Clasificación automática de correos electrónicos

Autor:
Juan Manuel Fernández

Dirección:
Dr. Marcelo Errecalde
(Director)
Mg. Mario Oloriz
(Co-director)

Tesis presentada para obtener el grado de
Magister en Inteligencia de datos orientada a Big Data

Facultad de Informática
Universidad Nacional de La Plata
La Plata, Argentina

Agosto 2022

Clasificación automática de correos electrónicos © Agosto 2022

Autor:
Lic. Juan Manuel Fernández

Dirección:
Dr. Marcelo Errecalde (Director)
Mg. Mario Oloriz (Co-director)

Universidad:
Universidad Nacional de La Plata, Buenos Aires, Argentina

AGRADECIMIENTOS

Honestamente siento que son muchas las personas que me acompañan de forma diaria e incondicional en mi vida, lo cual es motivo de una gratitud infinita de mi parte.

En primer lugar, quiero agradecer a Marcelo, mi Director de Tesis, por acompañanarme y enseñarme en este proceso de profundo aprendizaje. Siento que gracias a sus interminables conocimientos en la disciplina y gran generosidad he podido integrar los conocimientos que me brindó la Universidad Nacional de La Plata a lo largo de toda la Maestría, a la vez que he crecido como investigador y docente.

En este mismo sentido, tengo que agradecer a Mario, mi co-director de Tesis y fundamentalmente mi guía profesional, amigo y compañero de mil aventuras en nuestra Universidad, la Universidad Nacional de Luján. He elegido, y sigo eligiendo, dejar mi granito de arena en esta Institución, a la que conocí acompañando a mi padre cuando era un niño.

Desde lo personal, Gael es y será el motivo y estímulo de todas las acciones que salen de mi y tienen valor. Tenerlo es una bendición y una motivación constante por superarme y ser su ejemplo.

También quiero agradecer a mis padres especialmente, si no fuera por ellos no hubiera llegado aquí, estoy profundamente orgulloso de la familia que me tocó; mis hermanas y sus familias a su vez también son parte de esto.

Además quiero agradecer a mis amigos y compañeros de trabajo, a los que siempre estuvieron y siempre están.

Sin dudas que otra parte importante en este logro -de las más importantes-, son Cari, Coqui y Angi; una familia que elegí, que hoy me hace sentir pleno, y que junto a Gael endulzan cada uno de mis días.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE FÓRMULAS

RESUMEN

En la actualidad se generan millones de datos cada día y su aprovechamiento e interpretación se han vuelto fundamentales en todos los ámbitos. Sin embargo, la mayor parte de esta información posee un formato textual, sin la estructura ni la organización de las bases de datos tradicionales, lo cual representa un enorme desafío para su procesamiento mediante técnicas de aprendizaje automático. Otro de los desafíos inherentes al procesamiento masivo de datos comprende el etiquetado de los mismos, actividad necesaria para las técnicas de aprendizaje supervisado donde la estrategia tradicional consiste en el etiquetado manual.

Por su parte, el correo electrónico es una de las herramientas de comunicación asincrónica más extendida en la actualidad, habiendo desplazado a los canales más clásicos de comunicación debido a su alta eficiencia, costo extremadamente bajo y compatibilidad con muchos tipos diferentes de información. Existen trabajos que han recogido estimaciones respecto de la utilización mundial de este medio de comunicación tomando como referencia al Grupo Radicati, quienes afirman que actualmente existen más de 3930 millones de usuarios y se proyectan 4371 millones para el año 2023, alcanzando el tráfico actual de 293.6 billones de correos enviados diariamente. Muchos de estos correos electrónicos son enviados a centros de contacto de organizaciones públicas y privadas debido a que este medio se ha constituido en un canal de comunicación estándar. Sin embargo, éste es un canal que requiere una importante afectación de recursos humanos.

Con el fin de mejorar su uso y aprovechar a los correos electrónicos como fuente de conocimiento se han aplicado diversas técnicas de minería de datos a este tipo de información, entendiendo a la minería de datos como una etapa del proceso de descubrimiento de conocimiento que consiste en aplicar algoritmos de análisis y explotación de datos para producir una enumeración particular de patrones (o modelos) sobre los datos.

A su vez, el correo electrónico como fuente de datos posee un conjunto de características particulares respecto de otras fuentes de datos que hace que existan diferencias y problemáticas particulares entre la minería de textos tradicional y la minería de correos electrónicos, conocida como *email mining*.

En este contexto, se ha aplicado *email mining* con diferentes objetivos como la detección de correo electrónico no deseado, la categorización de correo electrónico, el análisis de contactos, de propiedades de red de correo electrónico y visualización.

En este trabajo, en primer lugar se intenta dimensionar la cantidad de conocimiento que supone el intercambio de correos diariamente a nivel mundial, así como entender su evolución y características técnicas. A continuación, se realiza un estudio del estado del arte de la disciplina, partiendo del proceso de descubrimiento de conocimiento y caracterizando el proceso de construcción de un clasificador automático de correos electrónicos.

Luego, quizás como principal contribución de esta investigación, se propone una nueva estrategia de etiquetado semi-supervisado híbrido con tres variantes. Se parte de de una base inicial con correos etiquetados de forma tradicional y se realiza una extracción de las características principales para cada clase, utilizando tres técnicas como la regresión logística, TF-IDF y SS3. Luego, con la base de conocimiento completa indexada en un motor de búsqueda de propósito general como *Elasticsearch*, se recuperan documentos de cada clase en función de las características detectadas por cada técnica y se construye un clasificador, el cual se evalúa en función de un conjunto de datos de prueba diferente del utilizado para el proceso anterior.

En términos del desarrollo experimental, se trabaja a partir de un caso de estudio basado en correos electrónicos en idioma español propiedad de la Universidad Nacional de Luján. Esta Universidad cuenta con un sistema informático propio para llevar adelante la gestión académica de las actividades inherentes a la enseñanza de grado y pregrado, así como los trámites que de éstas se desprenden. Este sistema de gestión cuenta con una interfaz web a la que acceden los estudiantes para realizar todos los trámites relacionados a su vinculación con la Institución. A su vez, posee una funcionalidad para realizar consultas vía correo electrónico al staff administrativo.

El sistema, ante la formulación de una consulta por parte de los estudiantes envía, mediante un servidor SMTP, la consulta a una dirección de correo electrónico especialmente destinada para este fin. Al cuerpo de ese correo, además del texto escrito por el estudiante, se agregan datos académicos y de la persona tales como nombre y apellido, legajo, documento, Carrera, teléfono y email personal.

Utilizando una porción de esa base de conocimiento, en este trabajo se aborda el desafío de generar un modelo, en el marco de la disciplina de aprendizaje automático para clasificar cual es el tema de cada consulta realizada en función del contenido de los mensajes enviados.

A su vez, se realizan experimentaciones en términos del proceso de clasificación semi-supervisada propuesto. A partir de este proceso, se demuestra que, para los datos utilizados, estas técnicas de extracción de características, utilizadas como estrategias de etiquetado para la clasificación semi-supervisada, mejoran la capacidad de los clasificadores cuando se incorporan las instancias etiquetadas automáticamente a las etiquetadas de forma manual para entrenar el modelo.

Por último, se reformula esta estrategia para ser utilizada como una estrategia de balanceo para el aprendizaje automático desde conjuntos de datos desbalanceados. Nuevamente, se demuestra que la estrategia sigue siendo competitiva, al menos para este conjunto de datos, en relación a algunas de las técnicas de remuestreo más utilizadas de la actualidad, tanto de *oversampling* como de *undersampling*.

ABSTRACT

Millions of data are generated every day and their use and interpretation have become essential in all fields. However, most of this information is in textual format, without the structure and organization of traditional databases, which represents an enormous challenge for its processing by machine learning techniques. Another challenge inherent to massive data processing involves labeling the data, a necessary activity for supervised learning techniques where the traditional strategy consists of manual labeling.

E-mail is one of the most widespread asynchronous communication tools today, having displaced the more traditional communication channels due to its high efficiency, extremely low cost and compatibility with many different types of information. Some studies have compiled estimates regarding the worldwide use of this means of communication, taking as a reference the Radicati Group, which states that there are currently more than 3930 million users and 4371 million are projected for the year 2023, reaching the current traffic of 293.6 billion e-mails sent daily. Many of these emails are sent to contact centers of public and private organizations because this medium has become a standard communication channel. However, this is a channel that requires a significant allocation of human resources.

In order to improve its use and take advantage of e-mails as a source of knowledge, several data mining techniques have been applied to this type of information, understanding data mining as a stage of the knowledge discovery process that consists of applying data analysis and exploitation algorithms to produce a particular enumeration of patterns (or models) on the data.

In turn, e-mail as a data source has a set of particular characteristics with respect to other data sources, which leads to particular differences and problems between traditional text mining and what is known as e-mail mining.

In this context, email mining has been applied with different objectives such as spam detection, email categorization, contact analysis, email network properties and visualization.

In this work, first of all, the amount of knowledge involved in the daily exchange of emails worldwide is measured, as well as its evolution and technical characteristics are analyzed. Then, a study of the state of the art of the discipline is carried out, starting from the process of knowledge discovery and characterizing the process of building an automatic email classifier.

Then, perhaps as the main contribution of this research, a new hybrid semi-supervised labeling strategy with three variants is proposed. It starts from an initial base with traditionally labeled mails and performs an extraction of the main features for each class, using three techniques such as logistic regression, TF-IDF and SS3. Then, with the complete knowledge base indexed in a general purpose search engine such as *Elasticsearch*, documents of each class are retrieved based on the features detected by each technique

and a classifier is built, which is evaluated based on a different test data set than the one used for the previous process.

In terms of experimental development, we work from a case study based on e-mails in Spanish language owned by the National University of Luján. This University has its own computer system for the academic management of activities inherent to undergraduate and graduate education. This management system has a web interface for students to carry out all the procedures related to their relationship with the Institution. At the same time, it has a functionality to make inquiries via e-mail to the administrative staff.

When students ask a question, the system sends, through an SMTP server, the query to an e-mail address specially designed for this purpose. In the body of the e-mail, in addition to the text written by the student, academic and personal data are added, such as name and surname, academic record, document, career, telephone and personal e-mail.

Using a portion of this knowledge base, this work addresses the challenge of generating a model, within the framework of the machine learning discipline, to classify the subject of each query based on the content of the messages sent.

In turn, experiments are conducted in terms of the proposed semi-supervised classification process. From this process, it is shown that, for the data used, these feature extraction techniques, used as labeling strategies for semi-supervised classification, improve the capacity of the classifiers when automatically labeled instances are incorporated into the manually labeled ones to train the model.

Finally, this strategy is reformulated to be used as a strategy for machine learning from unbalanced datasets. Again, it is shown that the strategy remains competitive, at least for this data set, in relation to some of the most widely used resampling techniques today, both *oversampling* and *undersampling*.

PUBLICACIONES

Esta tesis incluye trabajos del autor que han sido publicados o, a la fecha, han sido presentados para su publicación. En particular, algunos datos, ideas, opiniones y figuras que se presentan aquí han aparecido previamente o pueden aparecer poco después de la presentación de este documento de la siguiente manera:

- Fernandez, J. M. & Errecalde, M. (2022). Instance retrieval from non-labeled data as a strategy for automatic classification of imbalanced e-mail datasets. XXVIII Congreso Argentino de Ciencias de la Computación **(en evaluación)**.
- Fernández, J.M., & Errecalde, M. (2022). Multi-class E-mail Classification with a Semi-Supervised Approach Based on Automatic Feature Selection and Information Retrieval. In: Rucci, E., Naiouf, M., Chichizola, F., De Giusti, L., De Giusti, A. (eds) Cloud Computing, Big Data & Emerging Topics. JCC-BD&ET 2022. Communications in Computer and Information Science, vol 1634, (pp. 75-90). Springer, Cham.
- Fernández, J. M., Cavasin, N., & Errecalde, M. L. (2021). Classic and recent (neural) approaches to automatic text classification. In Short papers of the 9th Conference on Cloud Computing Conference, Big Data & Emerging Topics, pp. 20-24.
- Fernandez, J. M., Cavasín, N., Rodriguez, A., & Errecalde, M. (2021). Clasificación automática de correos electrónicos. In XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021, Chilecito, La Rioja).

A su vez, se listan publicaciones previas en las que participó el tesista que están relacionadas con la temática abordada en esta investigación:

- Banchero, S., Fernández, J. F., Tonin Monzón, F., Giordano, L. A., Marrone, A., Lulic, M., & Tolosa, G. H. (2020). Modelos para aprendizaje automático en tiempo real sobre entornos de big data. In XXII Workshop de Investigadores en Ciencias de la Computación (WICC 2020, El Calafate, Santa Cruz).
- Banchero S.; Fernandez, J.M.; Tonín Monzón F.; Giordano L.; Marrone A.; Paz Soldán, C.; Tolosa G.: "Modelos de aprendizaje automático en tiempo real sobre entornos de Big Data". XXI Workshop de Investigadores en Ciencias de la Computación, Universidad Nacional de San Juan. Abril 2019. ISBN: 978-987-3984-85-3.
- Banchero S.; Tolosa G.; Tonín Monzón F.; Fernandez, J.M.; Paz Soldán, C.; Giordano L.; Marrone A.: "Algoritmos de aprendizaje automático para respuestas en tiempo real sobre entornos masivos de datos". XX Workshop de Investigadores en Ciencias de la Computación, Universidad Nacional del Nordeste, Corrientes. Abril 2018. ISBN: 978-987-3619-27-4.

1 INTRODUCCIÓN

1.1 CONTEXTO

El correo electrónico es una de las herramientas de comunicación asincrónica más extendida en la actualidad, habiendo desplazado a los canales más clásicos de comunicación debido a su alta eficiencia, costo extremadamente bajo y compatibilidad con diferentes tipos de información [91].

Existen trabajos que han recogido estimaciones respecto de la utilización mundial de este medio de comunicación tomando como referencia al Grupo Radicati, quienes afirman que actualmente existen más de 3930 millones de usuarios y se proyectan 4371 millones para el año 2023 [39], alcanzando el tráfico actual de 293.6 billones de correos enviados diariamente [12, 89].

Muchos de estos correos electrónicos son enviados a centros de contacto de organizaciones públicas y privadas debido a que este medio se ha constituido en un canal de comunicación estándar [87]. Sin embargo, éste es un canal que requiere una importante afectación de recursos humanos. A efectos de cuantificar el costo de esta intervención humana, algunos autores han relevado este aspecto a través de estudios de casos; por ejemplo, se demostró que responder un correo electrónico de un ciudadano enviado a la Agencia de Pensiones de Suecia lleva unos 10 minutos y, por lo tanto, los 99000 mensajes que reciben por año pueden necesitar hasta 10 empleados de tiempo completo para responderlos [46].

Con el fin de mejorar su uso y aprovechar a los correos electrónicos como fuente de conocimiento se han aplicado diversas técnicas de minería de datos a este tipo de información [91], entendiendo a la minería de datos como una etapa del proceso de descubrimiento de conocimiento que consiste en aplicar algoritmos de análisis y explotación de datos para producir una enumeración particular de patrones (o modelos) sobre los datos [27]. En este sentido, existe un área particular de la minería de datos, denominada Minería de Textos, donde el conocimiento es generado mediante la utilización de bases de datos exclusivamente textuales como fuentes de datos [95].

Estos sistemas de análisis de texto se enfrentan a problemáticas muy complejas dentro del área de la ciencias de la computación, debido principalmente a la dificultad del análisis del lenguaje (derivada de su ambigüedad) fundamentalmente en la etapa de análisis semántico, como así también, a los relativamente escasos materiales de entrenamiento y a la capacidad de cómputo necesaria para correr determinados algoritmos muy demandantes en recursos de hardware [17].

A su vez, el correo electrónico como fuente de datos posee un conjunto de características particulares respecto de otros elementos de texto que hace que existan diferencias y problemáticas peculiares entre la minería de textos tradicional y la minería de correos electrónicos, conocida como *email mining* [12].

Por un lado, los correos electrónicos poseen información adicional en el encabezado que pueden ser explotados para la obtención de conocimiento. Asimismo, poseen una extensión reducida que hace que muchas técnicas de minería de textos sean ineficientes para estas fuentes de datos. Este tipo de comunicaciones, muchas veces, se da en un contexto informal o inmersos en una cultura organizacional particular, por lo tanto los errores ortográficos y gramaticales, así como los abreviaturas o acrónimos, aparecen con frecuencia.

Por otro lado, además de los datos textuales, los correos electrónicos pueden contener tipos más ricos de datos, como enlaces URL, marcas HTML e imágenes. Aprovechar al máximo esos datos no textuales en los correos electrónicos es un problema interesante y desafiante para abordar [91].

En este contexto, se ha aplicado *email mining* con diferentes objetivos como la detección de correo electrónico no deseado, la categorización de correo electrónico, el análisis de contactos, de propiedades de red de correo electrónico y visualización.

En lo relativo a clasificación de correos electrónicos, existen abordajes desde el procesamiento y generación de resúmenes [94], utilización de redes neuronales [4], clasificación para respuesta automática de correos [86] y aplicación de técnicas basadas en máquinas vector-soporte y Naive Bayes [91], entre otras.

1.2 OBJETIVOS

El objetivo general de este trabajo consiste en estudiar y analizar el conocimiento existente sobre técnicas aprendizaje automático aplicadas a la clasificación automática de textos, particularmente de correos electrónicos, y generar un modelo que aborde un problema concreto. Esto trae aparejados los siguientes objetivos específicos:

- Analizar, describir y sistematizar el estado del arte de la clasificación automática de correos electrónicos.
- Diseñar un proceso general para el tratamiento y clasificación automática de correos electrónicos, intentando categorizar esta problemática dentro de la disciplina general de Minería de Textos, la cual abarca las características y particularidades que se originan en esta forma de comunicación.
- Abordar un estudio experimental a partir del procesamiento de las consultas que los estudiantes de la Universidad Nacional de Luján formulan, mediante correo electrónico, asegurando la calidad de los datos y etiquetando, a partir de especialistas en el dominio, un subconjunto de los mismos con las temáticas a las que corresponden en ese dominio.
- Consolidar, a partir de lo anterior, una base de conocimiento con los correos electrónicos etiquetados, representados a partir de diferentes estrategias de representación

de textos incorporando a los mismos un proceso de curado y generación de atributos estáticos.

- Entrenar un modelo para la clasificación automática de estos correos electrónicos, abordando de manera concreta esta problemática y dimensionando el problema.
- Indagar en el estado del arte de la clasificación semi-supervisada de documentos, proponiendo una estrategia de esta índole para mejorar la performance de clasificación del modelo entrenado.

1.3 EL CASO DE ESTUDIO

La Universidad Nacional de Luján es una universidad nacional, de gestión pública, de la República Argentina, de dimensión mediana (25.600 estudiantes) que presta servicios en seis ciudades de la Provincia de Buenos Aires: Luján, San Miguel, Campana, Chivilcoy, San Fernando y la Ciudad Autónoma de Buenos Aires.

La Universidad cuenta con un sistema informático propio para llevar adelante la gestión académica de las actividades inherentes a la enseñanza de grado y pregrado, así como los trámites que de éstas se desprenden. Este sistema de gestión cuenta con una interfaz web a la que acceden los estudiantes para realizar todos los trámites relacionados a su vinculación con la Institución. A su vez, posee una funcionalidad para realizar consultas vía correo electrónico al staff administrativo.

Figura 1.1: Captura de pantalla de la opción de envío de Consultas del Módulo Web

El sistema, ante la formulación de una consulta por parte de los estudiantes envía, mediante un servidor SMTP, la consulta a una dirección de correo electrónico especialmente destinada para este fin. Al cuerpo de ese correo, además del texto escrito por el estudiante, se agregan datos académicos y de la persona tales como nombre y apellido, legajo, documento, Carrera, teléfono y email personal.

Ante la llegada de un correo electrónico, el personal administrativo de la Institución debe verificar la situación y dar respuesta al estudiante dentro de las 48 horas de realizada la solicitud.

Como política de resguardo de la información, la Universidad Nacional de Luján realiza periódicamente una copia de seguridad con estas consultas y las respuestas brindadas a los estudiantes en cada caso, llegando actualmente a un total almacenado de 24700 correos con consultas y sus respectivas respuestas.

Utilizando esa base de conocimiento, en este trabajo se propone abordar el desafío de generar un modelo, en el marco de la disciplina de minería de textos –más específicamente email mining– para identificar y clasificar el tema de cada consulta realizada en función del contenido de los mensajes recibidos. La implementación de un modelo de estas características permitiría al staff administrativo de la Universidad Nacional de Luján organizar las tareas y remitir los correos a cada sector interviniente de manera automática, considerando que actualmente dos personas están dedicadas de forma casi exclusiva a redirigir las consultas manualmente.

1.4 ORGANIZACIÓN DEL DOCUMENTO

En esta sección se resume el contenido de los restantes capítulos que integran este documento:

Capítulo 2: El correo electrónico. Luego de describir la motivación, introducción y objetivos de este trabajo en el Capítulo actual, en el Capítulo 2 se hace un breve repaso de la historia, el estado actual, las proyecciones de utilización y la estructura –desde un abordaje técnico– del correo electrónico.

Capítulo 3: Marco teórico y revisión bibliográfica. En este capítulo se conceptualiza la disciplina de minería de texto, centrando la atención en la minería de correos electrónicos específicamente. A su vez, se busca categorizar el área de resolución del problema y se determina un proceso secuencial, para tareas de clasificación de correos electrónicos a partir de minería de texto, proponiendo un conjunto de alternativas posibles y técnicas a utilizar.

Capítulo 4: Clasificación semi-supervisada. En esta etapa se define el aprendizaje semi-supervisado, particularmente la clasificación relevando los principales antecedentes y desafíos por resolver. Luego, se avanza en una de las principales contribuciones de este trabajo, la cual consiste en la presentación de una nueva estrategia de clasificación semi-supervisada basada en la recuperación automática de instancias a partir de la combinación de tres técnicas diferentes de selección de características como el vocabulario de SS3, los coeficientes de la regresión logística y la ponderación TFIDF de los términos de la base de correos agrupadas por clase.

Capítulo 5: Trabajos experimentales. Se define el esquema de trabajo y el diseño experimental, el cual es explicado en detalle. Aquí, se hace hincapié no sólo en los modelos

desarrollados sino también en los desafíos y dificultades encontradas en cada etapa del problema, así como también las limitaciones y criterios tomados para alcanzar los objetivos propuestos en este trabajo. Por su parte, se detallan los experimentos llevados a cabo con cada una de las estrategias de representación de documentos, técnicas de aprendizaje automático y hiperparámetros utilizados.

