de usuario (profile). Al final del proceso se obtienen en total 375 dimensiones.

Las cuatro líneas de acción expuestas extraen la mayoría de dimensiones del conjunto de datos final. A pesar de ello aun se consideran posibles dimensiones que aún pueden aportar valor al proceso de aprendizaje e inferencia. Por ejemplo, la dimensión 'language' de cada usuario es la moda de todos los lenguajes de cada mensaje, indica el lenguaje preferido por el usuario. 'Pepeated_tweets' que, como su nombre indica, indica el número de mensajes idénticos que tiene cada usuario, aunque parezca poco común en las exploraciones previas se ha observado todo lo contrario. La dimensión 'isTest' es la encargada de mantener los conjuntos de train y 'test' separados de la misma forma que se obtuvieron, esta dimensión va a permitir generar varios hilos de aprendizaje con varias particiones en los conjuntos, una por defecto y otra por elección propia. Los nombres de los usuarios, por su unicidad, han sido usados como clave primara en todo el proceso de construcción. Al finalizar el proceso de extracción y ensamblaje se obtiene un conjunto de datos completo con 416 dimensiones y 5416 muestras. El número de muestras es producto de las 1313 muestras del conjunto de 'train', 2493 del conjunto de 'test' y las 1611 muestras del pequeño conjunto de usuarios 'spammer'. En la tabla 9 siguiente se muestra un resumen de las 416 dimensiones extraídas en todo este proceso.

name	mean47	max24	min1	min53	median30	prof2	prof54
count	mean48	max25	min2	min54	median31	prof3	prof55
class	mean49	max26	min3	min55	median32	prof4	prof56
influencer	mean50	max27	min4	min56	median33	prof5	prof57
languaje	mean51	max28	min5	min57	median34	prof6	prof58

Repeated Tweets	mean52	max29	min6	min58	median35	prof7	prof59
mean1	mean53	max30	min7	min59	median36	prof8	prof60
mean2	mean54	max31	min8	min60	median37	prof9	prof61
mean3	mean55	max32	min9	min61	median38	prof10	prof62
mean4	mean56	max33	min10	min62	median39	prof11	prof63
mean5	mean57	max34	min11	min63	median40	prof12	prof64
mean6	mean58	max35	min12	min64	median41	prof13	prof65
mean7	mean59	max36	min13	min65	median42	prof14	prof66
mean8	mean60	max37	min14	min66	median43	prof15	prof67
mean9	mean61	max38	min15	min67	median44	prof16	prof68
mean10	mean62	max39	min16	min68	median45	prof17	prof69
mean11	mean63	max40	min17	min69	median46	prof18	prof70
mean12	mean64	max41	min18	min70	median47	prof19	prof71
mean13	mean65	max42	min19	min71	median48	prof20	prof72
mean14	mean66	max43	min20	min72	median49	prof21	prof73
mean15	mean67	max44	min21	min73	median50	prof22	prof74
mean16	mean68	max45	min22	min74	median51	prof23	prof75
mean17	mean69	max46	min23	min75	median52	prof24	favourites_count
mean18	mean70	max47	min24	median1	median53	prof25	followers_count
mean19	mean71	max48	min25	median2	median54	prof26	friends_count
mean20	mean72	max49	min26	median3	median55	prof27	statuses_count
mean21	mean73	max50	min27	median4	median56	prof28	onDomain_tweets
mean22	mean74	max51	min28	median5	median57	prof29	link_campaign
mean23	mean75	max52	min29	median6	median58	prof30	link_entity
mean24	max1	max53	min30	median7	median59	prof31	link_mediaNews
mean25	max2	max54	min31	median8	median60	prof32	link_socialMedia
mean26	max3	max55	min32	median9	median61	prof33	link_photo
mean27	max4	max56	min33	median10	median62	prof34	msg_long
mean28	max5	max57	min34	median11	median63	prof35	msg_countWords
mean29	max6	max58	min35	median12	median64	prof36	msg_longWords
mean30	max7	max59	min36	median13	median65	prof37	msg_ratioWords
mean31	max8	max60	min37	median14	median66	prof38	msg_retweet
mean32	max9	max61	min38	median15	median67	prof39	msg_mention
mean33	max10	max62	min39	median16	median68	prof40	ratioDicc celebrity
mean34	max11	max63	min40	median17	median69	prof41	ratioDicc company
mean35	max12	max64	min41	median18	median70	prof42	ratioDicc employee
mean36	max13	max65	min42	median19	median71	prof43	ratioDicc investor
mean37	max14	max66	min43	median20	median72	prof44	ratioDicc journalist
mean38	max15	max67	min44	median21	median73	prof45	ratioDicc ngo
mean39	max16	max68	min45	median22	median74	prof46	ratioDicc professional
mean40	max17	max69	min46	median23	median75	prof47	ratioDicc public_institutions

mean41	max18	max70	min47	median24	Prof long	prof48	ratioDicc spammer
mean42	max19	max71	min48	median25	Prof countWords	prof49	ratioDicc sportsmen
mean43	max20	max72	min49	median26	Prof longWords	prof50	ratioDicc stockholder
mean44	max21	max73	min50	median27	Prof ratioWords	prof51	ratioDicc undecidable
mean45	max22	max74	min51	median28	profile	prof52	ratioDicc influencer
mean46	max23	max75	min52	median29	prof1	prof53	isTest

Tabla 9. Resumen de las dimensiones de los datos finales

6.2. Aprendizaje e inferencia

Ahora, con un conjunto de datos que posiblemente represente más fielmente las muestras que los conjuntos de datos originales, se trabaja la tarea de desarrollar el módulo de aprendizaje e inferencia de los datos preparados con el fin de cumplir con el segundo objetivo principal establecido, mejorar los resultados obtenidos por los trabajos anteriores. Se repasa pues las dos dimensiones objetivo:

- 'influencer' de contenido binario dicotómica [1, 0] representa si el usuario representa un perfil con influencia sobre otros usuarios o no.

- 'class', dimensión no dicotómica nominal, una multiclase categórico que contiene los perfiles de usuarios preestablecidos por las bases de los objetivos de RepLab 2014. Como se ha señalado, la nueva clase considerada en este trabajo, 'spammer', es considerada como otra clase más a predecir.

6.2.1. Preparación de los datos

A pesar de que se consideran ambas clases compatibles ente sí ya que, por ejemplo, no es nada raro ver una 'celebrity' 'influencer', se consideran todos aquellos usuarios con perfil 'spammer' como no 'influencer' para mantener. Por definición un usuario 'spammer' genera contenido malicioso o de nula utilidad, lo que contradice la definición de influencer. Por tanto, los 1611 usuarios 'spammer' tienen la dimensión 'influencer' en 0.

Para adecuar los datos a los sistemas de aprendizaje se tratan los valores nulos de los ejemplos. La mayoría se corresponden a los usuarios de RepLab2014 que no coinciden con el conjunto de SLOD-BI (60% del conjunto). Estos valores nulos se sustituyen por -1, ya que el 0 en este caso representa la ausencia del recuento de alguna de las métricas.