Capítulo 6: Conclusiones y trabajos futuros. El manuscrito finaliza con un apartado de conclusiones finales, principales contribuciones, análisis de resultados y líneas posibles de trabajo futuro a partir de la investigación realizada.

Apéndice A: Aprendizaje automático a partir de datos desbalanceados. En este apéndice, se reformula la estrategia presentada en el Capítulo 4 para presentarla como una estrategia de balanceo para el aprendizaje automático desde conjuntos de datos desbalanceados. Aquí, se demuestra que la estrategia sigue siendo competitiva, al menos para este conjunto de datos, en relación a algunas de las técnicas de remuestreo más utilizadas de la actualidad, tanto de *oversampling* como de *undersampling*.

2

EL CORREO ELECTRÓNICO

2.1 HISTORIA

El correo electrónico, tal como lo conocemos hoy en día, ha seguido un proceso evolutivo constante desde su aparición, en el año 1971. En realidad, se atribuye su aparición a ese año puesto que fue el momento en el cual se realizó la primera comunicación de correo electrónico entre dos computadoras diferentes.

Sin embargo, el correo electrónico como herramienta de comunicación en una misma computadora ya existía una década antes, a partir del software SNDMSG [93]. El software SNDMSG era un programa de correo local entre usuarios que permitía componer, dirigir y enviar un mensaje a los buzones de otros usuarios de una única computadora. No obstante, otros autores atribuyen a MAILBOX, creado en 1965 por el prestigioso Massachusetts Institute of Technology (MIT) ser el primer software para el envío de correos [72].

Fue el Ingeniero Ray Tomlinson quién adaptó el software SNDMSG añadiendo la posibilidad a éste de enviar mensajes entre diferentes usuarios que estuvieran conectados a una red más amplia, pero sin que fueran conocidos, sólo a partir de referenciar una dirección o, lo que se conoce actualmente como, correo electrónico [72]. Aquí es donde cobra relevancia, en términos informáticos, el símbolo @ (arroba) como separador entre el nombre del usuario y del servidor en la dirección del correo electrónico. Según testimonios del propio Ray Tomlinson, escogió el arroba por el simple hecho de utilizar un símbolo que estuviese en todos los teclados pero que no apareciera en los nombres propios de las personas, empresas o de los servidores [93].

Al margen de esta experiencia inicial, resulta claro que el contexto de la época no contribuía a la proliferación de esta herramienta de comunicación, puesto que aún no existía lo que hoy conocemos como "Internet". No fue hasta el año 1969 que se estableció -en Estados Unidos- ARPANET, la primera red sin nodos centrales, de la que formaban parte cuatro universidades estadounidenses: Universidad de California Los Angeles (UCLA), Universidad de California Santa Barbara (UCSB), Universidad de Utah y Stanford Research Institute (SRI). A su vez, la fecha de la primera transmisión en esa red tuvo lugar el 29 de octubre de 1969, entre UCLA y SRI [6].

En lo sucesivo, fueron incorporándose a ARPANET diversas universidades e instituciones y para el año 1971 ya había 15 nodos. En 1973, ARPANET se internacionalizó con la incorporación de la Universidad College of London (Inglaterra) y NORSAR (Noruega).

Sin embargo, existen interpretaciones diversas respecto del nacimiento de Internet, tal cual se conoce hoy. Algunos, prefieren marcar como hito el año 1982, en el momento en que

irrumpe como estándar el protocolo TCP/IP (Transfer Control Procotol/Internet Protocol) [6]. Por su parte, otros consideran que Internet aparece al año siguiente, en el año 1983 cuando el Ministerio de Defensa de Estados Unidos abandona ARPANET para establecer una red independiente bajo su control absoluto (MILNET) y nace, a partir de esta decisión una Internet "abierta", al separarse la parte militar y la civil de la red [58].

Independientemente del momento puntual del surgimiento, resulta evidente que esa Internet incipiente aún no resultaba atractiva para el público en general puesto que las funcionalidades con impacto social eran escasas al momento. Sin embargo, esto se modificaría a partir de la creación del lenguaje de marcado HTML para páginas web por parte de Tim Berners-Lee en octubre de 1990, el cual permitía combinar texto, imágenes y establecer enlaces a otros documentos en la red; y a su vez, Berners-Lee, avanzó en la especificación del protocolo HTTP, e intervino en el desarrollo del primer servidor World Wide Web y el primer programa cliente World Wide Web, cuestiones que resultaron elementos claves para la generalización del acceso por parte de la Sociedad [6].

A partir de ello, el acceso al correo y las aplicaciones para tal fin fueron incorporando funcionalidades, al mismo tiempo que la utilización de esta herramienta siguió sumando adeptos, los cuales se incrementaron de manera exponencial [72]. En términos de la forma de acceso, la aparición de las páginas web como las conocemos hoy a partir de HTML permitió que aparezcan los grandes proveedores de servicios de correos electrónicos como Gmail o Hotmail, dado que hasta el momento las aplicaciones por excelencia para el envío de correo eran los denominados "lectores fuera de línea". Esos lectores sin conexión permitieron a los usuarios de correo electrónico, en los inicios de Internet, almacenar su correo electrónico en sus propias computadoras personales, y luego leerlo y preparar respuestas sin estar realmente conectados a la red.

No obstante, es importante aclarar que, previo a esta explosión de la World Wide Web, el correo electrónico ya había despertado el interés de los usuarios comunes, de hecho, hasta el momento de la desaparición de ARPANET, el 75 % de todo su tráfico era correo electrónico. Aún hoy, y a pesar de todas las posibilidades que ofrece la red mundial, el correo electrónico sigue siendo una de las aplicaciones más importantes de Internet y de las más utilizadas [72].

En el mismo sentido, las cifras y proyecciones muestran que el correo electrónico seguirá siendo una parte central de la vida digital diaria. Sin embargo, es importante marcar que existen variaciones en la forma de acceso, ya que el correo electrónico móvil mundial, a diciembre 2018, representó el 43 por ciento de las aperturas mientras que el correo web el 39 por ciento. En efecto, este comportamiento no es sorprendente dado que el acceso a internet ha experimentado un fuerte cambio hacia los dispositivos móviles en los últimos años [89].

2.2 ESTADO ACTUAL

Como se viene sosteniendo, el correo electrónico fue una de las herramientas más importantes desde el surgimiento de Internet y lo sigue siendo actualmente. Si bien esta afirma-

ción resulta intuitivamente verdadera, es importante indagar en cifras sobre su utilización a efectos de dimensionar el volumen de información -y potencialmente conocimiento- que reside en el intercambio mundial de correos electrónicos.

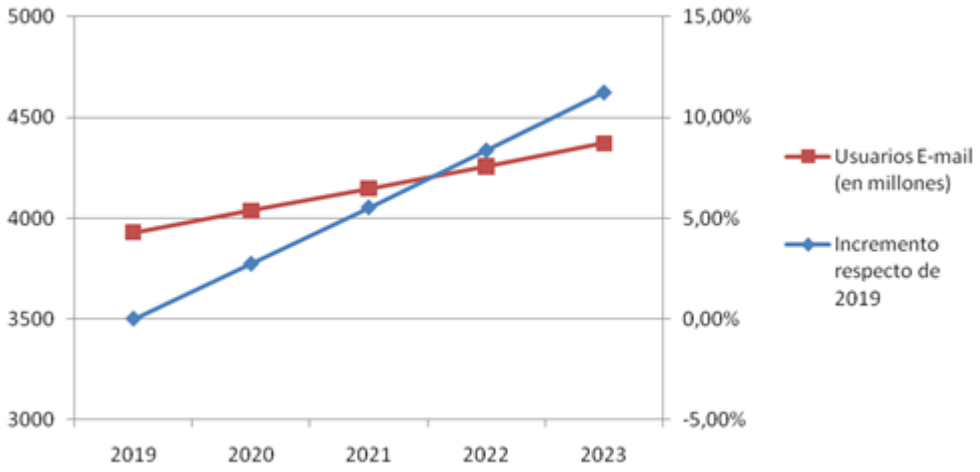

Figura 2.1: Cantidad de usuarios de e-mail y proyección 2019-2023 [39]

.

En primer lugar, como puede observarse en la Figura 2.1, analizando la cantidad de usuarios globales de correo electrónico se vislumbra que el mismo aumentará a 4.400 millones de usuarios en 2023, frente a los 3.930 millones que existen actualmente. A su vez, intentando plantear estas cifras en relación a la población mundial, puede afirmarse que más de la mitad de los habitantes de este planeta utilizan correo electrónico actualmente [39]. Otra cuestión saliente de este gráfico es que el crecimiento anual promedio ronda el 2,5 %, acumulado entre el año actual, 2019, y las proyecciones a 2023, en sólo 4 años, un crecimiento del 11,22 %.

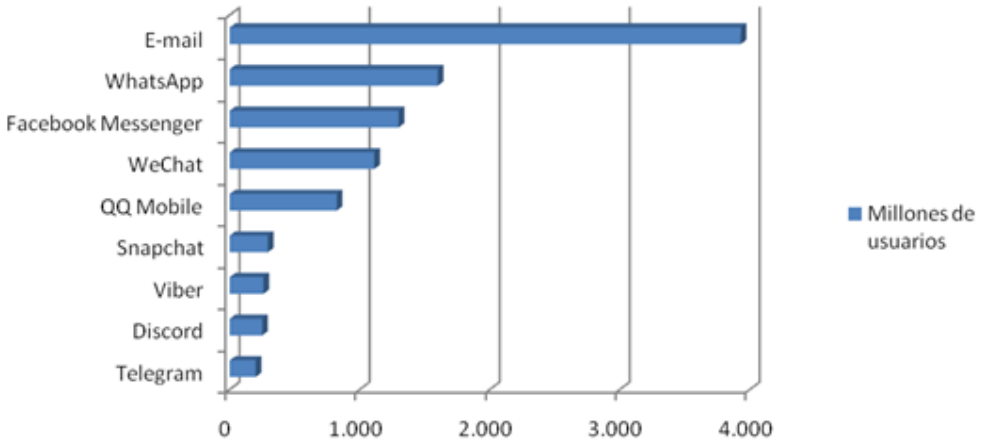

Figura 2.2: Cantidad de usuarios de mensajería móvil y de e-mail 2019 [89]

.

Por otro lado, resulta interesante comparar, en esta nueva era de mensajería móvil, cual es el nivel de utilización del correo electrónico respecto de las aplicaciones para mensajería

móvil más utilizadas actualmente. Al margen de posibles interpretaciones respecto de los criterios de utilización de la población en relación al correo electrónico y la mensajería móvil, la Figura 2.2 resulta esclarecedor para posicionar y dimensionar el uso actual del correo electrónico que, además, lejos de mermar su utilización en el futuro se prevé que siga con una tendencia positiva en los próximos años.

Otro aspecto importante a analizar y que resulta indispensable dimensionar, independientemente de la cantidad de usuarios de correo activos, es la actividad diaria que existe en términos de intercambio de mensajes de correo electrónico mundialmente.

Como se puede apreciar en la Figura 2.3, la cantidad de correos enviados diariamente asciende a los 293 billones, 24 billones más que hace solo dos años y se proyecta que se incremente en casi 54 billones hacia el año 2023, lo cual totaliza un incremento total acumulado en la serie estudiada, que comprende los años 2017-2023, de nada menos que el 29,1 %.

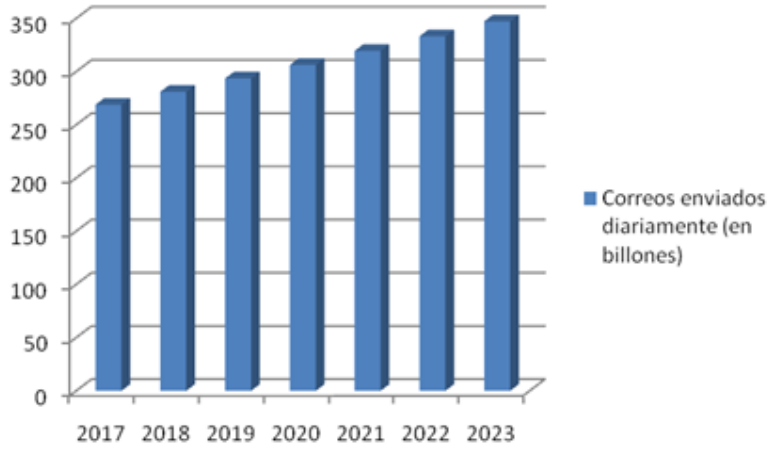

Figura 2.3: Cantidad de e-mails enviados mundialmente por día en 2019. [89]

Para intentar entender el volumen de estas comunicaciones, supongamos por un momento que cada correo electrónico enviado diariamente, en promedio posee un tamaño de 1kb, lo cual sabemos es una suposición conservadora. Si esto fuera así, se necesitarían 273.436,31 terabytes de almacenamiento para persistir las comunicaciones mundiales de un solo día.

Estas cuantificaciones no hacen más que confirmar la relevancia de intentar obtener conocimiento a partir de los correos electrónicos ya que, como se demostró anteriormente, además de acompañarnos hace casi medio siglo, tienen y seguirán teniendo en nuestra vida digital una vital importancia y se constituyen en grandes repositorios potenciales de información y conocimiento.

2.3 ESTRUCTURA DEL CORREO ELECTRÓNICO

Los estándares, protocolos, procedimientos y programas más importantes de Internet son documentados a través de documentos denominados Request for comments (en ade-

lante RFC por sus siglas en inglés) por parte de la Internet Engineering Task Force (IETF) [34].

El primer documento RFC para documentar los mensajes de correo electrónico fue publicado el 21 de noviembre de 1977 con el número 733. No obstante, el estándar para el formato de mensajes de correo electrónico fue evolucionando y actualmente está documentado por el RFC 5322 que a su vez se complementa con el RFC 6854.

En el RFC 5322 se define el estándar con toda la especificación para el envío de mensajes de correos electrónicos entre usuarios de computadoras. Si bien escapa al alcance de este trabajo hacer un análisis exhaustivo del estándar, como cuestión saliente se encuentra en el mismo la estructura general de los correos electrónicos. En líneas generales, se define en ese documento que un correo electrónico consta de un header o encabezado y un body o cuerpo, el cual es opcional.

Respecto del header, posee un conjunto de campos que brindan información. Los más importantes se listan a continuación:

Fecha de envío (orig-date): este campo es obligatorio y representa la fecha de envío del correo electrónico.

Datos del remitente (from, sender, reply-to): estos campos describen a quien origina el mensaje.

Datos del/de los destinatario/s (to, cc, bcc): estos campos describen a quien/quienes recepcionarán el mensaje.

Otros campos (in-reply-to, references, subject, comments, keywords): son campos que brindan información de identificación del mensaje (mensaje-id), del contenido del mensaje que será enviado (subject, comments, keywords) y de identificación de conversaciones o hilos (in-reply-to, references).

En cambio, el body es totalmente desestructurado y está formado por un conjunto de líneas compuestas de caracteres.

A partir de la definición de la estructura de los correos electrónicos puede inferirse que el correo electrónico como fuente de datos posee un conjunto de características particulares respecto de los textos convencionales que hace que existan diferencias y problemáticas peculiares entre la minería de textos tradicional y la minería de correos electrónicos, conocida como email mining [12].

Por un lado, los correos electrónicos poseen información adicional en el encabezado que pueden ser explotados para la obtención de conocimiento. Por otro lado, y aunque el RFC no ahonda en estas características, además de los datos textuales, los correos electrónicos pueden contener tipos más ricos de datos, como enlaces URL, marcas HTML e imágenes. En relación a estas particularidades, aprovechar al máximo estas características distintivas de los correos electrónicos plantea un problema interesante y desafiante para abordar [91].

3

MARCO TEÓRICO Y REVISIÓN BIBLIOGRÁFICA

3.1 CONCEPTOS PRELIMINARES

En la actualidad, y producto de la masificación del acceso internet, se generan millones y millones de datos cada día y su aprovechamiento e interpretación se han vuelto fundamentales en todos los ámbitos.

Muchas áreas del conocimiento se muestran interesadas en extraer conocimiento a partir de la información almacenada, lo cual resulta vital para la toma eficiente de decisiones.

Como desafío adicional al hecho de poder interpretar grandes volúmenes de información, la mayor parte de ella posee un formato textual, sin la estructura ni la organización de las bases de datos tradicionales. Este texto, por sí mismo, no tiene ningún tipo de estándar ni restricción y, por lo tanto, procesarlo se ha vuelto una tarea extremadamente difícil dada la heterogeneidad léxica, sintáctica y semántica.

Este formato resulta algo menos atractivo que otros como el sonido, las imágenes y el video, pero es, sin lugar a duda, el principal medio de comunicación entre seres humanos en la actualidad [81]. Cada correo electrónico enviado, cada búsqueda realizada en Internet y cada publicación subida a la red implica, en mayor o menor medida, datos en formato texto [97].

Como ya se planteó antes, y producto de estas cuestiones, existe una creciente necesidad de desarrollar una nueva generación de teorías computacionales y herramientas que permitan a los humanos extraer información útil (conocimiento) de los volúmenes de datos digitales, los cuales están en constante crecimiento.

De la mano de esta necesidad, aparece a mediados de la década del 90, un campo emergente denominado *descubrimiento de conocimiento en bases de datos* (KDD por su sigla en inglés) [27]. En un nivel abstracto, el KDD se ocupa del desarrollo de métodos y técnicas para dar sentido a los datos a partir de transformar datos de bajo nivel, que generalmente son demasiado voluminosos para comprender e interpretar, en otras formas que podrían ser más compactas (por ejemplo, un informe breve), más abstractas (por ejemplo, una aproximación descriptiva o modelo del proceso que generó los datos), o más útil (por ejemplo, un modelo predictivo para estimar el valor de casos futuros). El núcleo del proceso, aunque el mismo no se agota allí, es la aplicación de métodos específicos de minería de datos para el descubrimiento y extracción de patrones.

Históricamente, la noción de encontrar patrones útiles en los datos ha recibido diversos nombres, incluyendo minería de datos, extracción de conocimiento, descubrimiento de información, recolección de información, arqueología de datos y procesamiento de patrones

de datos. El término minería de datos ha sido utilizado, y lo es actualmente, principalmente por estadísticos, analistas de datos y gestores. También ha ganado popularidad en el campo de las bases de datos.

El término descubrimiento de conocimiento en bases de datos fue acuñado en el primer taller de KDD en 1989 para enfatizar que el conocimiento es el producto final de un descubrimiento basado en datos.

KDD se refiere al proceso general de descubrir conocimiento útil de los datos mientras que la minería de datos se refiere a un paso particular en este proceso que comprende la aplicación de algoritmos específicos para extraer patrones de datos. Los pasos adicionales en el proceso de KDD, como la preparación de datos, la selección de datos, la limpieza de datos, la incorporación de conocimientos previos apropiados y la interpretación adecuada de los resultados de la minería, son esenciales para garantizar que se obtengan conocimientos útiles de los datos [27].

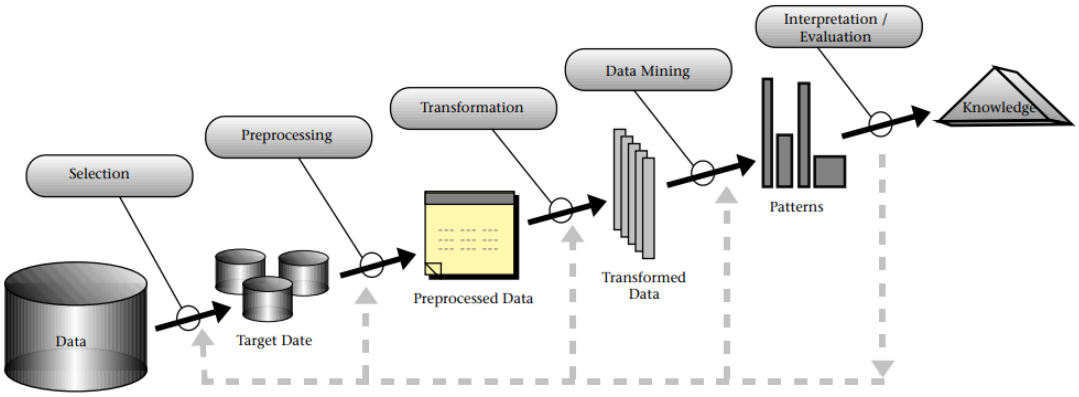

Figura 3.1: Esquema del Proceso de Descubrimiento de Conocimiento [27]

El proceso KDD es interactivo e iterativo, e involucra numerosos pasos con muchas decisiones tomadas por el usuario [13].

Como se planteó antes, el proceso de descubrimiento de conocimiento está estructurado en un conjunto de etapas, las cuales se describen, brevemente, a continuación:

- **Fase 1: Entendimiento del dominio.** El primer desafío del proceso consiste en desarrollar una comprensión del dominio de aplicación y el conocimiento previo relevante, e identificar el objetivo del proceso KDD desde el punto de vista del problema a abordar.
- **Fase 2: Selección de datos.** En segundo lugar, o como segunda fase de este proceso, están las tareas inherentes a crear un conjunto de datos a partir del cual obtener el conocimiento. Esta actividad consiste en seleccionar un conjunto de datos o centrarse en un subconjunto de variables o muestras de datos en el que se realizará el descubrimiento.
- **Fase 3: Preprocesamiento.** La tercera etapa es la de limpieza de los datos y preprocesamiento. Las operaciones clásicas incluyen eliminar el ruido si es apropiado,

recopilar la información necesaria para modelar o dar cuenta del ruido, decidir estrategias para manejar los campos de datos faltantes, tener en cuenta la información de la secuencia de tiempo y los cambios conocidos.

- **Fase 4: Transformación de datos.** La siguiente etapa, la cuarta, es la de reducción y proyección de los datos y consiste en encontrar características para representar los datos dependiendo del objetivo del proceso. Con los métodos de reducción o transformación de dimensionalidad, se puede reducir el número efectivo de variables bajo consideración, o se pueden encontrar nuevas representaciones para los datos.

- **Fase 5: Selección de tarea de minería de datos.** Las etapas cinco, seis y siete están estrechamente relacionadas y están enfocadas en la minería de datos propiamente dicha. La fase cinco tiene por objetivo hacer coincidir los objetivos del proceso de descubrimiento de conocimiento con un método particular de minería de datos como por ejemplo: el resumen, la clasificación, la regresión, la agrupación, etc.

- **Fase 6: Calibración y Selección del modelo.** Esta fase consiste en el análisis exploratorio y la selección de modelos e hipótesis: elegir los algoritmos de minería de datos y seleccionar los métodos que se utilizarán para buscar patrones de datos. Este proceso incluye decidir qué modelos y parámetros podrían ser apropiados y hacer coincidir un método particular de extracción de datos con los criterios generales del proceso KDD.

- **Fase 7: Utilización del modelo.** La séptima etapa es la minería de datos: la búsqueda de patrones de interés en una forma de representación particular o un conjunto de tales representaciones, incluidas reglas de clasificación o árboles, regresión y agrupación.

- **Fase 8: Interpretación del conocimiento.** La anteúltima etapa, la octava, consta de la interpretación de los patrones minados, posiblemente volviendo a cualquiera de los pasos 1 a 7 para una nueva iteración. Este paso también puede implicar la visualización de los patrones y modelos extraídos o la visualización de los datos dados los modelos extraídos.

- **Fase 9: Utilización del conocimiento.** Por último, se culmina con el proceso actuando sobre el conocimiento descubierto: usando el conocimiento directamente, incorporando el conocimiento en otro sistema para acciones adicionales, o simplemente documentándolo y reportándolo a las partes interesadas. Este proceso también incluye verificar y resolver posibles conflictos con conocimiento previo (o extraído).

Como completa Fayyad en *"From data mining to knowledge discovery in databases"*, y este concepto sigue vigente, si bien muchas de las investigaciones se centran en la minería de datos, las demás fases son tan importantes como ésta (y probablemente más) para la implementación exitosa de un proceso de descubrimiento del conocimiento.

El mismo autor plantea, ya a mediados de la década del 90', que los objetivos del proceso de descubrimiento de conocimiento están definidos por el uso que se plantea dar

al mismo. En este sentido, el autor distingue entre dos tipos de objetivos: verificación y descubrimiento. Desde el punto de vista de la verificación, el sistema se limita únicamente a verificar una hipótesis preexistente del usuario. En el caso del descubrimiento, el sistema encuentra de forma autónoma nuevos patrones sin una hipótesis previa. A su vez, es posible subdividir aún más el objetivo de descubrimiento en predicción, donde el sistema encuentra patrones para predecir el comportamiento futuro de algunas entidades, y descripción, donde el sistema encuentra patrones para su presentación a un usuario en una forma comprensible para los humanos [27].

En este sentido, abordando la problemática desde el punto de vista del objetivo del descubrimiento de conocimiento, existe cierto consenso en categorizar en cuatro tipos principales a estas tareas de minería de datos: clasificación, predicción numérica, asociación y agrupamiento [14].

Como caracterización más general, existe una categorización previa que dependerá de la estructura de los datos a analizar. Los datos a partir de los cuales se entrenan los modelos en minería de datos, poseen un conjunto de ejemplos (llamados instancias), cada una de los cuales comprende los valores de una serie de variables, que en la minería de datos a menudo se llaman atributos. Hay dos tipos de conjuntos de datos, que se tratan de formas radicalmente diferentes.

En el primer tipo de conjunto de datos, hay un atributo especialmente designado y el objetivo es utilizar el resto los datos para predecir el valor de ese atributo en instancias que aún no se han visto. Los datos de este tipo se denominan etiquetados y la minería de datos con datos etiquetados se conoce como aprendizaje supervisado. Si el atributo designado es de tipo nominal o categórico la tarea se llama de clasificación. En cambio, si el atributo designado es numérico, consiste en una tarea de regresión.

La clasificación es una de las aplicaciones más comunes para la minería de datos. Corresponde a una tarea que ocurre con frecuencia en la vida cotidiana. Por ejemplo, un hospital puede querer clasificar a los pacientes médicos en aquellos que tienen un riesgo alto, medio o bajo de adquirir una determinada enfermedad o una compañía de encuestas de opinión puede clasificar a las personas entrevistadas en aquellas que probablemente voten a un determinado partido político o estén indecisos, entre otras tareas.

La clasificación es una forma de predicción, donde el valor a predecir es una etiqueta. La predicción numérica (comúnmente denominada regresión) es la otra. En este caso, deseamos predecir un valor numérico, como las ganancias de una empresa o el precio de una acción.

Por otro lado, en el segundo tipo de conjuntos de datos, los datos no tienen ningún atributo especialmente designado, se denominan no etiquetados y en minería de datos se conoce a estos problemas como de aprendizaje no supervisado. Allí, el objetivo es simplemente extraer la mayor cantidad de información posible de los datos disponibles.

En el aprendizaje no supervisado, existen tareas de agrupamiento o de asociación. Los algoritmos de agrupamiento examinan los datos para encontrar grupos de elementos que son similares. Por ejemplo, una compañía de seguros podría agrupar a los clientes de acuerdo con los ingresos, la edad, los tipos de póliza o la experiencia previa en reclamos.

Por último, los algoritmos relacionados con la búsqueda de asociación son utilizados para encontrar cualquier relación existente entre los valores de las variables, mayoritariamente a través de reglas [85].

3.2 EMAIL MINIG: CONCEPTO Y ESTADO DEL ARTE

Existe un área particular de la minería de datos, denominada minería de textos, donde el conocimiento es generado mediante la adopción de bases de datos exclusivamente textuales como fuentes de datos [95].

Estas técnicas de análisis de texto se enfrentan a problemáticas muy complejas dentro del área de las ciencias de la computación, debido principalmente a la dificultad del análisis del lenguaje (derivada de su ambigüedad), fundamentalmente en la etapa de análisis semántico, como así también, a los relativamente escasos materiales de entrenamiento y a la capacidad de cómputo necesaria para ejecutar determinados algoritmos muy demandantes en recursos de hardware [17].

A su vez, el correo electrónico como fuente de datos posee un conjunto de características particulares respecto de otros elementos de texto que hace que existan diferencias y problemáticas particulares entre la minería de textos tradicional y la minería de correos electrónicos, conocida como *email mining* [12].

Por un lado, los correos electrónicos poseen información adicional en el encabezado que pueden ser explotados para la obtención de conocimiento. Asimismo, poseen una extensión reducida que hace que muchas técnicas de minería de textos sean ineficientes para estas fuentes de datos. Este tipo de comunicaciones, muchas veces se da en un contexto informal o inmersos en una cultura organizacional particular, por lo tanto, los errores ortográficos y gramaticales, así como abreviaturas y acrónimos aparecen con frecuencia.

Por otro lado, además de los datos textuales, los correos electrónicos pueden contener tipos más ricos de datos, como enlaces URL, marcas HTML e imágenes. Aprovechar al máximo esos datos no textuales en los correos electrónicos es un problema interesante y desafiante para abordar [91].

En este contexto, se ha aplicado *email mining* con diferentes objetivos como la detección de correo electrónico no deseado, la categorización de correo electrónico, el análisis de contactos, de propiedades de red de correo electrónico y visualización.

En lo relativo a clasificación de correos electrónicos, existen abordajes desde el procesamiento y generación de resúmenes [94], utilización de redes neuronales [12], clasificación para respuesta automática de correos [86], aplicación de técnicas basadas en máquinas vector soporte (SVM, por su acrónimo en inglés) y Naïve Bayes [91] así como utilización de *multi-view* y *semi-supervised learning* [103], entre otras.

Algunos autores que abordaron la clasificación de correos electrónicos para la respuesta automática categorizan las técnicas de acuerdo a, básicamente, tres enfoques de recuperación de texto: categorización de texto por aprendizaje automático, cálculo de similitud estadística de texto y coincidencia de patrones de texto y plantillas [86].

En relación a la categorización de texto mediante técnicas de aprendizaje automático, existen trabajos que desarrollaron modelos utilizando las técnicas de K-NN, Naïve Bayes, RIPPER y SVM, encontrando que SVM fue la técnica que demostró mejor performance [46].

También bajo el enfoque de categorización mediante aprendizaje automático, existen trabajos [87] en los cuales se compara la precisión de técnicas como K-means++, k-NN y Naïve Bayes, alcanzando niveles de precisión muy altos, por encima del 96 %, para K-means++.

En el mismo sentido, existen otras experiencias donde se realizan comparaciones entre los métodos de clasificación de Naïve Bayes, SMO, J48 y Random Forest [78]. En esas experiencias, el método fue simple, se preprocesaron los datos, se aplicó el algoritmo y luego se evaluó la performance. Del estudio se observó que el algoritmo *Random Forest* fue el que obtuvo la mejor precisión, siendo esta de un 95.5 % mientras que el algoritmo Naïve Bayes fue el más veloz en la construcción del clasificador.

Siguiendo la misma línea, en otra experiencia se comparó la precisión de diferentes algoritmos de clasificación como árboles de decisión, redes neuronales, Naive Bayes, K-Nearest Neighbor y SVM. Se utilizaron datos académicos para predecir la performance de los alumnos encontrando que los árboles de decisión y redes neuronales fueron los que mejor performance obtuvieron [91].

Otros abordajes a partir de la clasificación de correos electrónicos mediante el cálculo de similitud estadística también obtuvieron resultados alentadores [3]. En estos casos, el modelo mantiene respuestas estándar asociadas a una variedad de preguntas etiquetadas como preguntas frecuentes. Cuando llega un correo electrónico de consulta, el sistema hace coincidir las oraciones en la consulta con las preguntas de la etiqueta considerando la distancia entre conceptos en las oraciones utilizando *WordNet*.

Un enfoque alternativo es el basado en coincidencia de patrones de texto y plantillas [4], donde el sistema mantiene un diccionario que contiene palabras y la probabilidad de que una palabra aparezca en un mensaje de una determinada categoría de texto, categorizando los mensajes en base a esa probabilidad junto con información adicional que toma de los mensajes de consulta.

También existen técnicas de clasificación de correos electrónicos utilizando un enfoque denominado de múltiples vistas o *multi-view* [103]; lo cual implica generar múltiples grupos de características de los correos y aprovechar los algoritmos de *Disagreement-based Semi-Supervised Learning* que proporcionan herramientas para ser entrenados en diferentes vistas. En algunas experiencias se generaron dos grupos de características de los correos, internas y externas, donde las primeras explotan el cuerpo del correo y las últimas aprovechan otras como el asunto y los destinatarios y luego se utilizó *Disagreement-based Semi-Supervised Learning* para generar varios modelos a múltiples vistas y permitirles colaborar para explotar ejemplos no etiquetados [61, 103].

De forma más reciente, surgen los abordajes basados en *Deep Learning* [87] que implementan una red neuronal basada en un modelo *Long-Short-Term-Memory* para clasificar correos no deseados. Para resolver el problema de la gran cantidad de datos etiquetados

necesarios para los métodos de *Deep Learning*, utilizaron un método de aprendizaje activo. Este método selecciona diferentes muestras y sólo entrena esas, buscando disminuir el costo del etiquetado manual de los datos. Este modelo demostró una mejor performance con respecto a los tradicionales CNN y RNN. Otro acercamiento al tema de la predicción de correos no deseados es utilizando redes neuronales [12], también obteniendo buenos resultados con precisiones superiores al 85 %, lo cual demuestra el potencial de las redes neuronales para dicha tarea.

Por último, actualmente las estrategias basadas en *transformers* están generado una gran repercusión ya que a partir de la irrupción del modelo de representación del lenguaje denominado *Bidirectional Encoder Representations from Transformers*, o simplemente BERT, se han obtenido mejoras significativas respecto a los abordajes previos para el tratamiento y la clasificación de textos [1, 90].

3.3 CLASIFICACIÓN AUTOMÁTICA DE TEXTOS

Las técnicas de minería de textos persiguen dos grandes propósitos: la descripción y la predicción. Por una parte, las tareas descriptivas buscan obtener patrones que explican o resumen las relaciones subyacentes en los datos. Esto permite, por ejemplo, formular nuevas hipótesis considerando las palabras que utilizan las personas cotidianamente [27]. Por otro lado, las tareas predictivas, en las cuales está centrado este trabajo, consisten en la construcción de clasificadores automáticos que estiman la variable dependiente, usualmente llamada etiqueta o resultado, en función de determinadas características (variables independientes) extraídas de los documentos [60, 82].

La construcción de un clasificador automático, tradicionalmente sigue un proceso cuyas tareas se ven reflejadas en la Figure 3.2.

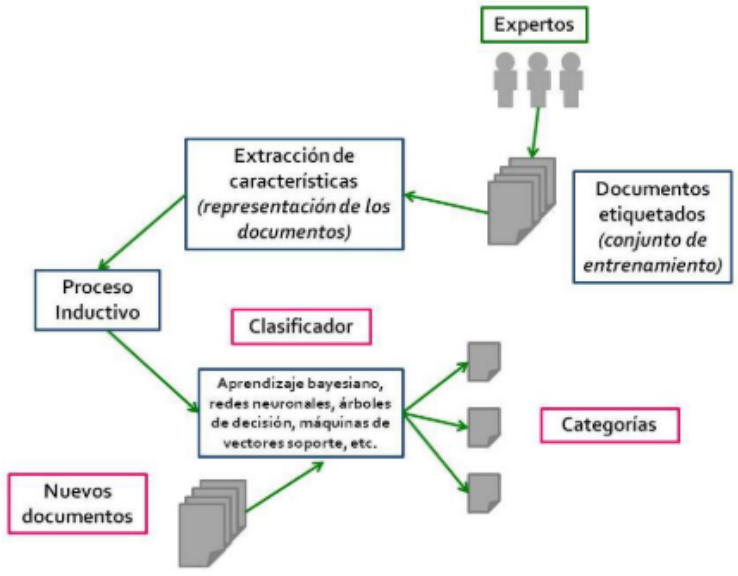

Figura 3.2: Etapas del proceso de construcción de un clasificador automático. [67]

Estas tareas o etapas, esquematizadas en la figura anterior, son las siguientes: etiquetado de documentos, extracción de características, entrenamiento del modelo, evaluación del modelo, utilización del modelo. En el trabajo *"Extracción de conocimiento con técnicas de minería de textos aplicadas a la psicología"*, de Mariñearena y otros[67], se introduce una noción de cada una de estas etapas:

1. **Etiquetado de documentos:** consiste en asignar la clase, categoría o valor numérico correcto (etiqueta) a cada documento del conjunto de entrenamiento.

2. **Representación de documentos / Extracción de características:** a partir de los documentos o textos crudos se genera una representación computacionalmente adecuada para su procesamiento por el módulo de análisis (aprendizaje inductivo). Un documento es una unidad de datos textual que puede corresponder a algún documento del mundo real, por ejemplo: un artículo científico, un escrito personal, un e-mail, los posts en los medios sociales como Facebook y Twitter, etc.

3. **Entrenamiento del modelo:** En las tareas predictivas, dada una colección de documentos, el siguiente paso será asignarle a cada documento una etiqueta o rótulo que representa una clase, categoría o valor numérico particular. La construcción de un clasificador automático que pueda realizar este tipo de tarea, se basa en un proceso inductivo de aprendizaje automático que para cada *input* o documento a clasificar siempre se genere el mismo *output* o asigne dicho documento a la misma clase.

4. **Evaluación del modelo:** Si un clasificador automático sólo fuera evaluado sobre los datos de entrenamiento con que fue generado, se correría el riesgo de obtener modelos que han "memorizado" dichos datos pero que tienen bajo desempeño sobre nuevos documentos. Por lo tanto, se evalúa la utilidad de las representaciones de los documentos y del modelo obtenido sobre un conjunto de prueba separado o utilizando esquemas más complejos. En estos esquemas, se mantiene separado el conjunto de entrenamiento del de prueba y se evalúa la precisión del clasificador midiendo capacidad de predecir la clase correcta para un documento no conocido para el modelo entrenado.

5. **Uso del modelo:** Una vez obtenido un clasificador con un desempeño "aceptable" de acuerdo al dominio de aplicación, éste es puesto en funcionamiento y sus resultados (predicciones) comienzan a ser aplicados sobre los nuevos datos que ingresan al sistema.

Si bien hasta aquí se introducen nociones de las etapas que entran en juego en la construcción de un clasificador automático, en los siguientes apartados se desarrollarán las principales técnicas utilizadas en estas etapas del proceso de extracción de conocimiento en documentos y que son pasibles de ser aplicadas a correos electrónicos en el marco del objetivo de este trabajo.

3.4 EMAIL MINING: CLASIFICACIÓN AUTOMÁTICA

El objetivo de este apartado, como se expresó anteriormente, es profundizar sobre las etapas de construcción de un clasificador automático, considerando particularmente la extracción de conocimiento de documentos textuales para su aplicación sobre correos electrónicos.

En particular, los clasificadores automáticos de texto, se pueden definir de la siguiente manera [88]: Dado un conjunto de documentos D, y un conjunto de clases (o etiquetas) C, se define una función F que asigna un valor del conjunto de C a cada documento en D; por ejemplo, en la clasificación de texto breve, D podría consistir en el conjunto de todos los anuncios clasificados en un periódico y, por lo tanto, C sería el conjunto de secciones de clasificados de ese mismo periódico.

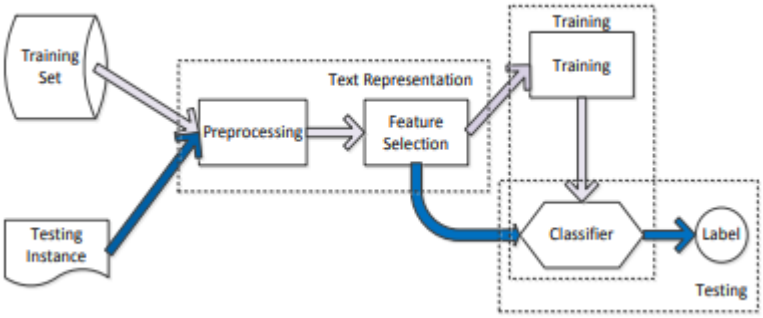

Figura 3.3: Estructura general de un clasificador automático de texto [88]

3.4.1 *Etiquetado de documentos*

De acuerdo al esquema planteado, en función de un proceso tradicional para la construcción de un clasificador automático de texto, una de las primeras tareas que se deben llevar a cabo es la clasificación inicial de un conjunto de documentos que luego serán utilizados como conjuntos de entrenamiento y prueba para el entrenamiento y validación del clasificador.

La estrategia tradicional para el etiquetado de documentos consiste en que esta tarea sea realizada por un humano, de forma manual. En muchas ocasiones, este etiquetado manual debe ser realizado por expertos en el tema que forma parte del problema que se desea abordar. Si bien estas etiquetas de expertos proporcionan la piedra angular tradicional para evaluar los modelos de aprendizaje automático, el acceso limitado o costoso a los expertos representa un cuello de botella [52].

A su vez, para caracterizar con precisión la efectividad de un sistema, la experiencia ha demostrado que deben evaluarse a la escala operativa en la que se utilizarán en la práctica, lo cual resulta en una limitación para esta metodología puesto que, debido a que los tamaños de las colecciones han crecido rápidamente en los últimos años, se ha vuelto cada

vez menos factible etiquetar manualmente tantos ejemplos usando el etiquetado experto tradicional [52].

En este sentido, han surgido metodologías alternativas que aportan mayor escalabilidad. En algunos sistemas de uso masivo, una estrategia posible es inferir etiquetas implícitas del comportamiento de las personas que utilizan el sistema, aunque para su consolidación requiere grandes poblaciones de usuarios, como es el caso de los buscadores de internet [49].

Otras estrategias de etiquetado de datos consisten en la *"supervisión distante"*, en la que los datos de entrenamiento son etiquetados a partir de algunas características del texto, como tags, emoticones y otros metadatos [37]. Este enfoque es particularmente interesante, y se han encontrado buenos resultados, en redes sociales en las cuales los emoticones pueden ser indicadores del sentimiento del usuario para los cuales se ha demostrado que poseen la ventaja de ser independientes del dominio, del tema y del tiempo [76].

Un enfoque alternativo que ha mostrado buenos resultados consiste en etiquetar un conjunto de palabras, representativas de cada clase para luego etiquetar un conjunto de documentos que se utilizarán para el entrenamiento del clasificador en función de la presencia de esas palabras representativas para una clase determinada. La clave para el funcionamiento de este enfoque es elegir un conjunto de palabras para cada clase que sean realmente representativas de la misma [65].

3.4.2 *Representación de documentos*

La representación de documentos consiste en una expresión computacionalmente adecuada de estas unidades de información, generalmente a partir de métricas cuantitativas y una estructura de datos, para su posterior procesamiento mediante una técnica de aprendizaje automático. Por su parte, un documento es una unidad de datos textual que corresponde a algún documento del mundo real y que será transformada en una instancia de un conjunto de datos, expresada a partir de un conjunto de sus características.

Desde el punto de vista de la minería de textos, la representación de los documentos consiste en una de técnica de preprocesamiento que se utiliza para reducir la complejidad de los mismos, transformando cada documento en un vector de características. La representación del texto y de su esencia es el aspecto más importante en la clasificación de documentos. Un documento de texto se representa típicamente como un vector de pesos correspondiente a sus términos, donde cada término aparece al menos una vez en un número mínimo de documentos [54].

Aunque un documento de texto expresa una gran variedad de información, lamentablemente carece de una estructura como la de las bases de datos tradicionales. Por lo tanto, los datos no estructurados, en particular los datos de texto de ejecución libre, deben transformarse en datos estructurados. Para ello, en la literatura se proponen muchas técnicas de preprocesamiento que se abordarán en este trabajo.

Luego de convertir estos datos no estructurados en estructurados, típicamente en un vector de características, es necesario definir un modelo de representación para los documentos que sea efectivo para la posterior construcción del sistema de clasificación [44].

En resumen, cuando se hace referencia a la representación de documentos, se habla de múltiples aspectos, donde dos de los más importantes son: la extracción de los atributos a partir de los cuales se representa un documento y la estructura que sostiene la representación del corpus de documentos.