Se parte pues de un conjunto de datos de 412 dimensiones útiles. La dimensión 'profile' por ser el texto de la descripción, y estar contenido en su representación numérica, se obvia. El nombre y las clases objetivo tampoco son tenidas en cuenta como dimensiones útiles que puedan aportar al aprendizaje. Se explora la posibilidad de reducción de la dimensionalidad del conjunto de datos ya que podría afectar al rendimiento de los modelos de aprendizaje, y algunas de estas dimensiones podrían estar altamente correlacionadas. Por ello se explora la matriz de correlación donde se observan 314 dimensiones con una correlación superior al 85% y de ellas 195 son superiores al 90%. Son indicios suficientes como para constatar la necesidad de reducir las dimensiones del conjunto.

La cantidad de muestras no es excesiva, pero la posibilidad de valores y combinaciones de las variables es alta. Esto provoca que haya muestras con configuraciones muy variadas y por ende configuraciones únicas. Puede que en la inferencia surjan muestras no vistas en el aprendizaje, pero en los conjuntos de entrenamiento se requiere de más de una muestra con un valor de dimensión objetivo único. Para solucionar este problema se amplían aquellas muestras con dichos valores únicos usando la técnica similar a 'data augmentation' solo que no se modifica ligeramente la muestra, solo se redunda a riesgo de producir sobreajuste en ese valor de la clase.

6.2.2. Diseño del experimento

En los experimentos de aprendizaje automático, los datos se gestionan con tres conjuntos diferentes: *'train', 'test'* y *'validation'.* Los algoritmos "aprenden" de los datos mostrados con el conjunto de *'train',* y evalúan y corrigen su aprendizaje mediante el conjunto de datos *'test',* no presente en el proceso de aprendizaje. Al finalizar el aprendizaje, los algoritmos infieren sobre el conjunto de validación.

La competición RepLab2014 facilita los conjuntos de 'train' y 'test' pero no el de validación. Este último conjunto es utilizado por los responsables del evento para evaluar la efectividad de los trabajos presentados. El conjunto de 'train' representa un 34.5% del total y un 65.5% el de 'test'. El artículo de RepLab2014 [50] expone los resultados finales de cada grupo obtenido por la inferencia en el conjunto de validación. Este trabajo no se enfrentará al conjunto de validación y, por tanto, los resultados finales se obtienen de la inferencia sobre el conjunto de 'test', tratado como el conjunto de validación. Y, por consiguiente, el conjunto de 'train' es dividido en 'train' y 'test'.

Los 1611 usuarios spammers se añaden en el conjunto de 'train' y la proporción entre 'train' y 'test' cambia a 54% y 46% respectivamente. Esta proporción o particionado ya definido se denominará de ahora en adelante 'particionado por defecto'. Se contempla otro tipo de particionado; concatenando los conjuntos y eligiendo, mediante selección de muestras aleatorias, un 66% de muestras para el conjunto de 'train' y 'test' y el 33% restante para el de validación. Este particionado se denominará 'particionado tradicional' y permite la posibilidad de muestras clasificadas como spammers en el conjunto de validación. La ilustración 8 muestra una representación gráfica de lo expuesto. Al haber dos tipos de particionado de los datos, el experimento se bifurca. Al final se compararán los resultados de cada particionado.

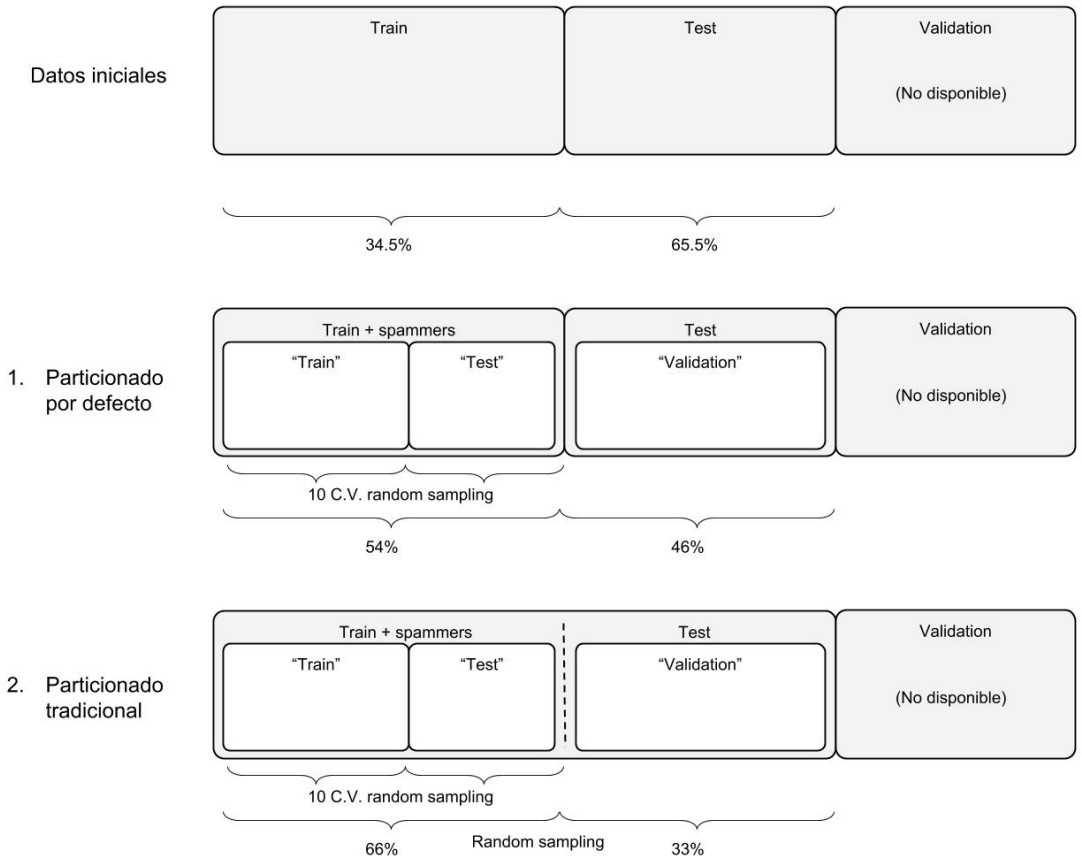


Ilustración 8.Esquema resumen de los estados iniciales del conjunto de datos

Como se ha comentado en el punto anterior, por un exceso de dimensiones y por existir una fuerte correlación entre muchas de ellas se emplea un método de reducción de la dimensionalidad. Se usa pues 'PCA' (Principal Component Analysis). Este método se emplea en aquellos modelos que lo necesiten, las redes neuronales no precisan de forma tan necesaria como el resto de modelos más tradicionales está técnica.

Para el aprendizaje se deben tener algunos aspectos importantes en cuenta. Por ejemplo, la selección de las muestras, en este caso se elige un muestreo aleatorio y estratificado, de ese modo se consigue aliviar en menor medida el desbalance de clases y en mayor se consigue una generalización. Además, y a costa de potencia de cálculo, se emplea validación cruzada de 10 particiones que ayudará en el proceso de generalización.