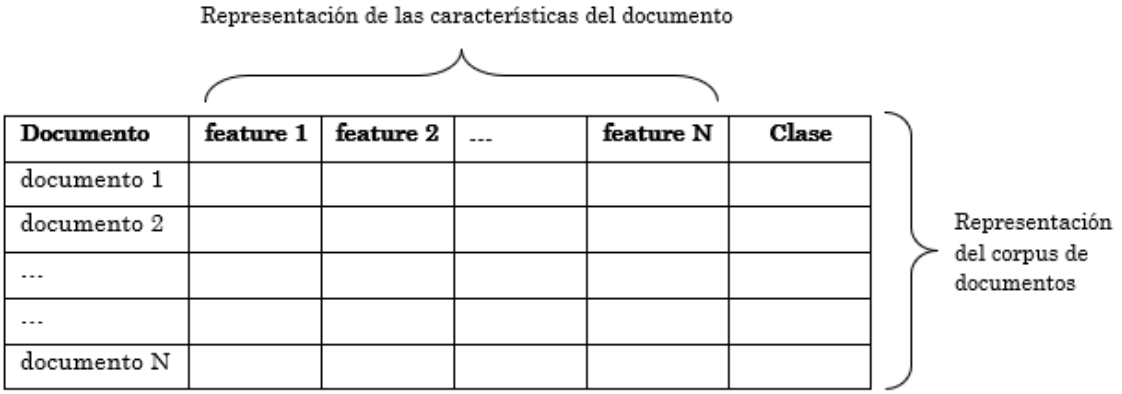

Figura 3.4: Esquema de representación de un corpus para la clasificación de documentos (elaboración propia)

3.4.3 *Extracción de características de los documentos*

Las características a través de las cuales se representan los documentos pueden ser de diferente naturaleza [56]. Por un lado, se encuentran las características estáticas, las cuales se denominan así dado que las métricas a calcular para cada documento son determinadas previo a su procesamiento y no están condicionadas por el corpus a procesar. Estas características estáticas pueden clasificarse, a su vez, en características léxicas, sintácticas, estructurales y específicas del contenido [104]. Al mismo tiempo, las características léxicas, pueden subdividirse en características basadas en caracteres o palabras y algunos autores las utilizan para realizar diferenciaciones estilográficas.

Inicialmente, estas representaciones de texto se desarrollaron para identificar qué subconjuntos de características serían más confiables para determinar la autoría en un entorno de aprendizaje supervisado. Se encontró que todos los conjuntos de características agregan algo de información, ya que la colección que incluía los cuatro subconjuntos de características fue la más precisa al aplicar una serie de algoritmos de aprendizaje supervisado al conjunto de datos resultante [56]. Para graficar esta clasificación, se define un ejemplo de característica por cada una de las cuatro categorías:

- Características léxicas: proporción de letras mayúsculas (basada en caracteres) y promedio del largo de las palabras (basada en palabras).
- Características sintácticas: frecuencia del uso de un determinado signo de puntuación.

- Características estructurales: cantidad de frases promedio por párrafo.
- Características específicas del contenido: cantidad de menciones a un determinado autor reconocido en la disciplina (si interpretamos al contenido como el dominio) y cantidad de enlaces (si interpretamos contenido como tipo de texto).

Por otro lado, Layton [56] identifica las mencionadas características dinámicas o variables, las cuales se derivan automáticamente del procesamiento de los documentos inherentes al trabajo, por lo que no se puede definir antes cuales serán exactamente estas características, ya que variarán de acuerdo a la colección de documentos considerada. Las características dinámicas se construyen a través de los términos presentes en los documentos o bien a partir de n-gramas. Cuando se trabaja con n-gramas, se considera a cada documento como una serie de subsecuencias superpuestas de tokens. Un documento puede considerarse como una secuencia de caracteres, palabras, oraciones o incluso párrafos. Un n-grama basado en caracteres considera un documento como una serie de subconjuntos secuenciales superpuestos de caracteres.

Uno de los principales desafíos en la clasificación de textos es la dimensionalidad extremadamente alta. Aquí es donde interviene fuertemente la etapa de preprocesamiento, la cual consiste en aclarar los límites de la estructura de cada idioma y eliminar, en la medida de lo posible, los factores dependientes del idioma, la tokenización, la eliminación de palabras vacías y la derivación de las raíces de la palabra o *stemming* [54].

La extracción de características es el primer paso del procesamiento y dos de las tareas asociadas consisten en eliminar las palabras vacías y las palabras derivadas. Estas tareas se sustentan en que los documentos están representados por una gran cantidad de características y la mayoría de ellas pueden ser irrelevantes o ruidosas.

Algunos autores, esquematizaron el proceso general asociado a la extracción de características y representación de documentos para la clasificación de textos previo a la aplicación de un algoritmo de aprendizaje [54].

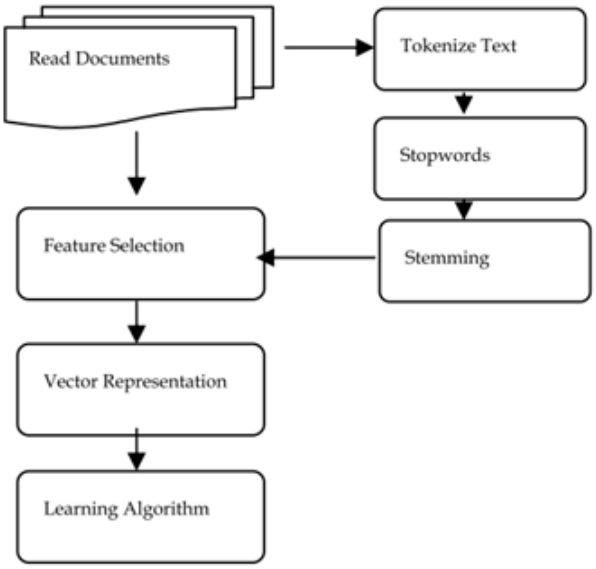

Figura 3.5: Proceso de extracción de características para la clasificación de documentos [54]

En ese esquema del proceso de extracción de características se identifican las siguientes tareas asociadas:

- **Tokenización:** un documento se trata como una cadena y luego se divide en una lista de tokens o términos.
- **Eliminación de palabras vacías:** Las palabras vacías como "el", "la", "y", son términos que ocurren con frecuencia en los documentos, pero no aportan información para diferenciarlos.
- **Palabras derivadas o stemming:** opcionalmente, se aplican algoritmos de derivación que convierten diferentes formas de las palabras a formas canónicas únicas. Este paso consiste en el proceso de combinar tokens a través de su raíz bajo el supuesto que todas las acepciones de la palabra poseen el mismo valor para la clasificación.

A continuación de las tareas de extracción de características, otra tarea importante del preprocesamiento consiste en la selección de características para construir un espacio vectorial que mejore la escalabilidad, la eficiencia y la precisión de un clasificador de texto. La idea principal de la tarea de selección de características es seleccionar un subconjunto de atributos de los documentos originales que conserven su esencia, proporcionen una mejor comprensión de los datos y mejoren el proceso de aprendizaje [73].

3.4.4 *Estrategias de Representación de documentos*

Aunque un documento de texto expresa una gran variedad de información, lamentablemente carece de la estructura impuesta en una base de datos tradicional; por lo tanto, los datos no estructurados, particularmente los datos de texto libre, deben transformarse en datos estructurados previo a la aplicación de técnicas de aprendizaje automático. Después de convertir datos no estructurados en datos estructurados, necesitamos tener un modelo de representación de documentos efectivo para construir un sistema de clasificación eficiente [44].

En los siguientes apartados, se presentan diferentes estrategias de representación de documentos, pasibles de ser utilizadas para la clasificación de correos electrónicos.

3.4.4.1 *Bolsa de palabras (Bag of words)*

Bag of Word (BoW) es uno de los métodos básicos para representar un documento y uno de los más antiguos. BoW consiste en generar un vector que representa un documento, generalmente utilizando el recuento de frecuencia de sus términos [44].

Este método de representación se denomina modelo de espacio vectorial [80] y asume que existe un espacio de documentos D_i, los cuales se identifican por uno o más términos de índice T_j y que están ponderados según su importancia o con ponderaciones restringidas a 0 y 1. Así, cada documento D_i está representado por un vector t-dimensional.

A partir de esta representación, dados los vectores para dos documentos, es posible calcular un coeficiente de similitud entre ellos, $S(D_i, D_j)$, que refleja el grado de semejanza

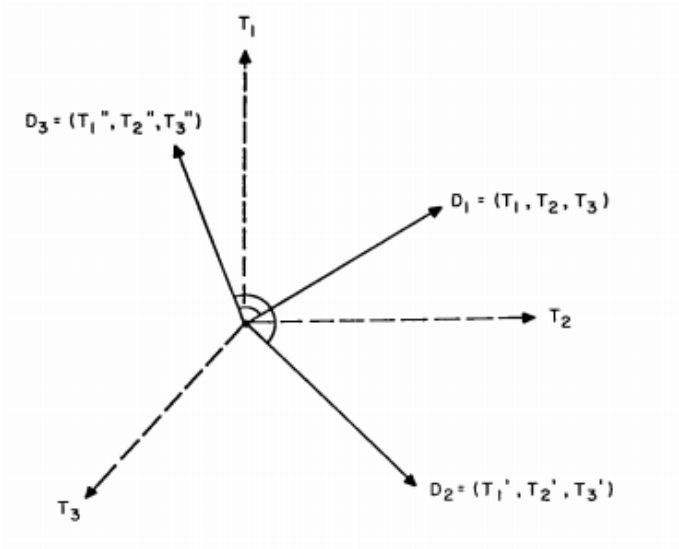

Figura 3.6: Representación vectorial del espacio de documentos. [80]

de sus términos y ponderaciones correspondientes. Tal medida de similitud podría ser el producto interno de los dos vectores o, alternativamente, una función inversa del ángulo entre los pares de vectores correspondientes.

Además, en lugar de identificar cada documento mediante un vector completo que se origina en el punto 0 del sistema de coordenadas, la distancia relativa entre los vectores se conserva normalizando todas las longitudes de los vectores a uno, y considerando la proyección de los vectores sobre la envolvente del espacio representado por la esfera unitaria. En ese caso, cada documento puede estar representado por un solo punto cuya posición está especificada por el área indicada según el vector correspondiente al documento. Luego, dos documentos con términos de índice similares se representan mediante puntos que están muy juntos en el espacio y, en general, la distancia entre dos puntos del documento en el espacio se correlaciona inversamente con la similitud entre los vectores correspondientes [80].

Esta estrategia es también denominada *bag of words* (bolsa de palabras) ya que las palabras son tomadas como características y los documentos se tratan simplemente como colecciones de palabras desordenadas donde los valores se asignan a cada palabra generalmente según si la palabra aparece en un documento o la frecuencia con que aparece [62]. Desafortunadamente, este esquema tiene sus limitaciones. Algunas de ellas son la alta dimensionalidad de la representación, la pérdida de correlación con palabras adyacentes y la pérdida de relación semántica que existe entre los términos de un documento.

La solución ampliamente aplicada para la primera restricción es la eliminación de características en el paso de preparación. Algunos criterios de selección como chi-cuadrado (χ^2) y la ganancia de información (GI) resultan interesantes para esta actividad [102].

Respecto al segundo problema o limitación, por un lado, se utilizan métodos de ponderación de términos para asignar los pesos adecuados al término para mejorar el rendimiento de la clasificación de texto. Por otro lado, se han propuesto representaciones ontológicas para un documento con el objetivo de mantener la relación semántica entre los términos

en ese documento. Este modelo de ontología conserva el conocimiento de dominio de un término presente en un documento. Sin embargo, la construcción automática de ontologías es una tarea difícil debido a la falta de una base de conocimiento estructurada. Otras soluciones utilizadas consisten en la utilización de n-gramas de palabras a partir de los cuales es posible extraer una cadena larga en un documento [44].

3.4.4.2 *Modelados de Tópicos*

Bajo el enfoque de *topic modeling*, la indexación semántica latente (o análisis semántico latente, LSA) se ha aplicado ampliamente a la matriz de documentos y términos para reducir su dimensionalidad y producir una dimensión latente informativa, más acotada. LSA utiliza la descomposición de valores singulares (SVD) como método para construir dimensiones significativas derivadas de una matriz documento-término. Al igual que otras técnicas, como Análisis de Componentes Principales (PCA), puede aproximarse a una matriz N-dimensional usando menos dimensiones [102]. A su vez, existen algunos enfoques de indexación latente, como PLSA (Probabilistic LSA), que no utilizan matrices sino métodos probabilísticos.

El Análisis Semántico Latente (LSA) se basa en el supuesto de que existe una estructura semántica subyacente en los datos textuales, y que la relación entre los términos y los documentos se puede reescribir en esta forma de estructura semántica. Basado en métodos estadísticos, LSA extrae y cuantifica la estructura semántica [88]. El proceso de LSA, basado en SVD, puede resumirse de la siguiente manera:

- Los documentos se representan como vectores en un espacio vectorial. Por lo tanto, la matriz término-documento se representa como $A_{mn} = [a_{ij}]_{mxn}$ donde cada posición corresponde a la presencia o ausencia ponderada de un término (una fila i) en un documento (una columna j). Esta matriz suele ser muy rala, ya que la mayoría de los documentos contienen solo un pequeño porcentaje del número total de términos que se ven en la colección completa.

- Se calcula el peso de cada a_{ij}. La forma tradicional consiste en la expresión $a_{ij} = LW_{ij}xGW_{ij}$ donde LW es el peso local del término i en el documento j y GW el peso global del término i en el dataset. El peso local de un término se calcula como el logaritmo de la frecuencia total del término i en el documento j mientras que el peso global de un término es igual a la entropía del término en el conjunto de datos.

- LSA usa la descomposición del valor singular (SVD) del término por la matriz $A_{mn} = [a_{ij}]_{mxn}$. SVD de $A_{mn} = [a_{ij}]_{mxn}$ consiste en el producto de tres matrices:

$$A = \sum_{i=1}^{r} u_i \sigma_i v_i = [u_1, ..., u_r] = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_r \end{bmatrix} \tag{3.1}$$

donde u y v son las matrices de los vectores singulares izquierdo y derecho y σ es la matriz diagonal de valores singulares. Los elementos de la diagonal están ordenados por magnitud y, por lo tanto, estas matrices se pueden simplificar estableciendo los valores k más pequeños en cero para luego eliminar estas columnas.

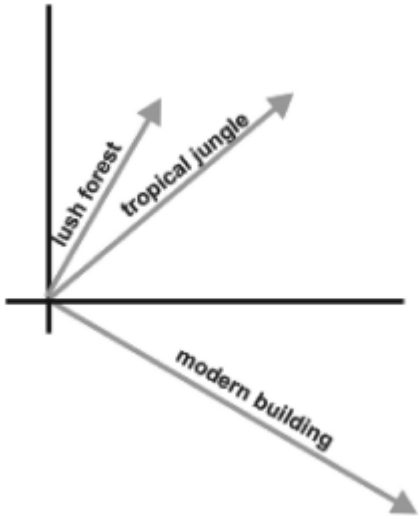

Figura 3.7: Ejemplo gráfico de LSA que representa tres textos mediante vectores [57]

En resumen, LSA generalmente mide la similitud entre dos fragmentos de texto usando el coseno entre los dos vectores. Si el coseno está cerca de uno las dos secciones del texto son muy similares semánticamente, y si el coseno está cerca de cero las dos secciones no están relacionadas semánticamente en absoluto. En resumen, LSA se ha propuesto como un modelo adecuado para simular la representación del léxico [51].

Todos los algoritmos de clasificación que son adecuados para el modelo de espacio vectorial también pueden aplicarse al modelo de clasificación LSA. Se proponen muchos métodos de clasificación que combinan LSA y algoritmos tradicionales como el algoritmo de clasificación de secuencias, Naive Bayes, KNN y SVM para mejorar la precisión de la clasificación de textos breves [88].

Otra aproximación al modelado de tópicos es LDA (*Latent Dirichlet Analisis*), el cual se puede considerar como otro modelo de representación de documentos en el que el algoritmo estocástico agrupa los documentos basándose en estadísticas de co-ocurrencia.

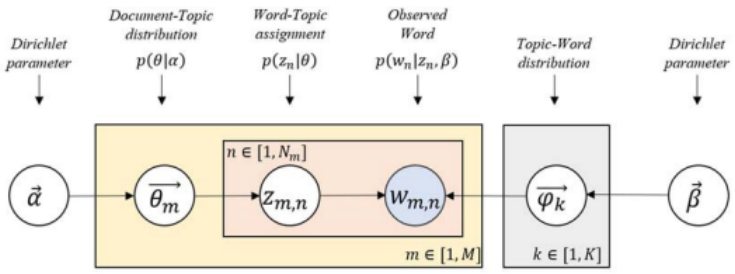

Figura 3.8: Modelo gráfico de *Latent Dirichlet Allocation (LDA)* [57]

La idea básica de LDA es que cada documento tiene un tema y estos temas se pueden definir como una distribución de palabras [57].

Los temas y sus probabilidades se aprenden como distribuciones discretas donde los temas consisten en un conjunto de palabras. Los modelos toman el tamaño del tema y el de las palabras como parámetro antes de la fase de entrenamiento.

3.4.4.3 *Incrustaciones de palabras (word embeddings)*

Una línea de investigación bastante actual respecto de la representación de documentos consiste en la utilización de información contextual junto con modelos simples de redes neuronales para obtener representaciones de palabras y frases en el espacio vectorial [101].

Tradicionalmente, muchos sistemas y técnicas del procesamiento del lenguaje natural tratan las palabras como unidades atómicas donde no existe una noción de similitud entre palabras, ya que se representan como índices en un vocabulario. Esta elección tiene varias buenas razones: simplicidad, solidez y la observación de que los modelos simples entrenados con grandes cantidades de datos superan a los sistemas complejos entrenados con menos datos. Sin embargo, las técnicas simples están en sus límites en muchas tareas [70].

En función de estas limitaciones, un objetivo del modelado de lenguaje estadístico consistió en aprender la función de probabilidad conjunta de secuencias de palabras en un idioma. Esto es intrínsecamente difícil debido a la alta dimensionalidad existente en el procesamiento del texto, dado que es probable que la secuencia de palabras en la que se probará el modelo sea diferente de todas las secuencias de palabras vistas durante el entrenamiento [11]. Para hacer frente a esta limitación, algunos autores [11], se propusieron aprender una representación distribuida de palabras que permitiera que cada oración de entrenamiento informe al modelo sobre un número exponencial de oraciones semánticamente vecinas. En esencia, el modelo aprende simultáneamente una representación distribuida para cada palabra junto con la función de probabilidad para secuencias de palabras, expresada en términos de estas representaciones. Este comportamiento permite la generalización puesto que una secuencia de palabras que nunca antes se ha visto obtiene una alta probabilidad si está formada por palabras que son similares (en el sentido de tener una representación cercana) a palabras que forman una oración ya vista. En pocas palabras, la idea del enfoque propuesto se puede resumir de la siguiente manera:

1. Se asocia cada palabra en el vocabulario a un vector de características de palabra distribuida (un vector de valores reales de m características definidas previo al entrenamiento),

2. Se expresa la función de probabilidad conjunta de secuencias de palabras en términos de los vectores de características de estas palabras en la secuencia, y

3. Se aprenden simultáneamente los vectores de características de las palabras y los parámetros de esa función de probabilidad.

Aquí, el vector de características representa diferentes aspectos de la palabra donde cada palabra está asociada con un punto en un espacio vectorial y la dimensionalidad de estas

características es, naturalmente, mucho menor que el tamaño del vocabulario. Además, es importante marcar que la función de probabilidad se expresa como el producto de probabilidades condicionales de la siguiente palabra dadas las anteriores.

Queda claro que, a partir del hallazgo de un vector de características que represente a cada palabra del vocabulario, es posible encontrar similitudes entre los vectores (términos del vocabulario) a partir de operaciones de álgebra lineal; sin embargo, esto representó en ese momento todo un desafío computacional. Recientemente, los modelos de lenguaje basados en redes neuronales (NNLM, por su acrónimo en inglés) han ganado gran atención ya que han demostrado un rendimiento prometedor y reducen la complejidad del tiempo insumido por el costo computacional. Una de sus características más importantes es la capacidad de generar incrustaciones densas y cortas, es decir, incrustaciones de palabras.

En esta arquitectura de redes neuronales, cada palabra se asocia inicialmente con un vector aleatorio y a medida que una red neuronal de dos capas procesa el corpus textual, los vectores se actualizan iterativamente mediante la aplicación de descenso de gradiente estocástico (SGD), donde el gradiente se mide por retropropagación, o *back-propagation*. El objetivo es adivinar la última palabra de una secuencia de palabras determinada y, por lo tanto, la tarea de predicción es típicamente similar a la clasificación de clases múltiples donde se usa la función *soft-max* para calcular estimación de probabilidad de clase. La red finalmente aprende las incrustaciones de todas las palabras que aparecen en el corpus por convergencia.

Como uno de los modelos de incrustaciones de palabras basados en redes neuronales más populares, se encuentra el modelo *word2vec*, el cual dispone de dos arquitecturas diferentes, a saber: bolsa de palabras continua (*CBoW*) y *Skip-gram* [70, 102]. De forma intuitiva, estas dos arquitecturas funcionan de manera opuesta dado que mientras que la arquitectura *CBoW* predice la palabra actual basada en el contexto, el enfoque *Skip-gram* predice las palabras circundantes en función de la palabra actual.

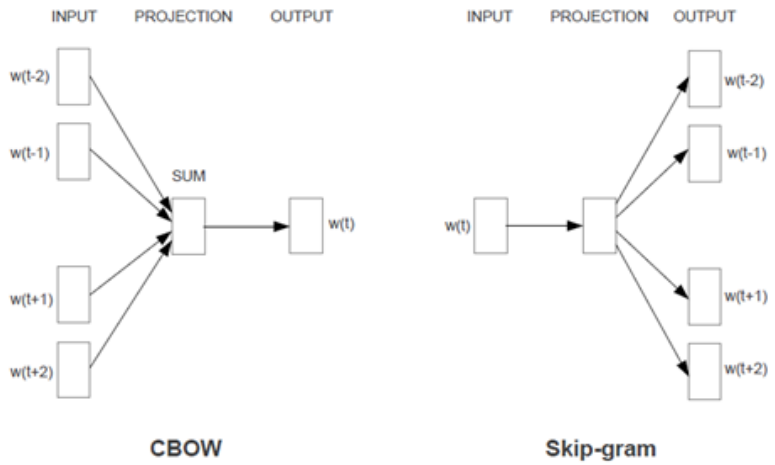

Figura 3.9: Arquitecturas *CBoW* y *Skip-gram* del Modelo word2vec. [70]

3.4.4.4 *Representación de codificadores direccionales de transformadores (BERT)*

Hasta la aparición de la arquitectura de *Transformers* [96], las redes neuronales recurrentes, y las redes LSTM se habían establecido como los enfoques de vanguardia en el modelado de secuencias y problemas de transducción como el modelado del lenguaje y la traducción automática de texto. En 2017 se propone una nueva arquitectura de red neuronal, más simple y paralelizable, denominada *Transformer* [96], basada únicamente en mecanismos de atención, prescindiendo por completo de recurrencia y convoluciones.

Hasta la aparición de los *transformers*, los modelos más competitivos de procesamiento y transducción de secuencias eran los basados en estructuras de *encoder-decoder* [23]. En cambio, aquí el codificador mapea una secuencia de entrada de representaciones de símbolos $(x_1, ..., x_n)$ a una secuencia de representaciones continuas $z = (z_1, ..., z_n)$. Dado z, el decodificador genera una secuencia de salida $(y_1, ..., y_m)$ de símbolos un elemento a la vez. En cada paso, el modelo es auto-regresivo, consumiendo los símbolos generados previamente como entrada adicional al generar el siguiente. El transformador sigue esta arquitectura general utilizando auto-atención apilada y capas puntuales y completamente conectadas tanto para el codificador como para el decodificador.

Los mecanismos de atención [8] surgen para resolver algunas limitaciones que presentan los modelos *encoder-decoder* para administrar la información contenida en los vectores de contexto dado que representar toda la cadena de entrada en un mismo vector puede ocasionar que se pierda información de los primeros elementos de la cadena. Por lo cual, el mecanismo de atención permite que el modelo se centre en las partes más importantes del vector de contexto. Para ello el encoder, en lugar de enviar sólo el último estado oculto, envía la información de todos los estados ocultos. Así, el decoder aplica el mecanismo de atención de modo que pueda leer el vector de contexto, actualizarlo, capturar toda la información relevante y devolver una salida adecuada en base al contexto actual.

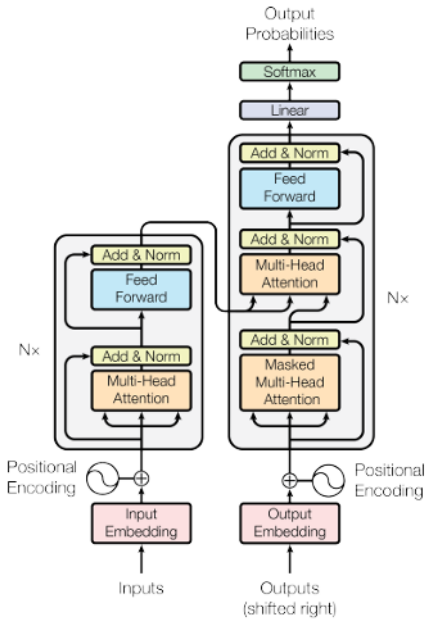

Figura 3.10: Arquitectura del modelo de *Transformer* [96]

A partir de las arquitecturas de *tranformers* basadas en mecanismos de atención, se presenta, en 2018, un nuevo modelo de representación de lenguaje llamado BERT *(Bidirectional Encoder Representations from Transformers)*. A diferencia de los modelos anteriores de representación de idiomas, BERT está diseñado para entrenar previamente representaciones bidireccionales profundas a partir de texto sin etiquetar en todas las capas. Como resultado, el modelo BERT previamente entrenado se puede ajustar con solo una capa de salida adicional para crear modelos de vanguardia para una amplia gama de tareas como la respuesta a preguntas y la inferencia de lenguaje, sin modificaciones sustanciales de la arquitectura específica de la tarea [25].

Para que BERT sea capaz de gestionar diferentes tareas de aprendizaje automático, la representación de la entrada debe ser capaz de identificar de manera inequívoca tanto una sola oración como un par de oraciones en una secuencia. En el contexto del modelo BERT, se entiende a una "sentencia" (u "oración") como un espacio arbitrario de texto contiguo, en lugar de una oración lingüística real.

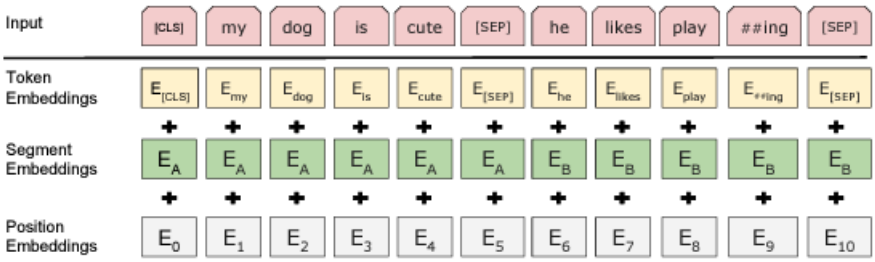

Figura 3.11: Representación de entrada en BERT [25]

Una "sentencia" se refiere a una secuencia de tokens de entrada a BERT, que puede ser una sola oración o dos oraciones empaquetadas juntas. Aquí, se diferencian sentencias de dos formas. Primero, se separa cada token, con un token especial ([SEP]). En segundo lugar, se agrega una incrustación aprendida a cada token que indica si pertenece a la oración *A* o a la oración *B*.

Luego, sintéticamente, el entrenamiento de BERT consta de dos pasos: pre-entrenamiento inicial y ajuste fino posterior. Durante el entrenamiento previo, el modelo se entrena con datos sin etiquetar en diferentes tareas. Luego, para el ajuste fino, el modelo BERT se inicializa primero con los parámetros del modelo pre-entrenado, los cuales se ajustan en esta etapa utilizando datos etiquetados de las tareas posteriores.

La etapa de pre-entrenamiento inicial está compuesta de dos tareas [25]:

- **Enmascarado LM:** intuitivamente, es razonable creer que un modelo bidireccional profundo sea estrictamente más poderoso que un modelo de izquierda a derecha o la concatenación superficial de un modelo de izquierda a derecha y de derecha a izquierda. Para entrenar una representación bidireccional profunda, este modelo enmascara un porcentaje de los tokens de entrada al azar y luego predice esos tokens enmascarados.

- **Predicción de la próxima sentencia:** muchas tareas específicas posteriores, como la respuesta a preguntas (QA) y la clasificación de texto, se basan en la comprensión de la relación entre dos oraciones. Con el fin de entrenar un modelo que comprenda las relaciones entre oraciones, se realiza un entrenamiento previo para una tarea binarizada de predicción de la siguiente oración que se puede generar trivialmente a partir de cualquier corpus monolingüe.

El proceso de *Fine-Tuning* resulta más sencillo que el anterior ya que el mecanismo de auto-atención en el *Transformer* permite que BERT modele muchas tareas posteriores, ya sea que involucren texto único o pares de texto, intercambiando las entradas y salidas apropiadas. Para cada tarea, simplemente se conectan las entradas y salidas específicas de la tarea en BERT y se ajustan todos los parámetros de un extremo a otro. En la salida, las representaciones de token se alimentan a una capa de salida para tareas de nivel de token, como etiquetado de secuencia o respuesta a preguntas, y la representación [CLS] se alimenta a una capa de salida para clasificación, como análisis de vinculación o sentimiento. En comparación con el entrenamiento previo, el ajuste fino es relativamente económico.

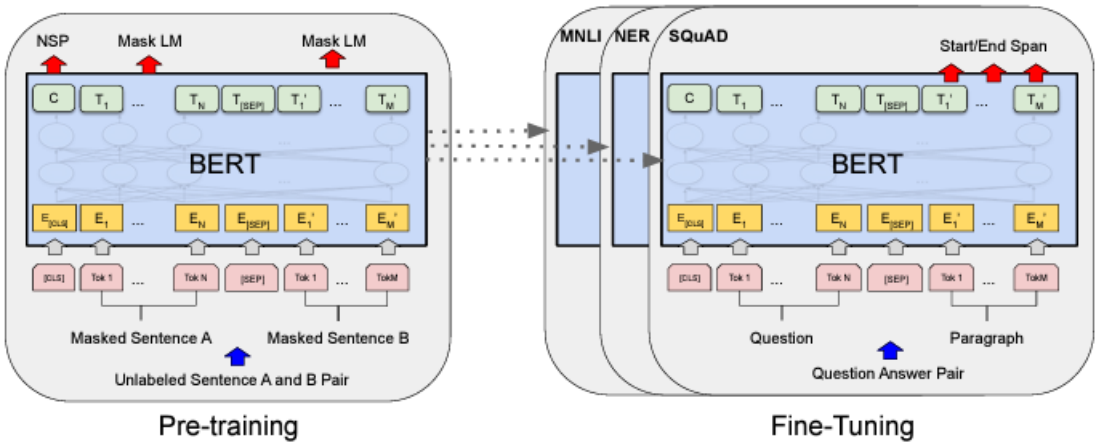

Figura 3.12: Procedimiento general de pre-entrenamiento y ajuste para BERT. [25]

3.4.5 *Entrenamiento del Modelo*

Luego de obtener los datos, realizar el preprocesamiento de los mismos para la extracción de características, etiquetarlos y avanzar en un esquema de representación, se entrena el clasificador utilizando distintos enfoques o algoritmos [77], algunos de los cuales son: el aprendizaje bayesiano [68], regresión logística, redes neuronales, árboles de decisión y máquinas de vectores soporte [50].

El modelo generado a partir del entrenamiento debe ser capaz de capturar las características distintivas de los documentos del conjunto de entrenamiento para luego poder analizar otros textos no observados previamente, lográndose así la capacidad de generalización del clasificador, el cual se suele evaluar sobre otro conjunto de prueba separado [67].

Este proceso de aprendizaje, en matemática, se lo conoce como aproximación de una función y consiste en buscar en un espacio de hipótesis, una hipótesis que sea consistente con los datos de entrenamiento pero que pueda además clasificar correctamente otros datos no presentes en ese conjunto.

A la fecha, y debido a la cantidad de algoritmos de aprendizaje existentes, resulta muy complejo sistematizar todos los abordajes posibles; no obstante, a continuación, se realiza una breve reseña de los algoritmos más utilizados para la clasificación automática de textos.

3.4.5.1 *Clasificador de Naïve Bayes*

Naïve Bayes es el clasificador probabilístico más simple que se utiliza para categorizar documentos de texto, generalmente elegido como *baseline* o linea base en las experimentaciones.

Este clasificador se basa en el Teorema de Bayes [10], elaborado por Thomas Bayes -un clérigo del siglo XVIII-, para el cálculo de probabilidades condicionales que plantea lo siguiente [31]:

- Sea $A_1, A_2, ..., A_n$ un conjunto de sucesos mutuamente excluyentes y cuya unión es el total o sea 1, y tales que la probabilidad de cada uno de ellos es distinta de cero.
- Sea B un suceso cualquiera del que se conocen las probabilidades condicionales $P(B|A_i)$.

Entonces la probabilidad $P(A_i|B)$ viene dada por la expresión:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} = \frac{P(B|A_i)P(A_i)}{\sum_{j=1}^{n} P(B|A_j)P(A_j)} \tag{3.2}$$

En la fórmula anterior, $P(A_i)$ son las probabilidades a priori, $P(B|A_i)$ es la probabilidad de B en la hipótesis A_i y $P(A_i|B)$ son las probabilidades a posteriori.

El clasificador de Naïve Bayes estima la probabilidad conjunta de que un documento d_i pertenezca a la clase C_k expresada como $P(d_i|C_k)$. La salida del clasificador es la probabilidad de que el documento pertenezca a cada una de las clases, representado en un vector de $|C|$ elementos.

Además, en términos del cálculo de la probabilidad, el Teorema de Bayes original puede ser reescrito, de forma más simple, como:

$$P(C_k|d_i) = P(d_i|C_k) \times \frac{P(C_k)}{P(d_i)} \tag{3.3}$$

A su vez, puesto que $P(d_i)$ será una constante para todas las clases, es posible eliminar el denominador del segundo factor, ya que lo que se busca es maximizar la fórmula para optar por una clase para la clasificación, la que finalmente se estima como:

$$P(C_k|d_i) = P(d_i|C_k) \times P(C_k) \tag{3.4}$$

Lo cual, llevando esta situación a la clasificación de documentos, es posibles expresar como:

$$P(d_i|C_k) = P(W_{1,i}, w_{2,i}, ..., w_{|V|,i}|C_k) = \prod_{j=1}^{|V|} P(w_{j,i}|C_k) \tag{3.5}$$

Es importante marcar que el clasificador Naïve Bayes presupone que la palabra w_j en el documento d_i no está correlacionada con la aparición del resto de las palabras w_{ji}.

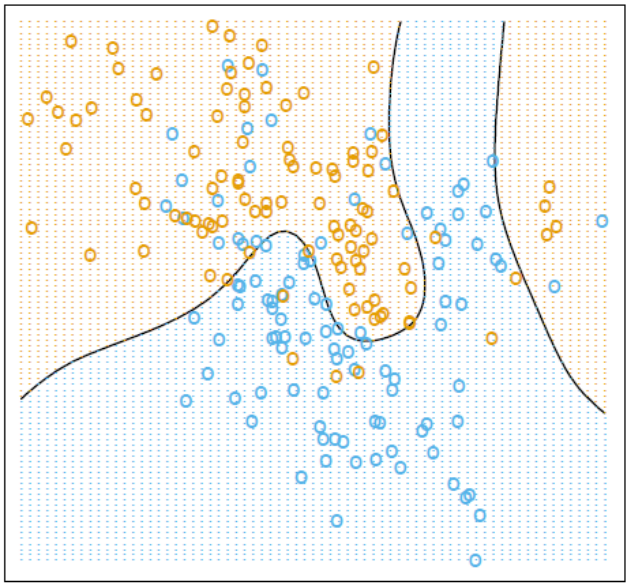

Figura 3.13: Frontera de decisión de un clasificador bayesiano para un problema binario [36]

Luego de las aclaraciones anteriores, el problema de clasificación del modelo se reduce a estimar la probabilidad de cada palabra $w_{j,i}$ respecto de la clase C y optar por la clase que maximice esa probabilidad [44].

3.4.5.2 *Máquina Vector Soporte*

La máquina de vector soporte (SVM) es un algoritmo de clasificación y regresión que fue desarrollado por Vapnik a mediados de 1990 y fue ganando popularidad a lo largo del tiempo debido a algunas características atractivas y su rendimiento empírico. SVM contiene una gama de algoritmos de clasificación y regresión que se basan en el principio de Minimización del Riesgo Estructural (SRM) de la teoría del aprendizaje estadístico, el cual consiste en encontrar un hiperplano óptimo para el que se pueda garantizar el error verdadero más bajo [48].

El objetivo principal de las máquinas de vector soporte es seleccionar el hiperplano que separe las instancias de entrenamiento con un criterio de distancia máxima. Este hiperplano objetivo se encuentra seleccionando hiperplanos que son tangenciales a categorías diferentes, es decir, que incluyen al menos una instancia de entrenamiento de cada categoría, al tiempo que proporcionan una separación perfecta entre todas las instancias de entrenamiento de esa clase.

Los hiperplanos tangenciales que se definen a partir de las instancias de entrenamiento son los vectores soporte mientras que la distancia entre los dos hiperplanos tangenciales es el margen. Una vez que se ha maximizado el margen, el hiperplano objetivo está en el medio [85].

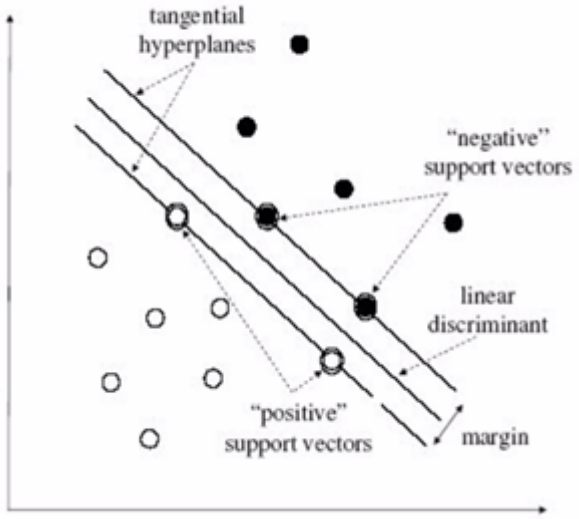

Figura 3.14: Esquema de Máquina de vector soporte para un problema linealmente separable [48]

Para el cálculo de estas distancias y la búsqueda de los hiperplanos, las SVM utilizan funciones denominadas *kernels*. Las funciones del *kernel* son las que permiten separar los puntos d-dimensionales en n dimensiones. La idea básica de un kernel es que proporciona el equivalente a mapear un espacio de entrada separable no lineal, a un espacio de características de mayor dimensión que es linealmente separable.

La técnica de SVM requiere experiencia para su uso eficaz, puesto que hay muchas funciones de kernel diferentes disponibles y la selección de la óptima resulta fundamental para su rendimiento ya que cada una de estas funciones tiene ventajas en ciertos conjuntos de datos, por lo que es necesario explorarlas. Por otro lado, SVM es una técnica que funciona mejor en conjuntos de datos de tamaño mediano [85].

3.4.5.3 *Long short-term memory (LSTM)*

LSTM *(Long short-term memory)* es un tipo de red neuronal que se encuentra en el grupo de las redes neuronales recurrentes, las cuales admiten como entrada todo tipo de datos, aunque este trabajo se centra en las cadenas de texto, asumiendo que cada entrada es un token diferente. Este tipo de redes neuronales, proveen dos características que mejoran

sustancialmente el rendimiento de las redes neuronales convencionales para el tratamiento de texto.

En primer lugar, muchas aplicaciones centradas en secuencias, como el texto, a menudo se procesan como bolsas de palabras. Este enfoque ignora el orden de las palabras en el documento y funciona bien para documentos de tamaño razonable. Sin embargo, en aplicaciones donde la interpretación semántica de la oración es importante, o en las que el tamaño del segmento de texto es relativamente pequeño, este enfoque es simplemente inadecuado. La solución provista por las redes neuronales recurrentes es evitar el enfoque de bolsa de palabras y crear una entrada para cada posición en la secuencia.

La otra mejora permite afrontar el problema de que la longitud de las oraciones de los documentos sea diferente. En algunos casos, la longitud de la secuencia de entrada de un texto puede llegar a los cientos de miles de palabras y cualquier cambio en el orden de las palabras pueden llevar a connotaciones semánticamente distintas, por lo que resulta importante codificar de alguna manera la información sobre el orden de cada palabra dentro de la arquitectura de la red [2].

Este requerimiento se satisface naturalmente con el uso de redes neuronales recurrentes (RNN). En una red neuronal recurrente, existe una correspondencia uno a uno entre las capas de la red y las posiciones específicas de la secuencia. La posición en la secuencia también se conoce como su marca de tiempo (*timestamp*). Por lo tanto, en lugar de un número variable de entradas en una sola capa de entrada, la red contiene un número variable de capas y cada capa tiene una única entrada correspondiente a esa marca de tiempo.

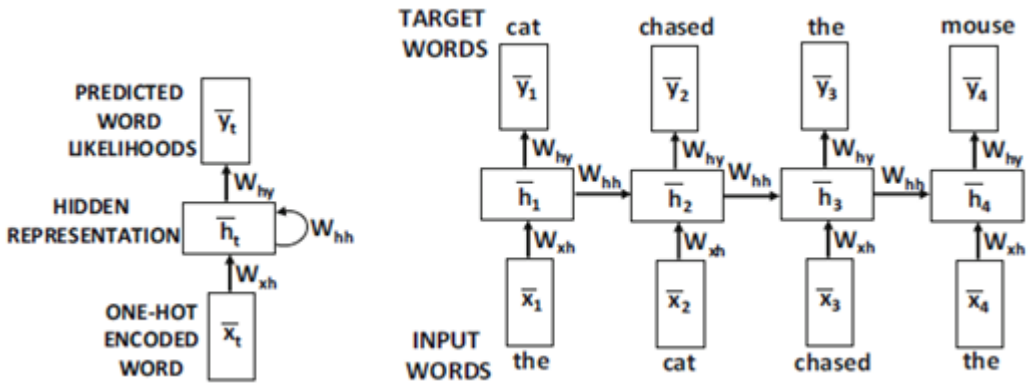

Figura 3.15: Esquematización de una RNN para la cadena de texto *"the cat chased the mouse"* [2]. Del lado izquierdo, el esquema general de la RNN y del lado derecho con la incorporación del esquema de representación de las marcas de tiempo.

En esta arquitectura, las entradas pueden interactuar directamente con capas ocultas dependiendo de sus posiciones en la secuencia. Cada capa utiliza el mismo conjunto de parámetros para garantizar un modelado similar en cada marca de tiempo y, por lo tanto, el número de parámetros también es fijo. En otras palabras, la misma arquitectura de capas se repite en el tiempo y es por ello que la red se denomina recurrente [2].

No obstante, hay varios desafíos prácticos para la formación de una RNN que hacen necesario el diseño de diversas mejoras arquitectónicas a la RNN. Una de ellas es debido

al hecho de que la cantidad de capas genera una red muy profunda, especialmente si la secuencia de entrada es larga. En otras palabras, la profundidad de la estratificación temporal depende de la entrada.

Si bien excede el alcance de este trabajo estudiar en profundidad las redes neuronales recurrentes y su variante LSTM, éstas últimas abordan este problema cambiando la ecuación de recurrencia para el vector oculto con el uso del LSTM de memoria a largo plazo. Estas operaciones de LSTM están diseñadas para tener un control detallado sobre los datos escritos en esta memoria a largo plazo.

3.4.5.4 *BERT para clasificación de documentos*

El modelo BERT-base contiene un codificador con 12 *transformers* apilados, 12 *multi-head attention* y un tamaño del estado oculto de 768. BERT toma una entrada de una secuencia de no más de 512 tokens y genera la representación de la secuencia [1].

Como se abordó anteriormente, la secuencia tiene uno o dos segmentos, el primer token de la secuencia es siempre $[CLS]$ y existe otro token especial, $[SEP]$, que se utiliza para separar segmentos.

Para las tareas de clasificación de texto, BERT toma el estado oculto final h del primer token $[CLS]$ como la representación de toda la secuencia. Se agrega un clasificador softmax simple en la parte superior de la arquitectura de BERT para predecir la probabilidad de la Clase c:

$$p(c|h) = softmax(Wh) \tag{3.6}$$

donde W es la matriz de parámetros específicos de la tarea. Deben ajustarse todos los parámetros de BERT y W conjuntamente maximizando la probabilidad logarítmica de la etiqueta correcta [90].

3.4.6 *Estrategias de evaluación de modelos*

El rendimiento de generalización de un método de aprendizaje se relaciona con su capacidad de predicción sobre datos de prueba independientes a los utilizados para entrenar el modelo [36]. La evaluación de este desempeño es extremadamente importante en la práctica, ya que guía la elección de la técnica o modelo de aprendizaje y provee una medida de la calidad del modelo finalmente elegido.