Los modelos elegidos, como marca el tercer objetivo de este trabajo, son: Máquina de soporte vectorial (SVM), 'Random Forest' y 'Gaussian Naive Bayes' como principales modelos tradicionales. Como modelo de reciente popularidad se emplea una simple red neuronal totalmente conectada. Los modelos considerados tradicionales deben ajustar su aprendizaje a los resultados del aprendizaje, para ello se usa un proceso automático de búsqueda de tales parámetros, la busqueda en red o 'Gridsearch'. Este proceso ejecuta un el aprendizaje con todas y cada una de las combinaciones entre los distintos parámetros que debiera tomar cada modelo. Por tanto, se añade mucha más carga de cómputo ya que en función del número de las posibilidades, las iteraciones o número de ejecuciones aumentan combinatorialmente. También, al haber dos clases a predecir se debe volver a doblar el proceso de aprendizaje

Un ejemplo claro son las 980 veces que se ejecuta SVM para encontrar los parámetros óptimos, teniendo en cuenta ambas opciones de partición son 1960 y aprendiendo por cada variable objetivo 3920 ejecuciones solamente para probar el modelo SVM. La siguiente tabla 10 muestra los parámetros elegidos de todos los modelos.

La red neuronal no ejecuta ningún sistema de ajuste de parámetros, no lo necesita. Es fácil predecir que esta será más veloz que los métodos tradicionales que obedecen a las iteraciones bajo la búsqueda de los parámetros óptimos. También, se emplean varios métodos diferentes para generalizar mejor y optimizar el proceso. En este caso se introduce en cada capa 'dropout' del 25%, un método de regularización que evita el sobre-ajuste eliminando el contenido de las neuronas al azar, introduciendo ruido; tal 25% es el porcentaje de neuronas aleatorias a eliminar en cada iteración. Además, se emplea un 'scheduler' de tres fases, otro método encargado de aumentar la calidad del aprendizaje mediante la programación a priori de la tasa de aprendizaje, tamaño del salto en el descenso por gradiente o learning rate, para que colapse el un mínimo local de forma controlada y precisa en función de las iteraciones en el aprendizaje; configurar el parámetro '*decay*' de forma dinámica. En este caso los umbrales han sido 240 y 350, 'epoch' donde se observa la estabilización, se dirige el 'learning rate' del 0.01 al 0.001 para finalizar superando los 350 'epoch' con 0.0001.

La configuración para cada clase objetivo es la misma salvo la última capa, el número de posibilidades que contempla cada variable a predecir; la dimensión 'influencer' una y la dimensión 'class' doce neuronas. Los parámetros de este modelo neuronal han sido elegidos también mediante la ejecución y observación a priori, pero de forma manual. Se observa que el número de 'epoch' es crítico en el resultado, pero no la topología o el optimizador.

Modelo	Parámetro	Valores
PCA	número de dimensiones	50, 150, 250, 350, 400
SVM	C	0.0001, 0.001, 0.01, 1, 10, 100, 1000
	tipo de Kernel	linear', 'poly', 'rbf', 'sigmoid'
	gamma	0.0001, 0.001, 0.01, 1, 10, 100, 1000
Random forest	profundidad de cada árbol	5, 15, 25, 35
	número de árboles	50, 100, 200, 300
Naive Bayes	-	-
Red neuronal	número de capas	5 ocultas
	número de epoch	750
	neuronas por capa	(412), 300, 200, 100, 50, 15, (12/1)
	activación por capa	Relu, Relu, Relu, Relu, Relu, SoftMax
	optimizador	Adam

Tabla 10. Parámetros seleccionados en cada modelo

En principio, el flujo puede parecer bastante complejo ya que existen dos versiones del aprendizaje en función de la partición. En cada partición se aprende e infiere el conjunto de validación las dos dimensiones a predecir. En cada dimensión a predecir se usan cuatro modelos de aprendizaje. En tres de los cuatro modelos se ejecutan múltiples veces y con diferentes parámetros para optimizar el aprendizaje. De esos tres, en cada ejecución se emplea validación cruzada sobre diez particiones. Para resumir parte de este flujo la siguiente ilustración 9 muestra un diagrama que resume de forma simplificada y de forma visual este proceso.

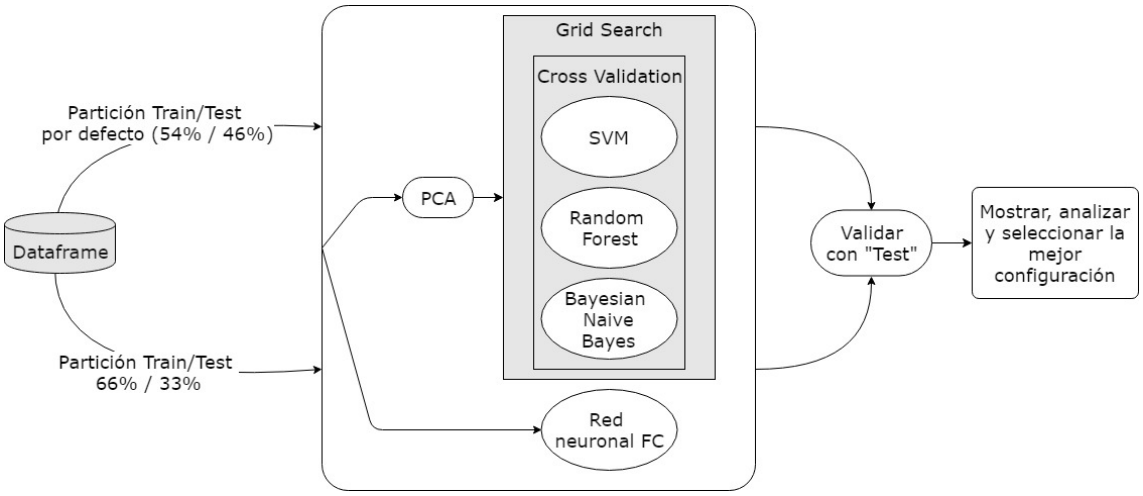


Ilustración 9. Diagrama del flujo del aprendizaje de una dimensión

El proceso de aprendizaje es bastante costoso computacionalmente. El tiempo que se debe emplear en el entrenamiento es bastante alto. Para la reducción del tiempo o se aligera carga en el proceso o se aumenta la potencia de cómputo. El proceso debe ser el indicado y por tanto se debe ejecutar en una máquina con una potencia de cómputo alta. Por ese motivo, el código desarrollado se porta a una máquina del departamento con 32 núcleos y 64Gb de RAM. El código exportado se paraleliza para explotar el beneficio multi-núcleo de la máquina. A pesar de estas medidas el tiempo de entrenamiento e inferencia no desciende del orden de días.