Normalmente, el aprendizaje automático implica mucha experimentación, por ejemplo, para el ajuste de los hiperparámetros que forman parte la técnica con la cual se trabaja en determinado momento. La ejecución de un algoritmo de aprendizaje sobre un conjunto de datos de entrenamiento con diferentes configuraciones de hiperparámetros dará como resultado diferentes modelos. Dado que normalmente se busca seleccionar el modelo de mejor rendimiento de este conjunto, es necesario encontrar un mecanismo para estimar sus respectivos rendimientos [75].

En realidad, la mayoría de las veces, resulta necesario comparar diferentes algoritmos, a menudo en términos de rendimiento predictivo y computacional. Los puntos principales por los que se evalúa el rendimiento predictivo de un modelo se plantean en los siguientes tres puntos [75]:

1. Estimar el rendimiento de generalización; es decir, el rendimiento predictivo de un modelo sobre datos futuros (no conocidos).

2. Aumentar el rendimiento predictivo ajustando el algoritmo de aprendizaje y seleccionando el modelo de mejor rendimiento sobre un espacio de hipótesis, o de búsqueda, determinado.

3. Identificar el algoritmo de aprendizaje automático que mejor se adapte al problema en cuestión a partir de la comparación entre diferentes algoritmos.

A partir de lo anterior, es importante tener en cuenta que pueden clasificarse en dos objetivos diferentes los que se persiguen en esta etapa del proceso [36]:

- **Selección de modelos:** estimar el rendimiento de diferentes modelos para elegir el mejor.

- **Evaluación del modelo:** habiendo elegido un modelo final, estimando su error de predicción (error de generalización) sobre nuevos datos.

En contextos en los cuales hay abundancia de datos, el mejor enfoque para ambos problemas es dividir aleatoriamente el conjunto de datos en tres partes: un conjunto de entrenamiento, un conjunto de validación y un conjunto de prueba. El conjunto de entrenamiento se utiliza para ajustar los modelos; el conjunto de validación se utiliza para estimar el error de predicción para la selección del modelo y el conjunto de prueba se utiliza para evaluar el error de generalización del modelo final elegido. En estos enfoques, idealmente, el conjunto de prueba debe guardarse en una "bóveda" y sacarse sólo al final del análisis de datos.

Es difícil dar una regla general sobre cómo elegir el número de observaciones en cada una de las tres partes; sin embargo, una división típica podría ser 50 % para entrenamiento, 25 % para validación y 25 % para evaluación [36].

No obstante, en escenarios donde la cantidad de datos disponibles son escasos, no es posible dividirlos en tres partes; en esos casos se utilizan dos conjuntos de datos: el primero para el entrenamiento y validación de los modelos y el segundo para la evaluación. A continuación, abordamos con algo más de profundidad tres de las estrategias más utilizadas en estas situaciones: el método de *holdout*, el método de *bootstrap* y la validación cruzada.

3.4.6.1 *Método del Holdout [75]*

El método de *Holdout* es indiscutiblemente la técnica de evaluación de modelos más simple. A grandes rasgos, consiste en tomar el conjunto de datos etiquetado y dividirlo en dos partes: un conjunto de entrenamiento y uno de prueba. A partir los datos de entrenamiento se ajusta el modelo y se predicen las etiquetas del conjunto de prueba. La fracción de

predicciones correctas se puede calcular comparando las etiquetas predichas con respecto a las etiquetas del conjunto de prueba.

Normalmente, la división de un conjunto de datos en conjuntos de entrenamiento y de prueba es un proceso simple de submuestreo aleatorio. Se supone que todos los datos se han extraído de la misma distribución de probabilidad (con respecto a cada clase) y se elige al azar 2/3 de estas muestras para el conjunto de entrenamiento y 1/3 de las muestras para el conjunto de prueba.

A continuación, se profundiza sobre esta idea a partir de la caracterización definida por algunos autores [75] a partir de la separación del proceso en diferentes pasos, los cuales además se esquematizan en diferentes Figuras.

PASO 1. En primer lugar, se dividen aleatoriamente los datos disponibles en dos subconjuntos: uno para entrenamiento y otro para prueba.

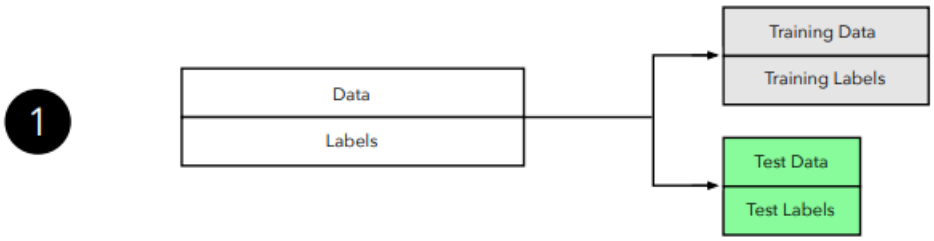

Figura 3.16: Paso 1 de la estrategia de *Holdout* [75]

Dejar de lado los datos de prueba es una solución alternativa para lidiar con las imperfecciones de un mundo no ideal, como datos y recursos limitados, y la incapacidad de recopilar más datos de la distribución generadora. Aquí, el conjunto de prueba representará datos nuevos no vistos para el modelo. Es importante que el conjunto de prueba se use solo una vez para evitar introducir sesgos cuando se estima el desempeño. Normalmente, se asignan 2/3 de los datos al conjunto de entrenamiento y 1/3 de los datos al conjunto de prueba. Otras divisiones comunes de entrenamiento y prueba son 60/40, 70/30 u 80/20, o incluso 90/10 si el conjunto de datos es relativamente grande.

PASO 2. Después de dejar de lado los ejemplos de prueba, se opta por un algoritmo de aprendizaje que podría ser apropiado para el problema dado y se especifican valores de hiperparámetros manualmente; el algoritmo de aprendizaje no los aprende de los datos de entrenamiento en contraste con los parámetros reales del modelo. Dado que los hiperparámetros no se aprenden durante el ajuste del modelo, es necesario algún tipo de "procedimiento adicional" o "bucle externo" para optimizarlos por separado.

PASO 3. Después de ajustar un modelo a partir del algoritmo de aprendizaje, debe indagarse respecto a la calidad del rendimiento. Aquí es donde entra en juego el conjunto de prueba independiente. Dado que el algoritmo de aprendizaje no conoce este conjunto de

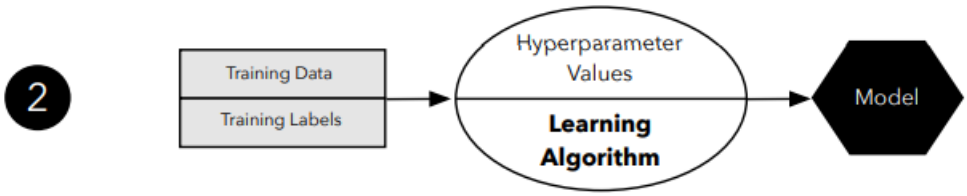

Figura 3.17: Paso 2 de la estrategia de *Holdout* [75]

pruebas, debería proporcionar una estimación relativamente imparcial de su rendimiento en datos nuevos no vistos. Entonces, se utiliza el conjunto de datos de prueba sobre el modelo ajustado para predecir las etiquetas de clase y luego se comparan respecto de las etiquetas de las instancias de prueba -es decir las etiquetas de clase correctas- para estimar la exactitud o el error de generalización del modelo.

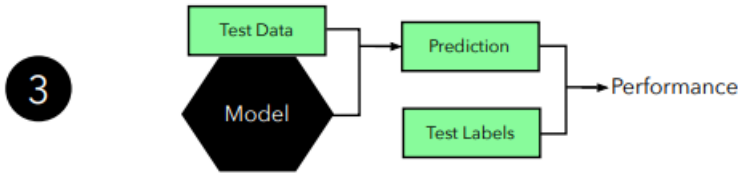

Figura 3.18: Paso 3 de la estrategia de *Holdout* [75]

PASO 4. Finalmente, se obtiene una estimación de cuan bien funciona el modelo con esos datos desconocidos. Dado que se asume que las muestras son I.I.D. (independientes e idénticamente distribuidas), no hay razón para suponer que el modelo funcionaría peor después de alimentarlo con todos los datos disponibles o del mundo real. Como regla general, el modelo tendrá un mejor rendimiento de generalización cuando los algoritmos utilizan más datos durante el entrenamiento, asumiendo que aún no ha alcanzado su capacidad.

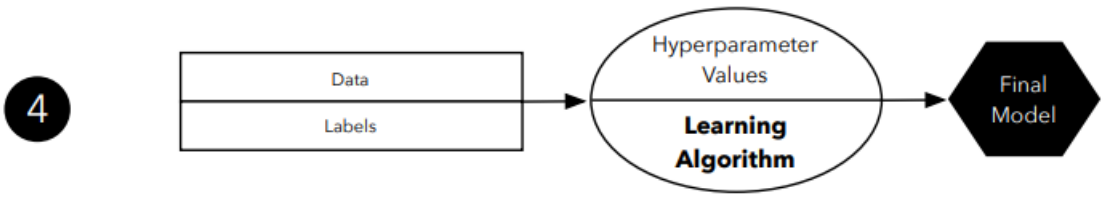

Figura 3.19: Paso 4 de la estrategia de *Holdout* [75]

Esta estrategia de evaluación de modelos tiene la ventaja de ser simple de implementar. Por otro lado, existen dos problemas principales cuando un conjunto de datos se divide en conjuntos separados de entrenamiento y prueba. El primer problema que ocurre es la potencial violación de la independencia y las proporciones cambiantes de clase en el submuestreo entre los dos subconjuntos resultantes. El segundo problema tiene que ver con que en muchas situaciones durante el entrenamiento, el modelo no ha alcanzado su máxima capacidad por lo cual la estimación de rendimiento estaría sesgada de forma pesimista.

3.4.6.2 *Método de Bootstrap*

La idea del método bootstrap es generar nuevos datos de entrenamiento a partir de una población mediante el muestreo repetido del conjunto de datos original con reemplazo.

Figura 3.20: Esquema de la separación de datos en *Bootstrap* [75]

El método de bootstrap, con algo más de detalle, consta de los siguientes cuatro pasos:

1. Se toma un conjunto de datos de tamaño n.
2. Durante b iteraciones de bootstrap:
   - Se extrae una sola instancia de este conjunto de datos y se asigna a la j-ésima muestra de bootstrap.
   - Se repite esto hasta que la muestra de arranque tenga un tamaño n (el tamaño del conjunto de datos original).
   - Cada vez, se extraen muestras del mismo conjunto de datos original, de modo que ciertos ejemplos pueden aparecer más de una vez en una muestra de arranque y otros no.
3. Se ajusta un modelo a cada una de las b muestras de bootstrap y se calcula la exactitud de cada muestra.
4. Se calcula la exactitud del modelo como el promedio sobre las estimaciones de exactitud b.

3.4.6.3 *Validación cruzada*

En esta sección se introducen las nociones básicas de la técnica probablemente más utilizada para la evaluación y selección de modelos en la práctica del aprendizaje automático: la validación cruzada de *k-fold*.

La idea principal de esta estrategia es que en cada iteración se divide el conjunto de datos en k partes: una parte se usa para la validación, y las k - 1 partes restantes se fusionan en un subconjunto de entrenamiento para la evaluación del modelo.

En esta estrategia, se utilizan hiperparámetros fijos para ajustar los modelos a los pliegues de entrenamiento en cada iteración; entonces, en una validación cruzada de 5 veces, este procedimiento dará como resultado cinco modelos diferentes ajustados. Estos modelos se ajustaron a conjuntos de entrenamiento distintos pero parcialmente superpuestos y se validaron en conjuntos de validación no superpuestos. Finalmente, se calcula el rendimiento de la validación cruzada como la media aritmética sobre las k estimaciones de rendimiento de los conjuntos de validación.

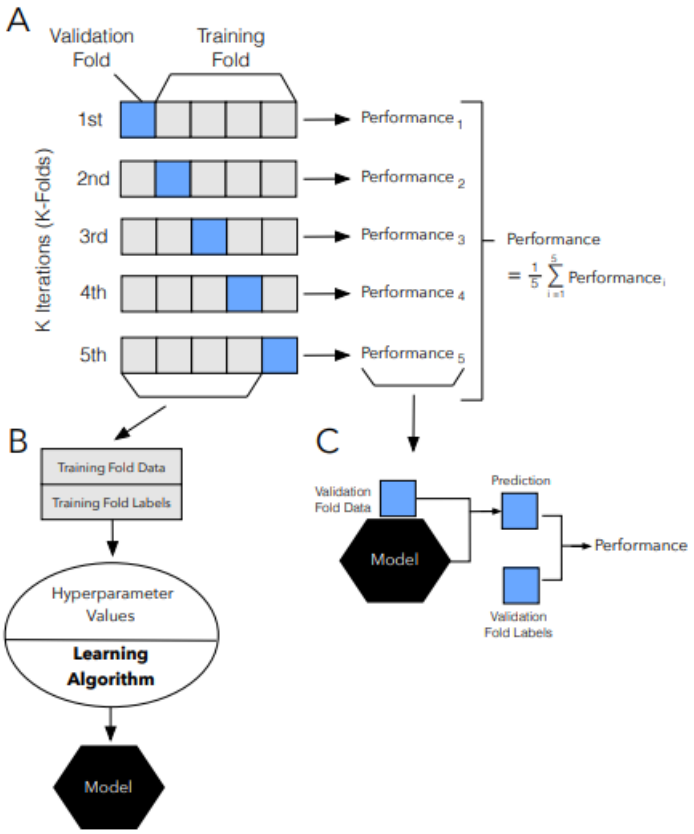

Figura 3.21: Proceso de validación cruzada para *5-fold validation* [75]

La idea detrás de este enfoque es reducir el sesgo pesimista mediante el uso de más datos de entrenamiento en contraste con dejar de lado una porción relativamente grande del conjunto de datos como datos de prueba. Y en contraste con el *método de holdout*, las *fold* de prueba en la validación cruzada de *k-folds* no se superponen. Además, la validación

cruzada de k veces garantiza que cada muestra se utilice para la validación en contraste con *holdout*, donde algunas muestras puede que nunca sean utilizadas.

3.4.7 *Métricas de selección de modelos*

Además de definir una estrategia de evaluación, para cuantificar la performance de un modelo, así como para comparar diferentes modelos, es necesario calcular una métrica cuantitativa. En este apartado, se introducen un conjunto de métricas de selección de modelos ampliamente utilizadas en la disciplina y que luego se aplicarán en el desarrollo experimental. Las métricas en cuestión son *accuracy* [42], *precision*, *recall*, *f1-score* y *Matthews correlation coefficient* o simplemente *MCC* [22] [35] [74].

3.4.7.1 *Matríz de confusión*

La matriz de confusión es una herramienta útil para analizar qué tan bien un clasificador puede reconocer instancias de las diferentes clases. Además, la mayoría de las métricas de selección de modelos utilizan la matriz de confusión para el cálculo [75]. En la Figura 3.22 se presenta una matriz de confusión de dos clases.

		Clase predicha	
		C_1	C_2
Clase observada	C_1	verdaderos positivos	falsos negativos
	C_2	falsos positivos	verdaderos negativos

Figura 3.22: Matriz de confusión para un problema de clasificación binario

Formalmente, dadas m clases, una matriz de confusión CM es una tabla de tamaño $m \times m$. La entrada $CM_{i,j}$ indica el número de instancias de clase i que fueron etiquetadas por el clasificador como clase j. Para que un clasificador tenga un buen rendimiento, idealmente la mayoría de las instancias deben estar representadas a lo largo de la diagonal de la matriz de confusión, desde la entrada $CM_{1,1}$ hasta la entrada $CM_{m,m}$, y el resto de las entradas de la tabla cercanas a cero. La tabla puede tener filas o columnas adicionales para proporcionar totales o tasas de reconocimiento por clase [42].

A su vez, se desprenden un conjunto de conceptos de la intersección de las clases predichas y observadas que resultan importantes para el análisis de costos y beneficios (o riesgos y ganancias) asociados con un modelo de clasificación: verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos.

En muchos casos, el costo asociado con un falso negativo (como, por ejemplo, predecir incorrectamente que un paciente con cáncer no tiene cáncer) es mucho mayor que el de un falso positivo (etiquetar de manera incorrecta pero conservadora a un paciente no canceroso como canceroso). De manera similar, los beneficios asociados con un verdadero positivo pueden ser superiores, generalmente, a los de un verdadero negativo. En tales

casos, es posible ponderar un tipo de error sobre otro asignando un costo diferente a cada uno [42].

3.4.7.2 *Exactitud (Accuracy)*

Luego del entrenamiento de uno o varios modelos, resulta necesario realizar comparaciones en torno a la performance de los mismos Una de las métricas clásicas de selección de modelos, y quizás la más utilizada, es el *accuracy* o, en español, la exactitud del modelo.

La exactitud de un clasificador en un conjunto de prueba determinado es el porcentaje de instancias del conjunto de prueba que el clasificador clasifica correctamente.

$$accuracy(y, \hat{y}) = \frac{1}{n_{\text{instancias}}} \sum_{i=1}^{n_{\text{instancias}}} (y_i = \hat{y}_i) \tag{3.7}$$

De la fórmula anterior, se deduce que la exactitud, o *accuracy*, es la cantidad de instancias para las cuales la clase observada y es igual a la clase predicha por el modelo $\hat{y}$ sobre el total de las $n_{\text{instancias}}$.

En la literatura de reconocimiento de patrones, esto también se conoce como la tasa de reconocimiento general del clasificador, es decir, refleja qué tan bien el clasificador reconoce las instancias de las distintas clases. También se deriva la tasa de error o tasa de clasificación errónea de un clasificador M, que es simplemente $1 - accuracy(M)$ [42].

3.4.7.3 *Precisión (Precision)*

Como se anticipó en una sección anterior, el método de cálculo de muchas de las métricas de selección de modelos parte de la matriz de confusión, tal es el caso de la precisión.

$$precision = \frac{verdaderos\_positivos}{verdaderos\_positivos + falsos\_positivos} \tag{3.8}$$

La precisión es una métrica que surge en la disciplina de la recuperación de información y mide la capacidad del modelo de clasificar correctamente las instancias de una determinada clase; es decir, la proporción de instancias clasificadas como clase C_i que realmente corresponden a esa clase [24] [92].

Esta métrica, habitualmente, es evaluada en conjunto con la métrica de *recall* o exhaustividad.

3.4.7.4 *Exhaustividad (Recall)*

Por su parte, como complemento de la precisión aparece la métrica de *recall* o exhaustividad.

$$recall = \frac{verdaderos\_positivos}{verdaderos\_positivos + falsos\_negativos} \tag{3.9}$$

El *recall* de un modelo cuantifica la capacidad del mismo de clasificar instancias de la clase C_i; esto es, mide la proporción de instancias clasificadas como clase C_i respecto al total de instancias existentes en el lote de pruebas de esa clase. Esta métrica permite evaluar la habilidad del modelo para encontrar todas las instancias relevantes de cada clase [92].

3.4.7.5 *Valor-F (F-Score)*

La métrica *F-Score* surge de tomar el promedio armónico (ponderado) de las dos métricas precedentes (precisión y exhaustividad) [38]:

$$F_\beta = \frac{(1+\beta^2)Precision}{(1+\beta^2)Precision + Recall} \tag{3.10}$$

Esta métrica surge con el objetivo de combinar las métricas de precisión y exhaustividad, introducidas anteriormente. A su vez, el parámetro, $\beta \in [0, \infty)$, permite ponderar la importancia relativa de cada uno de estos criterios:

- Si $\beta > 1$ entonces se le da más peso a la exhaustividad.
- En cambio, si $\beta < 1$, entonces se asigna más peso a la precisión.

En muchas situaciones, no se desea ponderar un criterio de selección por encima del otro, y -como en este trabajo- se utiliza el valor neutral del parámetro, $\beta = 1$. Esta variante se conoce como *F1-score* [55].

3.4.7.6 *Coeficiente de correlación de Matthews -MCC- (Matthews correlation coefficient)*

Habitualmente se cree que la métrica de rendimiento más razonable es el *accuracy*, es decir, la relación entre el número de muestras clasificadas correctamente y el número total de muestras [22]. Sin embargo, cuando el conjunto de datos está desbalanceado (la cantidad de muestras en una clase es mucho mayor que la cantidad de muestras en las otras clases), el *accuracy* ya no se puede considerar una medida confiable porque proporciona una estimación demasiado optimista de la capacidad del clasificador en la clase mayoritaria [22].

Una solución eficaz para superar el problema del desbalanceo de clases proviene del coeficiente de correlación de Matthews (MCC) [53]. El coeficiente de correlación de Matthews es un método basado en la matriz de confusión para calcular el coeficiente de correlación Pearson entre los valores reales de las clases y los predichos por el modelo. La fórmula se transcribe a continuación:

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp+fp)(tp+fn)(tn+fp)(tn+fn)}}. \tag{3.11}$$

En la fórmula anterior, tp corresponde a los verdaderos positivos, tn a los verdaderos negativos, fp a los falsos positivos y fn a los falsos negativos.

Existen investigaciones que abordan el rendimiento del coeficiente de correlación de Matthews en problemas de clasificación multiclase [53]. En ellos se demuestra, tanto analítica como empíricamente, que tiene un comportamiento consistente en casos prácticos. El MCC mejora el rendimiento respecto del *accuracy*, dado que este último maneja mal los problemas con clases desbalanceadas y no puede distinguir entre diferentes distribuciones de clasificación erróneas.

En general, MCC muestra un buen rendimiento en problemas con un número variable de clases, conjuntos de datos desbalanceados y aleatorización. Además, el comportamiento de MCC sigue siendo coherente en configuraciones tanto binarias como multiclase [53].

3.4.8 *Utilización del modelo*

El proyecto CRISP-DM (*Cross Industry Standard Process for Data Mining*) definió un modelo de proceso que proporciona un marco para llevar a cabo proyectos de minería de datos, el cual es independiente tanto del sector industrial como de la tecnología utilizada. El modelo de proceso CRISP-DM tiene como objetivo hacer que los grandes proyectos de minería de datos sean menos costosos, más confiables, más repetibles, más manejables y más rápidos. Este modelo de proceso, incorpora como última actividad al *deployment* o implementación del conocimiento obtenido [100].