7. Discusión de los resultados

7.1. Resultados obtenidos

Finalmente se extraen los resultados reflejados en las métricas de la predicción. En este punto se analizan los resultados y se comprueba si se ha logrado el objetivo de superar los resultados de los trabajos referencia anteriores. Por la naturaleza del diseño del experimento no es conveniente poner la totalidad de los resultados por la multitud de ejecuciones en el proceso. Quizás, los mejores resultados de cada modelo como comparación entre ellos, aunque el objetivo no es aprendizaje y comparación entre los modelos sino la comparación entre los mejores resultados de los trabajos señalados. Por eso, los resultados se muestran por métrica y no por modelo, es decir, que configuración de qué modelo genera el mejor valor para una métrica en particular. En la siguiente tabla 11 se muestran los resultados obtenidos en este trabajo en ambos particionados distintos. Con el particionado por defecto se aprenden muestras que no existen en la validación, los 'spammers', y se infieren muestras no vistas en el conjunto de entrenamiento. En el particionado tradicional [66/33] se muestrea de forma aleatoria, y junto con la técnica de 'cross-validation', las probabilidades de que ocurra lo mismo son menores. Se esperan mejores resultados en el particionado tradicional, aunque también una estimación del error mayor por inferir en un conjunto de validación menor en comparación con el particionado por defecto.

Partición tradicional	**Identificación de 'Influencers'**			
	Métrica	**Valor**	**Modelo**	**Parámetros del modelo**
	Accuracy	0,8841, std:0,008235	SVC	gamma:100, kernel:polinómico, C:0.01, DimensionesPCA: 50
	Precision	0,8158, std:0,09474	SVC	gamma:0.01, kernel:sigmoide, C:10, DimensionesPCA: 50
	Recall	0,8216, std:0,03085	Gaussian NB	DimPCA:250
	F1	0,5429, std:0,02932	SVC	gamma:1000, kernel:polinomial, C:0.001, DimensionesPCA: 250
	ROC_AUC	0,9175, std:0,01166	Random Forest	Profundidad_arbol: 5, Bootstrap, criterio: gini, n_arboles: 300, DimPCA:50
	Clasificación multi-etiqueta			
	Accuracy	0,7613, std:0,007947	SVC	gamma:100, kernel:rbf, C:100, DimensionesPCA: 250
Partición por defecto	**Identificación de 'Influencers'**			
	Métrica	**Valor**	**Modelo**	**Parámetros del modelo**
	Accuracy	0,8882, std:0,00415	SVC	gamma:100, kernel:polinómico, C:0.01, DimensionesPCA: 250
	Precision	0,7251, std:0,03571	Random Forest	Profundidad_arbol: 5, Bootstrap, criterio: gini, n_arboles: 300, DimPCA:50
	Recall	0,7954, std:0,03306	Gaussian NB	DimPCA:250
	F1	0,5739, std:0,03966	SVC	gamma:100, kernel:polinomial, C:1000, DimensionesPCA: 50
	ROC_AUC	0,9241, std:0,00675	Random Forest	Profundidad_arbol: 15, Bootstrap, criterio: gini, n_arboles: 300, DimPCA:50
	Clasificación multi-etiqueta			
	Accuracy	0,7742, std:0,00692	SVC	gamma:100, kernel:rbf, C:100, DimensionesPCA: 250

Tabla 11. Configuraciones de los mejores resultados finales con ambos particionados

Los valores de esta tabla otorgan una idea general de como ha funcionado el sistema. La tabla está dividida por el tipo de particionado y a su vez por el tipo de tarea. Dentro de cada tarea están las mejores métricas junto con la configuración del modelo responsable de estas. A pesar de no poder interpretar el conjunto de métricas como un mismo resultado por tratarse de configuraciones diferentes, todas ellas dan una idea general de varios aspectos importantes. Es necesario recordar que estos valores no son proporcionados por el conjunto de 'test' sino por el de validación y, por ejemplo, la variable precisión (accuracy) adquiere mayor relevancia que la obtenida por el conjunto de test.

Se analizan ahora los resultados de la tarea de identificación de usuarios 'influencer'. Como se ha indicado antes, los valores de las métricas responden a la configuración del algoritmo de aprendizaje que produce el mejor resultado. Por eso no se pueden comparar los particionados observando el conjunto de las métricas. Sí se pueden analizar los datos en su generalidad, por ejemplo, los resultados obtenidos por ambos particionados son similares a pesar de la variabilidad de los modelos. Como se ha predicho, las desviaciones estándar del error de casi todas las métricas del particionado por defecto son menores al particionado tradicional por haber inferido en un conjunto de validación mayor, aunque la diferencia es poca.

La máquina de soporte vectorial es el algoritmo que aporta mejores resultados en el particionado tradicional, cuatro de seis configuraciones ganadoras. Es posible que, bajo estas características, el algoritmo SVM sea el más viable para este tipo de problema, aunque en el particionado por defecto 'Random forest' también genera buenos resultados. Es notable el 88% de acierto, o error de acierto del 12%, en la predicción de clases en ambos tipos de particionado. En general los resultados son aceptables tratándose de métricas extraídas del conjunto de validación, aunque, de algún modo, podrían ser mejorables.

En el caso de la clasificación de las múltiples categorías, en las que, se recuerda se ha añadido la etiqueta 'spammer'. Solamente se observa, tanto en este trabajo como en los trabajos referencia, la métrica precisión (accuracy). Por suerte, la configuración del algoritmo de aprendizaje que genera el mejor valor de esta métrica es la misma y eso permite la comparación directa entre particionados.

La diferencia entre las precisiones de ambos particionados es de 0,0129 a favor de la partición por defecto, también la desviación estándar del error es ligeramente inferior, ambos valores decantan este método de particionado el más adecuado en este problema. Un 77% de acierto en la predicción es buen valor teniendo en cuenta el número de clases y su problema con el desbalance.

La matriz de confusión, mostrada en la tabla 12, no muestra un rendimiento bueno, el desbalance puede ser el responsable de que las clases minoritarias no se infieran y recaigan sobre las mayoritarias. La clase 'undecidable' es la más confusa, quizás por su naturaleza indeterminada o por ser la clase con más muestras. Se recuerda la particular metodología de "Twitter sólo texto", el principal documento referencia, que intentaba deshacerse de este problema con un flujo de acción particular, detectando previamente los 'influencers' para detectar posteriormente las clases más frecuentes: 'professional' y 'journalist'.

spammer	**0**	0	0	0	0	0	0	0	0	0	0	0
undecidable	11	**673**	76	45	5	1	0	0	0	0	0	0
professional	8	419	**188**	84	7	5	0	2	0	0	0	0
journalist	11	219	96	**165**	6	2	0	3	0	0	0	0
ngo	1	97	14	39	**3**	0	0	0	0	0	0	0
company	1	23	32	12	0	**1**	0	0	0	0	0	0
celebrity	0	115	7	4	1	0	**0**	0	0	0	0	0
public institution	0	32	8	18	0	0	0	**0**	0	0	0	0
sportmen	1	43	3	4	2	0	0	0	**0**	0	0	0
investor	0	18	12	6	0	0	0	0	0	**0**	0	0
employee	0	6	24	6	0	6	0	0	0	0	**0**	0
stockholder	0	0	0	0	0	0	0	0	0	0	0	**0**

Tabla 12. Matriz confusión de la mejor clasificación obtenida

En la tabla 12 no se ve por ninguna parte el modelo de red neuronal por no haber conseguido buenos resultados, ni tan siquiera próximos a considerarse aceptables. Los resultados se muestran en la tabla 13.

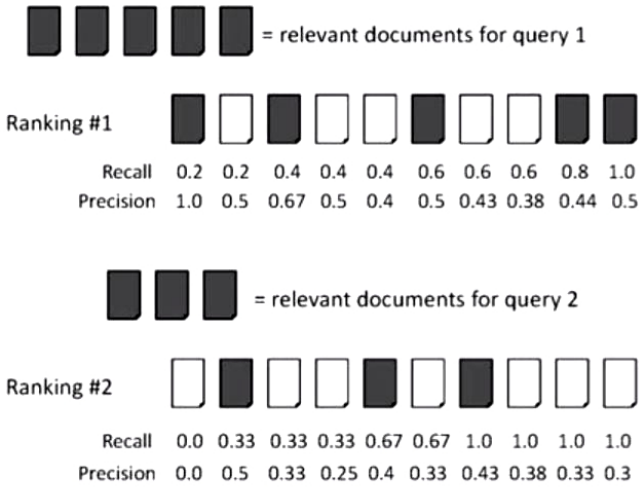
El bajo rendimiento del modelo de red neuronal ha refutado la equivocada expectativa indicada en el punto “Objetivos”. Es posible que la razón de este rendimiento se deba a la extrema sensibilidad de este tipo de modelos al desbalance de las clases como expone [72]. Una solución posible es crear muestras sintéticas o replicar muestras de clases minoritarias (‘Data Augmentation’).A pesar de todo, la red neuronal es el modelo que emplea menos tiempo de proceso a diferencia de los modelos tradicionales, esto puede deberse por parte a tratarse de una red neuronal con una configuración bastante simple o por no usar ‘Scikit Learn’ como módulo principal sino ‘Keras’, que paraleliza la tarea en GPU y vuelve la tarea más eficiente.

Tipo de partición	**Tradicional**	**Defecto**
Detección de 'influencer'	0,2176	0,3017
Clasificación múltiples etiquetas	0,5732	0,3513

Tabla 13. La precisión de la red neuronal

7.2. Comparación con los trabajos previos

Por otra parte, todos los trabajos comparten una métrica resultado común, el *M.A.P.* o 'mean average precision' en la tarea referida a los 'influencers', gracias a ello la comparación entre diferentes trabajos es simple y rápida. 'Mean average precision', es una métrica poco intuitiva y usada sobre todo en detección de objetos en imágenes o sistemas de recomendación. Se define como la media del promedio de la precisión de cada documento en cada consulta o inferencia, una forma de resumir con un solo valor los valores de 'recall' y 'precision'. Dicho de otro modo, es la media de cada 'average precision' en cada inferencia ejecutada en forma de consulta. En la generación del 'average precision' importa el orden de "llegada" de cada muestra y su resultado de inferencia. Normalmente, cada inferencia o consulta debe mostrar un ranquin o lista ordenada de muestras/documentos en función de un parámetro como por ejemplo su nivel de confianza en la predicción. La ilustración 10 expone un pequeño ejemplo bastante explicativo de la métrica. MAP es una métrica perfecta para la tarea de crear un ranquin de calidad en la clasificación de usuarios, se prefiere a otras métricas por su capacidad de síntesis y su adecuación para este tipo de tareas.



$$average\ precision\ query\ 1 = (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$$
$$average\ precision\ query\ 2 = (0.5 + 0.4 + 0.43)/3 = 0.44$$
$$mean\ average\ precision = (0.62 + 0.44)/2 = 0.53$$

Ilustración 10. Ejemplo explicativo de la métrica MAP (Mean Average Precission)

Aunque, en el estudio de las métricas y teniendo en cuenta que solamente se considera relevante una sola consulta; es el caso de la discriminación entre 'influencers' o no, sería más interesante considerar el uso de 'average precision' como métrica de la creación del ranquin. De todos modos,

A diferencia de los trabajos referencia previos, y como parte de la definición de este trabajo en los puntos iniciales, la tarea relacionado con los 'influencers' se ha limitado sólo a su identificación y no a la creación de un ranquin. Por ello, en la comparación de los resultados se emplean las variables mostradas en la ilustración 6 del punto "Desarrollo". A pesar de no

compartir las mismas métricas entre este y el resto de trabajos, es posible analizar y comparar fácilmente el rendimiento.

El trabajo “Twitter sólo texto” es el que mejores resultados obtiene de entre todos los trabajos referencia y por ello se enfrentan sus resultados con los de este trabajo. “Twitter sólo texto” presenta las mismas métricas que sus compañeros, MAP y precisión aunque este último lo desglosa en P-macro y P-micro para observar la eficiencia en clases de tamaños grandes y pequeños.

Como aclaración previa necesaria, LM es un modelo de distribución de probabilidad sobre secuencias de palabras en donde se representa cada documento d como una probabilidad de palabras $P(w_i|d)$, en ese trabajo se aplica de tal forma que cada clase c_j presenta una probabilidad determinada de generar ciertas palabras dadas $P(w_i|c_j)$. Por todo ello, se puede determinar la probabilidad de un documento o conjunto de ellos mediante si se factoriza como $P(c_j|d)=\sum P(c_j|w_i)* P(w_i|d)$.

Otra aclaración previa necesaria, LDSE o “Low Dimensionality Statistical Embedding” es un modelo de representación de documentos mediante distribución de probabilidad de las ocurrencias de las palabras en diferentes clases. Dicho de otro modo, este método averigua la probabilidad o peso de un término a pertenecer a una clase determinada. Así pues, cada documento es una distribución de pesos que se compara con la distribución de pesos de los documentos de una clase determinada. Al final del proceso de ‘codificación’ del documento se obtiene un vector, cada documento es representado por seis variables numéricas para, más en adelante, aprender e inferir con los modelos de aprendizaje automático:

- la media de los pesos de los términos del documento
- la desviación estándar de los pesos del documento
- el peso de menor valor del documento
- la media de los pesos por documento
- ratio de los términos únicos y el número total de términos usados

Primero se analizan ahora los resultados de la tarea de identificación y generación de un ranquin de los usuarios supuestamente ‘influencers’. Esta tarea no comparte las mismas métricas porque, como se ha comentado previamente, el objetivo de este trabajo era la identificación y no la generación del ranking. A pesar de ello, y gracias a que las métricas MAP y accuracy guardan cierta similitud es posible comparar en grandes rasgos los resultados. En “Twitter sólo texto” se alcanza un MAP del 0.874% en el ámbito automovilístico con el método LDSE mientras que los resultados conseguidos en este trabajo han sido del 0,888% de acierto o precisión con un error estándar 0,00415 en el particionado por defecto. La diferencia no es grande, aunque la comparación no debe contemplarse de forma exacta; también porque la metodología o flujo de acción de este es bastante diferente a este trabajo.