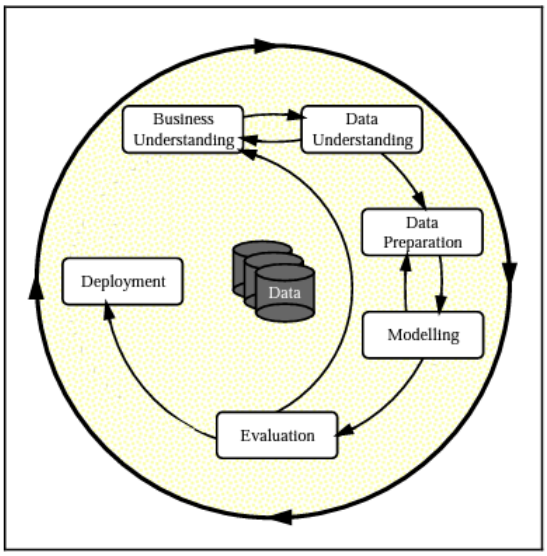

Figura 3.23: Fases del modelo de proceso CRISP-DM [100]

En este sentido, CRISP-DM plantea en la etapa de *deployment* que la creación del modelo generalmente no es el final del proyecto. Por lo general, el conocimiento adquirido debe organizarse y presentarse de manera que el cliente pueda utilizarlo. Dependiendo de los requisitos, la fase de implementación puede ser tan simple como generar un informe o tan compleja como implementar un proceso de minería de datos repetible. En muchos casos será el usuario, no el analista de datos, quien llevará a cabo los pasos de implementación. En cualquier caso, es importante comprender de antemano qué acciones deberán llevarse a cabo para poder hacer uso efectivo de los modelos creados [100].

Esta actividad además es abordada por el proceso de descubrimiento de conocimiento (KDD) en su novena y última etapa, la cual consiste en actuar sobre el conocimiento descubierto: usar el conocimiento directamente, incorporar el conocimiento en otro sistema para acciones posteriores, o simplemente documentarlo y reportarlo a las partes interesadas. Este proceso también incluye la verificación y resolución de posibles conflictos con el conocimiento previo (o extraído) [27].

4

CLASIFICACIÓN SEMI-SUPERVISADA

4.1 INTRODUCCIÓN

Sumado a las problemáticas planteadas, resulta evidente que, de acuerdo al proceso tradicional para la construcción de un clasificador automático de texto, una de las primeras tareas que se deben llevar a cabo es la clasificación inicial de un conjunto de documentos que luego serán utilizados como conjuntos de entrenamiento y prueba para el entrenamiento, validación y evaluación del clasificador.

La estrategia tradicional para el etiquetado de documentos consiste en que esta tarea sea realizada por una persona, de forma manual. En muchas ocasiones, este etiquetado manual debe ser realizado por expertos en el tema que forman parte del problema que se desea abordar. Si bien estas etiquetas de expertos proporcionan la piedra angular tradicional para evaluar los modelos de aprendizaje automático, el acceso limitado o costoso a ellos representa un cuello de botella [52].

A su vez, para caracterizar con precisión la efectividad de un sistema, la experiencia ha demostrado que deben evaluarse a la escala operativa en la que se utilizarán en la práctica, lo cual resulta en una limitación para esta metodología puesto que, debido a que los tamaños de las colecciones han crecido rápidamente en los últimos años, se ha vuelto cada vez menos factible etiquetar manualmente tantos ejemplos usando el etiquetado experto tradicional [52].

4.2 ANTECEDENTES

En las aplicaciones de aprendizaje automático de escala real, a menudo se da el caso en que se encuentran disponibles abundantes ejemplos de entrenamiento sin etiquetar. Sin embargo, los ejemplos etiquetados son bastante costosos de obtener ya que requieren mucho esfuerzo humano. Como consecuencia de ello, y el crecimiento de la cantidad de datos disponibles, el aprendizaje semi-supervisado ha generado mucha atención [105].

Como el nombre sugiere, el concepto de aprendizaje semi-supervisado se encuentra entre el aprendizaje supervisado y no supervisado; de hecho, la mayoría de las estrategias de aprendizaje semi-supervisado se basan en extender el aprendizaje supervisado o no supervisado para incluir información adicional típica del otro paradigma de aprendizaje. Específicamente, el aprendizaje semi-supervisado abarca varias áreas diferentes, que incluyen la clasificación semi-supervisada [105].

Formalmente, dado un conjunto de datos etiquetados $D_l = \{(x_i, y_i)|(x_i, y_i) \in X \times Y, i = 1, \ldots, l\}$, y un conjunto de datos no etiquetados $D_u = \{x_j|x_j \in X, j = l+1, \ldots, l+u\}$, donde X comprende el espacio de características de las instancias e Y las etiquetas o clases, un algoritmo semi-supervisado tiene como objetivo entrenar un clasificador f a partir de $D_l \cup D_u$, es decir de los datos etiquetados y no etiquetados, de modo tal que resulte mejor que el clasificador supervisado entrenado solo con los datos etiquetados [84, 105].

En este contexto, se han estudiado y desarrollado diferentes estrategias para la clasificación semi-supervisada de documentos con el objeto de aportar mayor escalabilidad.

En algunos sistemas de uso masivo, una estrategia posible es inferir etiquetas implícitas del comportamiento de las personas que lo utilizan, aunque para su consolidación requiere grandes poblaciones de usuarios, como en el caso de los buscadores de internet [49].

Un enfoque alternativo de etiquetado de datos consiste en la "supervisión distante", en la que los datos de entrenamiento son etiquetados a partir de algunas características del texto, como *tags*, emoticones y otros metadatos [37]. Este enfoque es particularmente interesante, y se han encontrado buenos resultados en redes sociales en las cuales los emoticones pueden ser indicadores del sentimiento del usuario dado que se ha demostrado que estos símbolos poseen la ventaja de ser independientes del dominio, del tema y del tiempo [76].

Otro de los abordajes para el etiquetado de documentos consiste en el utilizado para el tratamiento de problemas del tipo *"PU-learning"*, denominado así por el acrónimo en inglés de "aprendizaje a partir de ejemplos positivos y no etiquetados"(*Learning from Positive and Unlabelled examples*) [33]. Estos problemas también se conocen como clasificaciones de una clase. En este tipo de problemas, existe un conjunto de documentos de un determinado tema en particular o clase P (clase positiva), que se encuentran complementados por un subconjunto de documentos mixtos, los cuales no corresponden a una clase específica. La característica clave de este problema es que no hay datos de entrenamiento negativos etiquetados, lo cual hace que las técnicas tradicionales de clasificación de texto sean inaplicables. Un abordaje explorado para este tipo de problemas consiste en identificar manualmente un conjunto de documentos negativos fiables del conjunto sin etiquetar y luego, a partir de la construcción de un conjunto de clasificadores de forma iterativa, etiquetar nuevos ejemplos negativos identificados por esos clasificadores [32, 33], lo cual comprende una estrategia semi-supervisada de etiquetado de documentos.

En cuanto a los enfoques semi-automáticos para problemas multiclase, un enfoque que ha mostrado buenos resultados consiste en etiquetar un conjunto de palabras, representativas de cada clase, para luego etiquetar automáticamente un conjunto de documentos, que se utilizarán para el entrenamiento del clasificador, en función de la presencia de esas palabras representativas. La clave para el funcionamiento de este enfoque es elegir un conjunto de palabras para cada clase que sea realmente representativo de la misma [65].

Más aquí en el tiempo, se pueden observar trabajos que utilizan la extracción de características y el agrupamiento mediante *K-Means* para el etiquetado semi-supervisado de correos electrónicos [5, 40, 66]. Si bien estos trabajos abordan el análisis de sentimiento, el cual enfrenta desafíos diferentes que la clasificación multi-etiqueta abordada en este trabajo, existen paralelismos en los procesos desarrollados en cuanto al preprocesamiento del

corpus de documentos y las estrategias de extracción de características utilizadas para la representación de los sentimientos.

Sin embargo, algunos aspectos de los antecedentes aquí introducidos hacen particular al trabajo en cuestión. El aspecto fundamental reside en que en la mayoría de los trabajos anteriores se aborda el problema conocido como *análisis de sentimiento*, donde las clases poseen generalmente dos estados, haciendo viable la posibilidad de intensificar la labor para predecir uno de ellos y determinar el otro a partir del complemento del primero. Este abordaje permite, habitualmente a través de la utilización de técnicas de *clustering*, encontrar similitudes entre documentos de una misma clase. En cambio, en el marco de este trabajo, se aborda el desafío del etiquetado semi-supervisado multiclase, donde la selección de características, y etiquetado posterior a partir de ellas, se realiza en un caso de estudio basado en correos electrónicos en idioma español que consta de 16 clases diferentes.

La hipótesis acuñada para la estrategia propuesta es que la selección de características representativas de cada clase, combinada con enfoques de recuperación de la información, constituyen un método semi-supervisado válido y sencillo para la clasificación automática de correos electrónicos. En este contexto, se busca verificar que los resultados obtenidos con los modelos entrenados mediante este enfoque permitan mejorar aquellos que se obtienen sólo con datos etiquetados manualmente.

4.3 ESTRATEGIA SEMI-SUPERVISADA PROPUESTA

Como se abordó anteriormente, el objetivo general de este capítulo consiste en generar un proceso para la clasificación semi-supervisada de correos electrónicos a partir de la identificación de características claves de cada clase, utilizando técnicas de selección de características y la posterior recuperación de correos automáticamente a partir de un enfoque de recuperación de información.

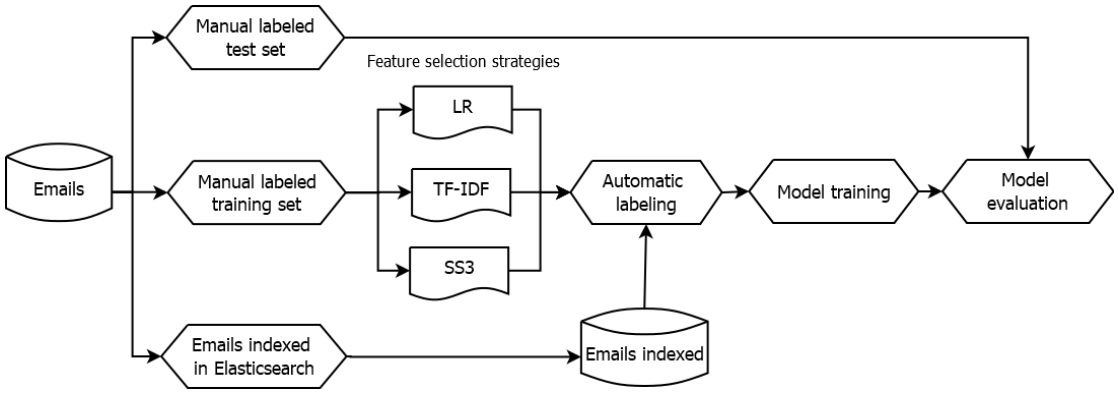

Figura 4.1: Flujo de trabajo para el etiquetado semi-supervisado de correos electrónicos

En la Figura 4.1 se ilustra el flujo de trabajo desarrollado en el marco de esta estrategia. La idea fuerza de la misma es partir de un conjunto de correos electrónicos multi-clase etiquetados de forma tradicional y realizar una extracción de las características principales para cada clase utilizando tres técnicas como regresión logística, TF-IDF y SS3.

Luego, con la base de conocimiento completa indexada en un motor de búsqueda de propósito general como *Elasticsearch*, recuperar los documentos de cada clase en función de las características detectadas por cada técnica y construir un clasificador, el cual se evalúa en función de un conjunto de datos de prueba reservado inicialmente.

4.3.1 *Conjunto de datos inicial de Correos electrónicos*

El flujo de trabajo para la estrategia propuesta inicia a partir de la disponibilidad de una base de conocimiento conformada por correos electrónicos, la cual se constituye como el conjunto de datos. De este conjunto de datos, se realiza un muestreo de un subconjunto de los correos y se etiquetan manualmente a partir de la estrategia convencional, mediante un experto del dominio. A su vez, esa muestra etiquetada manualmente se separa en dos conjuntos de datos, uno para entrenamiento y validación y el otro para evaluación, normalmente en una proporción de 80 % y 20 % respectivamente.

4.3.2 *Indexación de correos electrónicos con Elasticsearch*

Elasticsearch es un motor de búsqueda y análisis distribuido, gratuito y abierto que permite almacenar e indexar múltiples tipos de datos, incluidos textuales, numéricos, geoespaciales, estructurados y no estructurados[1]. A su vez, *Elasticsearch* soporta textos en 34 idiomas distintos y provee analizadores para cada uno. Los analizadores están compuestos por una cadena de filtros que ejecutan transformaciones sobre los textos a indexar, de forma que las transformaciones que realiza cada analizador dependen de los filtros que utiliza.

A los efectos de esta propuesta, se indexa la totalidad de los correos en una instancia de *Elasticsearch* y se aplica el analizador estándar para el idioma español, de acuerdo a los correos disponibles.

4.3.3 *Estrategias de selección de características*

Previo a la aplicación de las estrategias de selección de características, se aplican técnicas de preprocesamiento: se normaliza el texto, se eliminan palabras vacías y opcionalmente se generan atributos estáticos (como el largo de la consulta y la utilización de signos de puntuación, etc) para nutrir los generados a partir de los términos.

Como estrategias de extracción de características, en este trabajo se proponen la ponderación TF-IDF agrupada por clases, la valoración de palabras obtenida con el modelo para clasificación Sequential S3 (Smoothness, Significance, and Sanction) o simplemente SS3 y los coeficientes de las funciones de clasificación de la regresión logística para cada clase. A continuación, se realiza una breve sinapsis de las tres técnicas planteadas.

1 Extraido de https://www.elastic.co/es/what-is/elasticsearch

4.3.3.1 *TF-IDF*

El modelo vectorial es uno de los métodos básicos para representar un documento, y uno de los más antiguos. Éste modelo se utiliza para formar un vector que representa un documento usando el recuento de frecuencia de cada término en el mismo o algún otro método de ponderación [44]. Esta ponderación o peso, se utiliza para enfatizar la importancia de las características -términos o palabras- de un documento (correo electrónico).

Uno de los métodos de ponderación de términos más conocidos es TF-IDF (frecuencia de término – frecuencia inversa de documento), el cual plantea la idea de establecer una relación entre la frecuencia de un término dentro de un documento y su frecuencia en los documentos de toda la colección [92].

Esta métrica, a su vez, se conforma a partir del cociente entre dos métricas: TF e IDF. TF (frecuencia de término) captura la importancia de un término para un documento, en este caso consiste en el número total de veces que un término aparece en un correo electrónico. Por su parte, IDF refleja la importancia de un término para un documento en un corpus de documentos [79, 91].

Formalmente, la fórmula del pesado TF-IDF se define a continuación [92]:

$$TF * IDF_{ij} = TF_{ij} \times \log_2 \frac{N}{n} \tag{4.1}$$

donde:
TF_{ij} corresponde a la frecuencia del término t_i en el documento d_j, la cual generalmente se normaliza por la longitud de d_j.
N es el tamaño de la colección de documentos.
n es la cantidad de documentos donde el término t_i está presente.

En efecto, esta relación permite que el valor de TF-IDF aumente proporcionalmente al número de veces que aparece un término en un documento, pero que este valor se compense con la frecuencia de ese término en el corpus, lo cual ayuda a controlar el hecho de que algunas palabras son generalmente más comunes que otras [7].

En el marco de esta investigación, se utiliza el promedio observado de esta ponderación agrupado por clase para determinar cuales son los términos más importantes para cada clase.

4.3.3.2 *SS3*

SS3 o *"Sequential S3"(Smoothness, Significance, and Sanction)* es un nuevo clasificador de texto, creado inicialmente para abordar los problemas de riesgo temprano, *early risk detection*, como depresión y otros trastornos psíquicos. En este sentido, este tipo de técnicas, deben tener en cuenta 3 requisitos claves: clasificación incremental, soporte para la clasificación temprana y explicabilidad [15].

Este clasificador asume que existe una función $gv(w, c)$ para valorar las palabras en relación con las categorías, de forma más específica, gv toma una palabra w y una categoría

c y genera un valor numérico en el intervalo [0, 1] que representa el grado de confianza con el que w pertenece exclusivamente a c. Aquí, $gv(w, c) = v$ se lee como "w tiene un valor global de v en c" o, alternativamente, "el valor global de w en c es v". Por ejemplo, $gv(apple, technology) = 0{,}8$ se lee como "apple tiene un valor global de 0,8 en tecnología".

Además, se define $gv(w) = (gv(w, c_0), gv(w, c_1), ..., gv(w, c_k))$ donde $c_i \in C$ y C denota el conjunto de todos las categorías. Es decir, cuando gv solo se aplica a una palabra, genera un vector en el que cada componente es el valor global de esa palabra para cada categoría c_i. El vector $gv(w) = \vec{v}$ se denominará "vector de confianza de w", siendo que cada clase c_i se asigna a una posición fija i en el vector de salida.

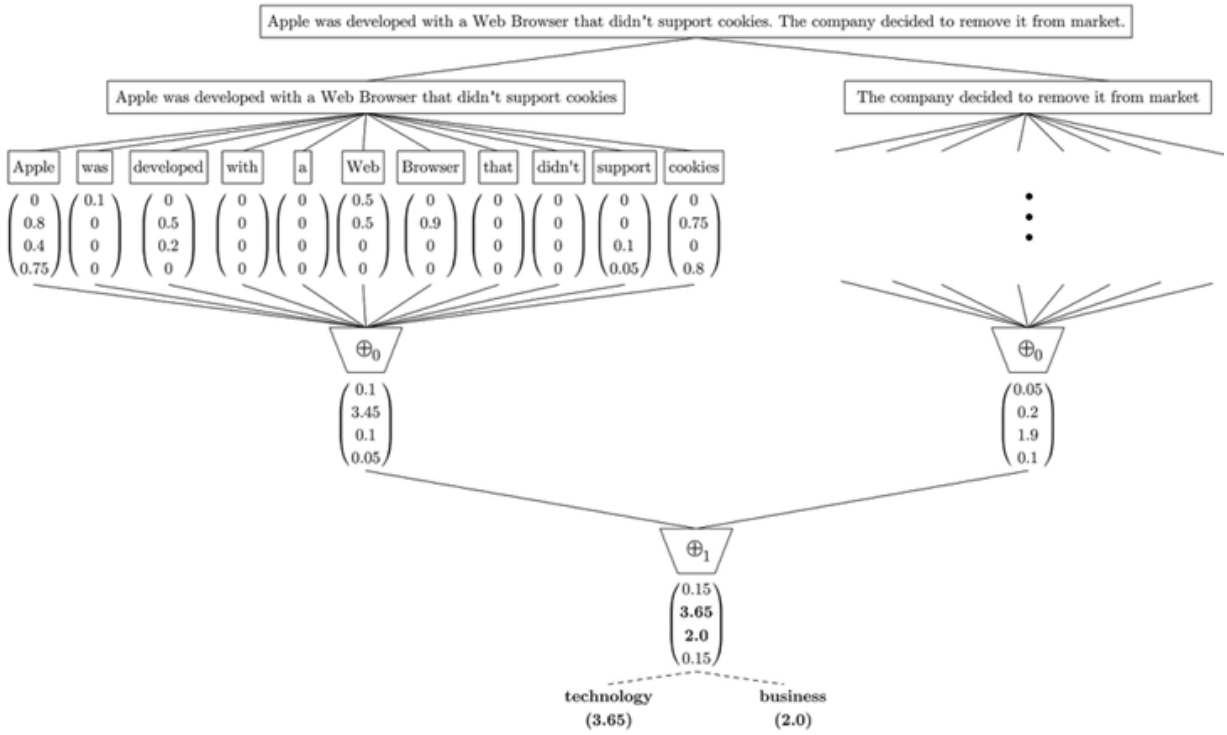

Figura 4.2: Proceso de clasificación para el documento "Apple was developed with a Web Browser that didn't support cookies. The company decided to remove it from the market" [15]

La clasificación se puede considerar como un proceso de dos fases. La primera fase comienza dividiendo la entrada dada (generalmente un solo documento) en varios bloques, luego cada bloque se divide repetidamente en unidades más pequeñas hasta que se alcanzan las palabras. Al final de esta fase, se habrá convertido la entrada previamente "plana" en una jerarquía de bloques; un documento se dividirá típicamente en párrafos, los párrafos en oraciones y las oraciones en palabras. Además, se genera una jerarquía, donde las palabras están en el nivel 0 en esta jerarquía, las oraciones en el nivel 1, los párrafos en el nivel 2, y así sucesivamente.

En la segunda fase, se aplica la función gv a cada palabra para obtener los vectores de confianza de nivel 0, que luego se reducen mediante un operador de resumen para generar los vectores de confianza del siguiente nivel. Este proceso de reducción se propaga de forma recursiva hacia los bloques de nivel superior, generando un único vector de confianza para toda la entrada. Por último, la clasificación real se realiza en función de

los valores de este vector de confianza único utilizando alguna política, por ejemplo la categoría con el valor máximo. Es importante señalar que las diferentes jerarquías podrían utilizar distintas políticas e incluso cualquier función de la forma $f : 2^{R^n} \rightarrow R^n$ podría usarse como operador de resumen.

En el marco de la estrategia de aprendizaje semi-supervisado propuesta en este capítulo, y en función de los conceptos introducidos, esta técnica permite ser utilizada para la extracción de características dado que, cómo se abordó antes, la función *gv(w, c)* valora las palabras en relación a categorías, asignando, para cada palabra w y categoría c, un número en el intervalo [0,1] que representa el grado de confianza con el que w pertenece exclusivamente a c.

4.3.3.3 *Regresión logística*

La regresión logística es un tipo de modelo de clasificación estadística probabilística. Se utiliza como modelo binario para predecir una respuesta binaria o el resultado de una variable dependiente de tipo categórica o discreta en función de una o más variables [47].

La fórmula de la regresión logística es la siguiente:

$$f(x) = \frac{1}{1 + e^{-\lambda x}} \tag{4.2}$$

Esta función es útil puesto que, para cualquier entrada numérica real, desde el infinito negativo hasta el infinito positivo, genera una salida restringida a valores entre 0 y 1 y, por lo tanto, puede interpretarse como una probabilidad.

Esta propiedad de la función generalmente se interpreta como la ocurrencia de un hecho cuando devuelve valores cercanos a 1 y la no ocurrencia con valores cercanos a 0 y por tanto resulta adecuada para la clasificación binaria.