Se observa ahora la tarea de clasificación de perfiles en múltiples etiquetas. Los resultados del trabajo “Twitter sólo texto” son: P-micro con 0,699 usando un modelo de ‘language modeling’ LM y P-micro con 0,243 con un modelo similar al usado por este trabajo, *Support vector machines* y *Doc2Vec*. En este trabajo, escogiendo la mejor métrica de ambos tipos de particionado, consigue una precisión del 0,7742 % con una desviación estándar del error de 0,00692. Ambos resultados superan los trabajos del documento RepLab 2014 con diferencia, pero existe una diferencia de precisión del 0.071% a favor de este mismo trabajo. Se ha conseguido superar aunque por poco.

Los resultados han sido mejorados gracias al enriquecimiento de los datos, ya sea por adicción de nueva información o extracción de dimensiones. La mejora no es sustancial pero sí significativa. Los resultados finales no son fácilmente comparables por la derivación de los objetivos en otros más simples y la diferencia de metodologías entre trabajos, aunque si se aprecia fácilmente el rendimiento de cada tarea por individual.

7.3. Comprobación utilidad de las métricas de SLOD-BI

Para comprobar si las métricas aportadas por SLOD-BI han sido relevantes en la mejoría de los resultados finales respecto los otros trabajos, se aplica el mismo modelo de aprendizaje sobre el conjunto de muestras que sí tienen métricas. Este conjunto de 2171 muestras representa un 40% del total, 5424 muestras totales. Las muestras del conjunto presentan desbalance ya que el conjunto está compuesto por 1611 usuarios 'spammer', siendo obligatoriamente 'no influencers', y 560 usuarios con el resto de categorías. Este desbalance puede dificultar la tarea de comprobar lo útiles que han sido estas dimensiones. Otra limitación presente es el particionado por defecto, ya que, de las 560 muestras con métricas de este conjunto de 2171 muestras, solo 360 muestras están en el conjunto de 'test' por defecto, tan solo 16%. Así pues se ejecuta el experimento con el particionado tradicional (66% - 33%). Al ser menos muestras, la precisión es más sensible al fallo. La siguiente tabla 14 muestra los resultados en el mismo formato que sus compañeras. Se aprecia claramente una disminución de la precisión en la clasificación multi-etiqueta, los parámetros más adecuados del modelo han resultado los mismos que en el experimento global de este trabajo mostrado en la tabla 11. En los resultados de la identificación de 'influencers', a diferencia del experimento principal, se aprecian notables diferencias. El modelo 'random forest' gana protagonismo como algoritmo de aprendizaje automático que proporciona mejores resultados. Se aprecia también la diferencia, no solo en la métrica 'ROC_AUC' que gana unas centésimas sino en la precisión que aumenta en un 2,83%. La clasificación multi-etiqueta también se ve mejorada en un 10.644% respecto al experimento principal.

	Identificación de 'Influencers'			
Partición por defecto	**Métrica**	**Valor**	**Modelo**	**Parámetros del modelo**
	Accuracy	0,9124, std:0,01166	Random Forest	Profundidad_arbol: 15, Bootstrap, criterio: gini, n_arboles: 100, DimPCA:50
	Precision	0,7574, std:0,0606	Random Forest	Profundidad_arbol: 5, Bootstrap, criterio: gini, n_arboles: 100, DimPCA:250
	Recall	0,6683, std:0,04197	Gaussian NB	DimPCA:50
	F1	0,5849, std:0,0700	Random Forest	Profundidad_arbol: 15, Bootstrap, criterio: gini, n_arboles: 100, DimPCA:250
	ROC_AUC	0,9562, std:0,0077	Random Forest	Profundidad_arbol: 15, Bootstrap, criterio: gini, n_arboles: 300, DimPCA:50
	Clasificación multi-etiqueta			
	Accuracy	0,88064, std:0,006749	Random Forest	Profundidad_arbol: 15, Bootstrap, criterio: gini, n_arboles: 300, DimPCA:50

Tabla 14. Resultados comprobar la utilidad del uso de métricas aportadas por SLOD-BI

A primera vista parece que la aplicación de las dimensiones aportadas por SLOD-BI sí son útiles, ya que los resultados mejoran cuando se emplea solamente muestras que tengan las 5 dimensiones. Pero, como se han indicado previamente, el conjunto solo representa un 40% del total y en este conjunto existe desbalance. Estas características obligan a generar un

nuevo experimento en el que el conjunto tenga muestras con y sin estas 5 dimensiones manteniendo el número total de muestras. De este modo se comprueba, en un conjunto con menor desbalance, si la mejoría respecto el experimento principal no se ha producido por la simple reducción de muestras. Por el bajo número de muestras es posible que ciertas clases no aparezcan en los dos conjuntos de 'train' y 'test' a la vez. Los resultados de esta segunda comprobación se muestran en la tabla 15, similar a la 14.

	Identificación de 'Influencers'			
Partición por defecto	**Métrica**	**Valor**	**Modelo**	**Parámetros del modelo**
	Accuracy	0,8594, std:0,0093	Random Forest	Profundidad_arbol: 15, Bootstrap, criterio: gini, n_arboles: 100, DimPCA:50
	Precision	0,7492, std:0,0357	Random Forest	Profundidad_arbol: 15, Bootstrap, criterio: gini, n_arboles: 300, DimPCA:250
	Recall	0,6578, std:0,0405	Gaussian NB	DimPCA:250
	F1	0,5357, std:0,0357	Random Forest	Profundidad_arbol: 15, Bootstrap, criterio: gini, n_arboles: 100, DimPCA:250
	ROC_AUC	0,8984, std:0,0135	Random Forest	Profundidad_arbol: 15, Bootstrap, criterio: gini, n_arboles: 300, DimPCA:50
	Clasificación multi-etiqueta			
	Accuracy	0,7070 std:0,01155	Random Forest	Profundidad_arbol: 15, Bootstrap, criterio: gini, n_arboles: 300, DimPCA:50

Tabla 15.Resultados corroborar la utilidad del uso de métricas aportadas por SLOD-BI

Como se observa a simple vista, los modelos con los mejores resultados han sido los mismos que los de la tabla 14, muchos con la misma configuración. Por ese motivo, en comparación directa, los resultados han sido de menor calidad que los del experimento anterior. Se descarta que la mejoría del experimento anterior fuera por la reducción del número de muestras y, por tanto, se corrobora la hipótesis; el uso de los datos aportados por SLOD-BI sí son de utilidad para el aprendizaje del sistema.

8. Mejoras y trabajo futuro

Los resultados de este trabajo han resultado aceptables, aunque también mejorables. Los factores que han influido en esta calidad posiblemente hayan sido:

- El desconocimiento a priori y la falta de experiencia no solo con las tecnologías usadas y también las metodologías empleadas. Se ha empleado tiempo en aprender y entender cada uno de los módulos y las técnicas con uso potencial para el trabajo.

- El tiempo empleado en creación del conjunto de datos final. Comprensión del problema, análisis de posibles soluciones, experimentación de nuevas formas de extraer información. El 80% del tiempo y esfuerzo empleado ha sido invertido en esta parte.

- El exceso de carga computacional. No solo en la parte de aprendizaje sino también en la extracción de características originales del texto. Procesar cadenas de caracteres no es un proceso ágil en comparación con otros formatos de variables y eso ralentiza el proceso de experimentación.

A pesar de todo ello, estos factores son necesarios e inevitables. Gracias a haber tenido en cuenta y superado estos aspectos el trabajo ha podido gestionarse de forma adecuada. Por otro lado, existen otra serie de aspectos mejorables en el caso de repetir este trabajo o continuar con su mejora y desarrollo. Aspectos que, de haber tenido conocimientos previos o una mejor fuente de información, hubieran sido de gran ayuda en la calidad de los resultados, hubieran añadido funcionalidad o hubieran facilitado la continuación del desarrollo o el despliegue. Estos se han señalado a lo largo del trabajo.

- La primera de estas tareas se introduce en el sub-apartado “Exploración y peculiaridades”, en ella se explican dos observaciones que hacen de este un trabajo poco sencillo. La primera es la asunción de la diferencia de calidad entre los conjuntos de RepLab 2014 y SLOD-BI, este último alberga usuarios con perfiles de comportamiento y características anómalas y el primero ha sido un conjunto filtrado por expertos para proporcionar una base limpia en el concurso. La segunda observación fundamental es la escasa intersección entre ambos conjuntos, este hecho a generado el problema de no haber podido aprovechar por completo los datos aportados por SLOD-BI y no aportar uno de los factores diferenciadores a los trabajos anteriores. En el sub-apartado se analiza las diferencias y similitudes entre los conjuntos ‘intersección de los conjuntos principales’ y ‘diferencia de este anterior y el conjunto SLOD-BI’ para poder observar aquellos perfiles similares a RepLab. Sí se observan patrones diferenciadores y por ello se concluye que una mejor selección de los perfiles del gran conjunto de SLOD-BI, teniendo en cuenta las diferencias observadas en las dimensiones ‘followers_count’, Friends_count’ y status_count’, sería beneficiosa para el aprendizaje con el conjunto de datos final. La tarea pendiente es la selección e incorporación de perfiles de calidad mediante la observación de las métricas coincidentes con RepLab 2014.

- Otra de las tareas es comentada en el sub-apartado de “Creación del ‘dataframe’ principal”. En esta parte del trabajo se descodifican los enlaces acortados presentes

en los mensajes para poder extraer características de estos como: si contienen entidades relacionadas con el dominio automovilístico o si contienen la cadena “utm” como parte de una campaña de análisis. Casi un 3% del total de los enlaces son enlaces cortos, este es una cantidad considerable teniendo en cuenta que el total de enlaces responde al orden de millones. La identificación y extracción de los enlaces del cuerpo de los mensajes se ha llevado a cabo con éxito. Pero las consultas y descodificaciones se han aplazado para una futura continuación del trabajo. El tiempo invertido por enlace es excesivamente largo, muchos lugares ya no existen, los servidores no siempre responden o la latencia de respuesta es demasiado larga, hay algunos con direccionamientos a lugares maliciosos o incluso dobles direccionamientos con enlace acortado. Todos estos problemas han consumido una parte del tiempo de desarrollo importante y todos los intentos por completar la tarea han resultado en fracaso. Completar la descodificación de todos los enlaces acortados supondría enriquecer el conjunto de datos y seguramente una significativa mejora en la calidad de los resultados. También, otra mejor línea de acción para el sistema sería tratar cada perfil de usuario o incluso cada mensaje de forma independiente, como un flujo de datos directo o ‘streaming’ para que el proceso de codificación y extracción de dimensiones no sea una tarea abrumadora. Esta consideración, a su vez, lleva a la última tarea.

- Este trabajo alberga una finalidad no comentada anteriormente. Es probable, que pueda ser de gran ayuda al desarrollo de un sistema que necesite de esta funcionalidad de clasificación de usuarios. Uno de los requisitos demandados en caso de convertirse en un módulo útil para el sistema sería preparar el código generado para que actúe como una ‘caja negra’. La capacidad de introducir todos los datos relacionados con un usuario usados en este trabajo, como: los mensajes, sus métricas, su descripción de usuario… y que la salida del módulo sea la clasificación de ambas dimensiones objetivo junto con sus porcentajes de confianza.

9. Referencias

[1] Mendenhall, Thomas Corwin. "The characteristic curves of composition." Science (1887): 237-249

[2] Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., & Inches, G. (2013). Overview of the author profiling task at PAN 2013. In CLEF Conference on Multilingual and Multimodal Information Access Evaluation (pp. 352-365). CELCT.

[3] Sebastiani, F. (2005). Text categorization. In Encyclopedia of Database Technologies and Applications (pp. 683-687). IGI Global.

[4] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, *34*(1), 1-47

[5] Salvador Oliván, José A.: Recuperación de Información. Buenos Aires : Alfagrama, 2008

[6] Chian, Y. Natural Language Processing (NLP)

[7] Pardo, F. M. R. (2016). Author Profiling en Social Media: Identificación de Edad, Sexo y Variedad del Lenguaje (Doctoral dissertation)

[8] Pak, A., & Paroubek, P. (2010, May). Twitter as a corpus for sentiment analysis and opinion mining. In LREc (Vol. 10, No. 2010, pp. 1320-1326)

[9] Sun, S., Luo, C., & Chen, J. (2017). A review of natural language processing techniques for opinion mining systems. Information Fusion, 36, 10-25.

[10] Weiss, S. (2005). Text Mining. Predictive Methods for Analyzing Unstructured information. (en inglés). EUA: Springer. p. 6.

[11] https://www.businessinsider.in/Furious-customers-are-deleting-the-Uber-app-after-drivers-went-to-JFK-airport-during-a-protest-and-strike/articleshow/56859124.cms {Visitado el 26 de agosto de 2018}

[12] Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., ... & Ramavajjala, V. (2016, August). Smart reply: Automated response suggestion for email. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 955-964). ACM.

[13] Jindal, N., & Liu, B. (2007, May). Review spam detection. In Proceedings of the 16th international conference on World Wide Web (pp. 1189-1190). ACM.

[14] Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. Science, 349(6245), 261-266.

[15] Scott, S., & Matwin, S. (1999, June). Feature engineering for text classification. In ICML (Vol. 99, pp. 379-388).

[16] Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. Journal of machine learning research, 3(Mar), 1289-1305.

[17] Y. Bengio; A. Courville; P. Vincent (2013). "Representation Learning: A Review and New Perspectives". IEEE Trans. PAMI, special issue Learning Deep Architectures. 35: 1798–1828

[18] Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[19] Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015, June). From word embeddings to document distances. In International Conference on Machine Learning (pp. 957-966).]

[20] Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. Behavior research methods, instruments, & computers, 28(2), 203-208.

[21] Rohde, D. L., Gonnerman, L. M., & Plaut, D. C. (2006). An improved model of semantic similarity based on lexical co-occurrence. Communications of the ACM, 8(627-633), 116.

[22] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American society for information science, 41(6), 391-407.

[23] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

[24] Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[25] Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. Journal of machine learning research, 3(Feb), 1137-1155.

[26] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013

[27] Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In International Conference on Machine Learning (pp. 1188-1196).

[28] Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[29] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. Journal of machine Learning research, 3(Jan), 993-1022.

[30] Trask, A., Michalak, P., & Liu, J. (2015). sense2vec-A fast and accurate method for word sense disambiguation in neural word embeddings. arXiv preprint arXiv:1511.06388.

[31] Dhingra, B., Zhou, Z., Fitzpatrick, D., Muehl, M., & Cohen, W. W. (2016). Tweet2vec: Character-based distributed representations for social media. arXiv preprint arXiv:1605.03481.

[32] Viera, Angel Freddy Godoy. (2017). Técnicas de aprendizaje de máquina utilizadas para la minería de texto. Investigación bibliotecológica, 31(71), 103-126.

[33] Joachims, T. (1998, April). Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning (pp. 137-142). Springer, Berlin, Heidelberg.

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

[35] Argamon, S., Koppel, M., Fine, J., Shimoni, A.R.: Gender, genre, and writing style in formal written texts. TEXT 23, 321–346 (2003)

[36] Holmes, J., Meyerhoff, M.: The Handbook of Language and Gender. Blackwell Handbooks in Linguistics, Wiley (2003)

[37] Burger, J.D., Henderson, J., Kim, G., Zarrella, G.: Discriminating gender on twitter. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1301–1309. EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011)

[38] Gung ¨ or Tunga. Part of speech tagging. In: Indur- ¨ khya N, Damerau FJ, eds. Handbook of Natural Language Processing. 2nd ed. Boca Raton, FL: Chapman and Hall/CRC Press; 2010, 205–236

[39] Zhang, C., Zhang, P.: Predicting gender from blog posts. Tech. rep., Technical Report. University of Massachusetts Amherst, USA (2010)

[40] Nguyen, D., Gravel, R., Trieschnigg, D., Meder, T.: "how old do you think i am?"; a study of language and age in twitter. Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media (2013)

[41] Zhang, Y., Marshall, I., & Wallace, B. C. (2016, November). Rationale-augmented convolutional neural networks for text classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing (Vol. 2016, p. 795). NIH Public Access.

[42] Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., Inches, G.: Overview of the Author Profiling Task at PAN 2013—Notebook for PAN at CLEF 2013. In: Forner et al.

[43] Rangel, F., Rosso, P., Chugur, I., Potthast, M., Trenkmann, M., Stein, B., Verhoeven, B., Daelemans, W.: Overview of the 2nd author profiling task at pan 2014. In: In: Cappellato L., Ferro N., Halvey M., Kraaij W. (Eds.) CLEF 2014 Labs and Workshops, Notebook Papers. CEUR-WS.org, vol. 1180 (2014)

[44] https://mott.marketing/origen-historia-e-informacion-completa-sobre-la-red-social-twitter/ *{Visitado el 26 de agosto de 2018}*

[45] http://www.cad.com.mx/historia_de_twitter.htm *{Visitado el 26 de agosto de 2018}*

[46]https://www.lainformacion.com/mundo/twitter-para-enfrentarse-a-los-desastres-naturales_YKtsDDrQWQaN3x37y8zSq6/ *{Visitado el 26 de agosto de 2018}*

[47] Kwak, H., Lee, C., Park, H., & Moon, S. (2010, April). What is Twitter, a social network or a news media?. In Proceedings of the 19th international conference on World wide web (pp. 591-600). AcM.

[48] https://blog.hubspot.com/marketing/twitter-usage-stats *{Visitado el 26 de agosto de 2018}*

[49] Makice, K. (2009). Twitter API: Up and running: Learn how to build applications with the Twitter API. " O'Reilly Media, Inc.".

[50] Amigó, E., Carrillo-de-Albornoz, J., Chugur, I., Corujo, A., Gonzalo, J., Meij, E., ... & Spina, D. (2014, September). Overview of replab 2014: author profiling and reputation dimensions for online reputation management. In International Conference of the Cross-Language Evaluation Forum for European Languages (pp. 307-322). Springer, Cham.

[51]http://limosine-project.eu/ *{Visitado el 9 de septiembre de 2018}*

[52]http://www.barriblog.com/2017/10/lo-siempre-quiso-saber-del-api-twitter-nunca-se-atrevio-preguntar-actualizado-2017/ *{Visitado el 26 de agosto de 2018}*

[53] Rafael Berlanga Llavori, Lisette García-Moya, Victoria Nebot, María José Aramburu, Ismael Sanz, Dolores María Llidó: SLOD-BI: An Open Data Infrastructure for Enabling Social Business Intelligence. IJDWM 1-28 (2015).

[54] Reidenbach, R. E., & Robin, D. P. (1990). Toward the development of a multidimensional scale for improving evaluations of business ethics. Journal of business ethics, 9(8), 639-653.

[55] Nebot, V., Rangel, F., Berlanga, R., & Rosso, P. (2018, June). Identifying and Classifying Influencers in Twitter only with Textual Information. In International Conference on Applications of Natural Language to Information Systems (pp. 28-39). Springer, Cham.

[56] Aleahmad, A., Karisani, P., Rahgozar, M., & Oroumchian, F. (2016). OLFinder: Finding opinion leaders in online social networks. Journal of Information Science, 42(5), 659-674.

[57] Vilares, D., Hermo, M., Alonso, M. A., Gómez-Rodríguez, C., & Vilares, J. (2014). LyS at CLEF RepLab 2014: Creating the State of the Art in Author Influence Ranking and Reputation Classification on Twitter. In CLEF (Working Notes) (pp. 1468-1478).

[58] Cossu, J. V., Dugué, N., & Labatut, V. (2015, September). Detecting real-world influence through Twitter. In Network Intelligence Conference (ENIC), 2015 Second European (pp. 83-90). IEEE.

[59] https://docs.python.org/3/tutorial/index.html *{Visitado el 26 de agosto de 2018}*

[60] https://jupyter-notebook.readthedocs.io/en/stable/notebook.html *{Visitado el 26 de agosto de 2018}*

[61] http://pandas.pydata.org/pandas-docs/stable/ *{Visitado el 26 de agosto de 2018}*

[62] https://keras.io/ *{Visitado el 26 de agosto de 2018}*

[63] https://radimrehurek.com/gensim/intro.html *{Visitado el 26 de agosto de 2018}*

[64] https://www.nltk.org/ *{Visitado el 26 de agosto de 2018}*

[65] http://scikit-learn.org/stable/ *{Visitado el 26 de agosto de 2018}*

[66] http://www.numpy.org/ *{Visitado el 26 de agosto de 2018}*

[67] https://matplotlib.org/ *{Visitado el 26 de agosto de 2018}*

[68] Nebot, V., Rangel, F., Berlanga, R., & Rosso, P. (2018, June). Identifying and Classifying Influencers in Twitter only with Textual Information. In International Conference on Applications of Natural Language to Information Systems (pp. 28-39). Springer, Cham.

[69] https://daringfireball.net/2010/07/improved_regex_for_matching_urls *{Visitado el 26 de agosto de 2018}*

[70] https://es.wikipedia.org/wiki/Twitter *{Visitado el 26 de agosto de 2018}*

[71] Rangel, F., Rosso, P., Potthast, M., Stein, B., & Daelemans, W. (2015, September). Overview of the 3rd Author Profiling Task at PAN 2015. In CLEF (p. 2)

[72] Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks, 106, 249-259.