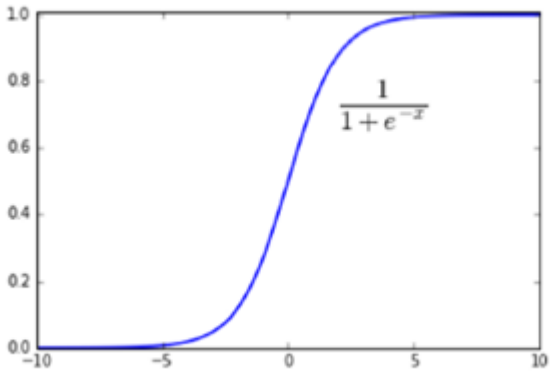

Figura 4.3: Gráfico de la función logit [85]

Asimismo, es posible extender la función inicial para que resulte de manera más efectiva para la clasificación, incluyendo, entre otras cuestiones, la posibilidad de extender la función $f(x)$ a una función $h(x, w)$ con un vector de entrada $x(m-1)$-dimensional [85]:

$$h(x, w) = w_0 + \sum_{i=1}^{m-1} w_i * x_i \tag{4.3}$$

Esta función puede complementarse con la función *logit* original para producir el clasificador múltiple de la siguiente manera:

$$f(x) = \frac{1}{1 + e^{-h(x,w)}} \tag{4.4}$$

En estos términos aún sigue siendo un clasificador binario, siendo que muchas veces -esta investigación es un caso- se requiere que los clasificadores puedan optar por más de dos etiquetas o clases. Intuitivamente la forma de abordar estos problemas es a partir de tratar de diferenciar una clase del resto sucesivas veces, una por cada clase presente en el problema como se observa en la Figura 4.4.

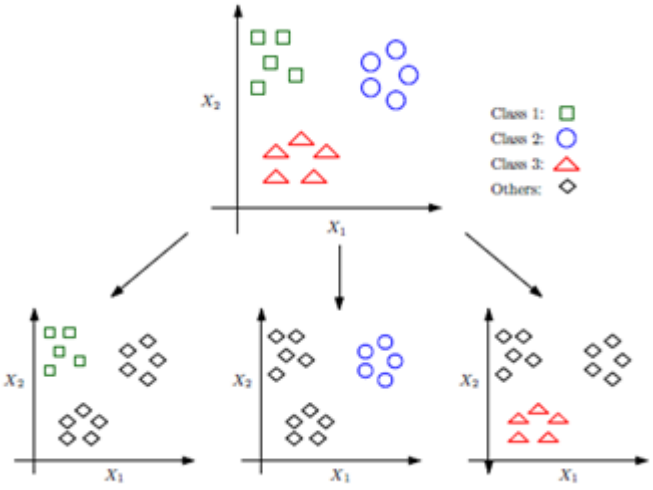

Figura 4.4: Tratamiento de problemas de clasificación multiclase [85]

En función de esta salvedad, se define una función diferente por cada clase, la cual separa esa clase del resto de las instancias. En este modelo, se utilizan los coeficientes generados para la función de cada clase como estrategia de selección de los términos más representativos para cada una de ellas.

La extracción de características se realiza a partir de un conjunto de datos de entrenamiento. Las restantes instancias, se reservan para la evaluación de los modelos. A partir de estas instancias de entrenamiento, se ejecutan las estrategias previamente presentadas.

En el caso de TF-IDF, se la propone como estrategia no supervisada de ponderación de términos, y se aplica en conjunto con una representación de documentos basada en el modelo vectorial, o bolsa de palabras [80].

Los términos se ponderan con TF-IDF[2] para cada término en cada documento, luego agrupando los términos por clase, quedando así una matríz con las clases como instancias y los términos (con su ponderación TF-IDF) como columnas. A efectos de seleccionar los términos más representativos para cada clase, se opta por seleccionar los N términos con mayor ponderación de TF-IDF promedio por clase.

Por su parte, se incorpora SS3 como una estrategia supervisada de extracción de características, puesto que esta técnica genera un vocabulario para cada clase con una ponderación realizada a partir de un valor de confianza. En este caso, nuevamente se ajusta un modelo de clasificación con SS3 para las instancias de entrenamiento y se obtienen los términos correspondientes al vocabulario de cada clase, utilizando los N mas representativos para la recuperación de correos.

Por último, el caso de la regresión logística se introduce debido a que, además de ser una estrategia supervisada, permite identificar, además de los términos representativos para cada clase, un conjunto de términos que son nocivos para la elección de un tópico determinado.

Para esta estrategia de extracción de características, se representan los documentos a partir del modelo vectorial, se ajusta un modelo en función de las instancias de entrenamiento y se seleccionan los $\frac{N}{2}$ términos con ponderaciones mas altas -en valor absoluto- tanto para los ejemplos positivos como para los negativos.

El valor N, que representa la cantidad de términos representativos a seleccionar para cada clase, debe ajustarse de forma empírica buscando que sean distintivos de cada clase y suficientes para capturar la mayor cantidad de instancias relacionadas con cada temática. A su vez, se debe ser equilibrado al establecer esta cantidad debido a que, al mismo tiempo que se incrementa el valor de N es natural que ciertos términos comiencen a aparecer de forma simultánea como término representativo de varias clases, lo cual no es aconsejable.

4.3.4 *Recuperación de correos electrónicos*

A partir de contar con los términos más representativos para cada clase -y su ponderación- según las tres técnicas de extracción de características abordadas, se recuperan los correos electrónicos indexados en el motor de búsqueda *Elasticsearch*, potencialmente a partir de dos diferentes estrategias.

Por un lado, es posible realizar una recuperación a partir de los términos para cada clase según su aparición o no, a modo de un *query* convencional. Por otro lado, se incorpora la ponderación de cada término según las distintas estrategias, lo que se denomina en el contexto de la herramienta como *boosting*.

2 En este trabajo se propone la fórmula de TF-IDF por defecto de la clase *TfidfVectorizer* de la librería *sklearn* para Python.

A su vez, para la estrategia basada en regresión logística, es posible realizar un *boosting negativo* para los términos con coeficiente menor a cero, penalizando a los documentos en que aparecen.

Es importante hacer notar que el motor de búsqueda, además de la posibilidad de limitar la cantidad de resultados, provee un *score* para magnificar el nivel de similitud de la búsqueda en relación a cada documento recuperado.

4.3.5 *Construcción del Modelo de clasificación*

Para la etapa de construcción del modelo de clasificación, en primer lugar se utiliza como conjunto de entrenamiento las instancias recuperadas para cada clase a partir de *Elasticsearch* en función de los términos generados por cada una de las técnicas de extracción de características y luego se adicionaron las instancias etiquetadas manualmente. En todos los casos, se compara la performance de los modelos generados a partir de la presencia de documentos etiquetados de forma automática con respecto a los modelos generados únicamente a partir de las instancias etiquetadas manualmente.

En cuanto a los datos utilizados para la etapa de entrenamiento, pueden diseñarse diferentes experiencias en función de la combinación de las instancias recuperadas mediante las estrategias de extracción de características y las etiquetadas manualmente.

Para la validación de los modelos, se deben reservar las instancias restantes de las etiquetadas manualmente. Por último, el análisis de selección de los modelos generados se realiza en función de métricas de selección de modelos tales como el *accuracy*, *precision*, *f1-score* y *MCC* o *Matthews correlation coefficient* [22].

5

TRABAJOS EXPERIMENTALES

En este capítulo se exponen las diferentes etapas de experimentación abordadas, las pruebas y los resultados obtenidos. Todos los recursos utilizados y productos generados se encuentran en el repositorio público de código de esta investigación, alojado en Github:

https://github.com/jumafernandez/clasificacion_correos/tree/main/tesis

En primera instancia, en la Sección 5.1, se describen las características de los correos electrónicos que conforman el conjunto de datos y se detallan las tareas de preprocesamiento y curado que se realizaron sobre los datos. A su vez, se aborda la actividad de etiquetado de datos y se presenta un análisis exploratorio general. Por último, se realiza una separación del conjunto de datos etiquetado en dos subconjuntos de datos, uno para entrenamiento y validación y otro para evaluación de los modelos. En la Sección 5.2, se aborda el entrenamiento de los modelos de clasificación en función de las técnicas elegidas, algunos ajustes y el análisis de los modelos obtenidos. Por último, en la Sección 5.3 se implementa el modelo de clasificación semi-supervisada propuesto en el Capítulo 4 y se indaga en las diferencias encontradas con respecto a los modelos alcanzados sin la adición de esta estrategia.

5.1 CONSOLIDACIÓN DEL CONJUNTO DE DATOS

Para la ejecución de los experimentos, se utilizó un conjunto de datos conformado por 24700 correos electrónicos generados a partir de consultas académicas realizadas por parte de estudiantes de la Universidad Nacional de Luján al staff administrativo sobre trámites derivados de su actividad académica.

5.1.1 *Origen de los correos electrónicos*

La Universidad Nacional de Luján (UNLu) cuenta con un sistema informático propio para llevar adelante la gestión académica de las actividades inherentes a la enseñanza de grado y pregrado, así como los trámites que de éstas se desprenden. Este sistema de gestión cuenta con una interfaz web a la que acceden los estudiantes para realizar todos los trámites relacionados a su vinculación con la Institución. A su vez, posee una funcionalidad para realizar consultas vía correo electrónico al staff administrativo.

El sistema, ante la formulación de una consulta por parte de los estudiantes envía, mediante un servidor SMTP, la consulta a una dirección de correo electrónico especialmente

destinada para este fin. Al cuerpo de ese correo, además del texto escrito por el estudiante, se agregan datos académicos y de la persona tales como nombre y apellido, legajo, documento, Carrera, teléfono y su email personal.

Figura 5.1: Captura de pantalla de la opción de envío de Consultas del Módulo Web

Para esta investigación, se utilizó una muestra sin clasificar de 24700 correos con consultas y sus respectivas respuestas que llegaron durante un lapso de tiempo continuo y que estaban almacenados en un archivo PST (*Personal Storage Table*)[1].

Figura 5.2: Ejemplo de correo electrónico de consulta UNLu

Cabe aclarar que para todo el proceso se utilizaron los correos de consulta originales sin supervisión humana sobre errores semánticos ni de sintaxis, si bien se incorporaron

1 Para la decompresión y procesamiento de estos correos, se generó un script denominado **script_procesar_correos.py**

diversas tareas de preprocesamiento y enriquecimiento de los datos que se describen en este Capítulo.

5.1.2 *Etiquetado de documentos*

Sobre la base de 24700 correos, se seleccionaron aleatoriamente 1000 interacciones que fueron etiquetadas en torno al tema de la consulta por un experto del dominio. Inicialmente, se identificaron 20 clases independientes entre sí. Estas clases son: 'Boleto Universitario', 'Cambio de Carrera', 'Cambio de Comisión', 'Carga de Notas', 'Certificados Web', 'Consulta por Equivalencias', 'Consulta por Legajo', 'Consulta sobre Título Universitario', 'Cursadas', 'Datos Personales', 'Exámenes', 'Ingreso a la Universidad', 'Inscripción a Cursadas', 'Pedido de Certificados', 'Problemas con la Clave', 'Reincorporación', 'Requisitos de Ingreso', 'Simultaneidad de Carreras', 'Situación Académica' y 'Vacunas Enfermería'.

Sin embargo, a partir de la primera iteración en la ejecución de los experimentos y el análisis de error que se describe en profundidad en la Sección 5.4.1, se decidió fusionar clases, resultando 16 clases definitivas, cuya temática de consulta se explica a continuación:

1. **Boleto Universitario:** en esta clase se concentran las consultas sobre la implementación del subsidio por parte del estado provincial para los viajes desde y hacia la Universidad.

2. **Cambio de Carrera:** consultas relacionadas con la posibilidad, fechas y requisitos para cambiar de propuesta formativa.

3. **Cambio de Comisión:** luego de la inscripción a cursadas, los estudiantes cuentan con la posibilidad, durante un tiempo acotado, de modificar el horario o sede en que cursan las asignaturas. En esta categoría se agrupan este tipo de consultas.

4. **Consulta por Equivalencias:** en esta clase se categorizan las consultas de los estudiantes que desean presentar documentación de actividades académicas aprobadas en otra Universidad o otra propuesta formativa y desean solicitar equivalencias con una nueva Carrera.

5. **Consulta por Legajo:** Al momento de ingresar en la Universidad, en oportunidad de inscribirse a las asignaturas del primer cuatrimestre, se genera el legajo de estudiante, el cual necesitan para las tramitaciones a lo largo de toda su trayectoria académica. Esta categoría se utiliza para consultas relacionadas con esta tarea.

6. **Consulta sobre Título Universitario:** en esta clase se agrupan las consultas relacionadas con la metodología de tramitación, requisitos y tiempos relacionados con la expedición del título universitario.

7. **Cursadas:** aqui se agrupan todas las consultas que tienen que ver con el cursado de actividades académicas: plazos de inscripción, problemas con la oferta, correlativas, etc.

8. **Datos Personales:** en ocasiones, sucede que los estudiantes desean modificar sus datos filiatorios producto de actualizaciones o que los mismos son incorrectos; estas consultas tienen que ver con esa dinámica.

9. **Exámenes:** la UNLu dispone de cinco turnos regulares de exámenes finales, donde los estudiantes pueden inscribirse en una mesa a efectos de rendir examen de las asignaturas. Esta clase aglutina todas las consultas relacionadas con la inscripción, la oferta, el contacto con los docentes, la asistencia al examen y otras cuestiones relacionadas a los exámenes finales.

10. **Ingreso a la Universidad:** aquí se agrupan todas las consultas que se originan en relación a la inscripción de los nuevos estudiantes, denominados aspirantes, en la Universidad Nacional de Luján. Estas consultas están relacionadas con la oferta por sede, momentos de inscripción y demás efectos previos al inicio de la inscripción.

11. **Pedido de Certificados:** Para la acreditación de su asistencia a la Universidad y su trayectoria académica, existen diferentes certificados que los estudiantes solicitan y cuyas consultas se etiquetan en esta clase.

12. **Problemas con la Clave:** La Universidad cuenta con un Sistema Web mediante el cual los estudiantes generan gran cantidad de tramitaciones y a la cual acceden mediante su legajo y una clave de acceso. En oportunidad de generación y reseteo de la misma, se generan diversas consultas que se etiquetan en esta clase.

13. **Reincorporación:** El Régimen General de Estudios de la Universidad prevé que los estudiantes que no hayan aprobado dos actividades académicas al año, queden en condición de libres en la Carrera, pudiendo reincorporarse en la misma a partir de un trámite y cumpliendo con una serie de requisitos temporales. Las consultas y solicitudes en ese sentido se aglutinan en esta clase.

14. **Requisitos de Ingreso:** Luego de manifestar la voluntad de inscribirse en la Universidad, los estudiantes deben completar una serie de requisitos documentales para completar esa tramitación, como por ejemplo el título del nivel medio, apto-médico o certificado de vacunación.

15. **Simultaneidad de Carreras:** De acuerdo a la normativa de la UNLu, los estudiantes pueden cursas hasta dos Carreras de forma simultánea, habiendo tenido que aprobar todas las asignaturas del primer año de su plan de estudios de origen para inscribirse en el segundo, lo cual genera consultas que se agrupan en esta clase.

16. **Situación Académica:** En esta última categoría se incorporan las consultas relacionadas con la carga de calificaciones, controles de promoción y otras situaciones relacionadas con su trayectoria académica.

Una vez definidas las clases presentes en esta muestra, los correos fueron etiquetados a partir del asesoramiento del Director General de Asuntos Académicos de la Universidad

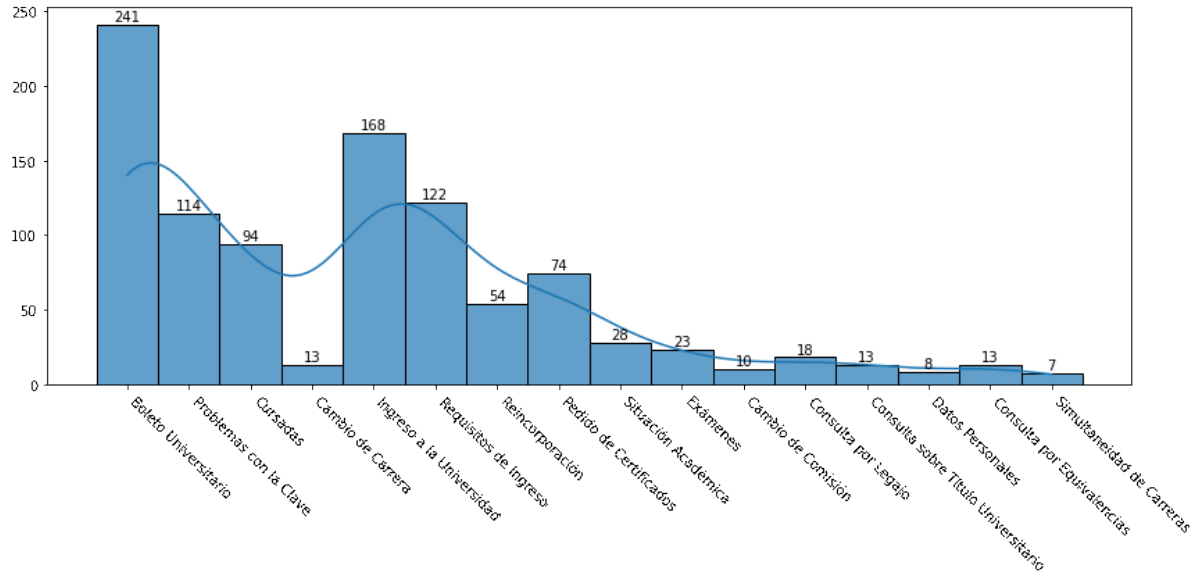

Figura 5.3: Frecuencia observada para las clases resultantes del etiquetado manual

Nacional de Luján, el cual adoptó el rol de especialista del dominio para la temática objeto de esta investigación.

La distribución de la frecuencia por cada clase puede visualizarse en la Figura 5.3. Como se desprende del recurso, las clases están altamente desbalanceadas, lo cual hace más compleja la tarea de construcción de un clasificador automático sobre un conjunto de datos masivos. Mientras que existen algunas clases, como "Boleto Universitario" que contienen el 24.1 % de las instancias, otras clases, como "Simultaneidad de Carreras" y "Datos Personales", poseen menos de diez instancias sobre las mil etiquetadas.

5.1.3 *Preprocesamiento de los correos*

A lo largo de todo el proceso, se realizaron diversas actividades de preprocesamiento de los correos que se explican en esta etapa.

En la primera etapa de preprocesamiento, el proceso[2] itera sobre el repositorio de las consultas y, por cada correo, se realizan las siguientes operaciones[3]:

1. Se tomó el texto plano del correo y se solucionaron problemas de codificación del texto sobre las letras que llevan tilde. A su vez, se transforma el texto a minúsculas y se eliminaron saltos de línea y caracteres especiales.

2. A continuación, por cada correo se separó el campo de la consulta de la respuesta por parte del staff de la Universidad a partir de la identificación de un token especial (texto: ––*mensaje original*––"). A los efectos de este trabajo, en esta etapa, se desecharon las cadenas de consultas y respuestas que tuvieran más de una interacción.

2 Para este preprocesamiento inicial de los correos, se utilizó el script denominado **script_procesar_correos.py**
3 El dataset resultante está en la carpeta **data** y se denomina **00-correos_etiquetados.csv**

3. Luego, se estructuraron las consultas en texto plano en un conjunto de atributos que determinan diferentes aspectos del correo, delimitando el inicio y el fín de la consulta y prescindiendo del encabezado y pié del correo, que estaban relacionados con la seguridad de la comunicación que no incorporaban valor al contenido. Los atributos resultantes del campo consulta original son:
   - **Fecha de la consulta:** Es la fecha en la que se envió el correo de la consulta en un formato DD-MM-AAAA.
   - **Hora de la consulta:** Se almacena el horario de la consulta en un formato HH:MM:SS.
   - **Apellido y nombre del estudiante:** Si bien el nombre y apellido luego no se utiliza para la construcción del modelo, estos datos se almacenan en la tabla.
   - **Legajo:** El legajo es un número entero de entre cinco y seis cifras.
   - **Documento:** Número de identificación de la identidad del estudiante.
   - **Carrera:** Propuesta formativa en la que se encuentra inscripto.
   - **Teléfono:** Teléfono proporcionado por el estudiante en la consulta.
   - **Dirección de E-mail:** Dirección de correo electrónico proporcionada por el estudiante en la consulta.
   - **Consulta:** Es el texto de la consulta realizada por el estudiante. La mayor parte del preprocesamiento desarrollado en la segunda etapa se realiza sobre este campo.
4. A los campos del ítem anterior se adicionó la respuesta. Este atributo se utilizó mayoritariamente para determinar la clase de la consulta.

El conjunto de datos resultante de este primer preprocesamiento se utiliza como insumo para la próxima etapa, en la cual se realiza una transformación de los atributos antes presentados. Es importante aclarar que la etapa de preprocesamiento se aplicó tanto sobre los 1000 correos tomados como muestra para el entrenamiento de los modelos así como para el resto del repositorio. Luego del preprocesamiento y depuración de la información, el conjunto de datos quedó conformado por 20876 correos.

5.2 ANÁLISIS EXPLORATORIO DEL CONJUNTO DE DATOS

A continuación, se realizó un análisis exploratorio de los datos para conocer la distribución de los mismos a partir de métodos gráficos.

Para ello, se generaron una serie de atributos en función de los existentes, ya introducidos en el apartado anterior. Los atributos incorporados para el análisis exploratorio se describen a continuación, así como su método de cálculo[4].

4 Para este procesamiento se desarrolló una notebook denominada **01-incorporacion-atributos_estaticos.ipynb**

5.2.1 *Análisis de la fecha de la consulta*

A partir del atributo *fecha* se generan los siguientes atributos:

- **dia_semana:** se convierte en un atributo numérico asignando el 1 para el lunes y el 7 para el domingo.
- **semana_del_mes:** se convierte en un número entero entre 1 y 5 que determina el orden de la semana del mes.
- **mes:** se convierte en un valor entre 1 y 12 para identificar el mes.
- **cuatrimestre:** se convierte en un valor entre 1 y 3 para identificar el momento del año.
- **anio:** se convierte en el valor entero que identifica el año.

5.2.1.1 *Atributo: dia_semana*

Se realiza un gráfico de barras sobre el atributo **dia_semana**:

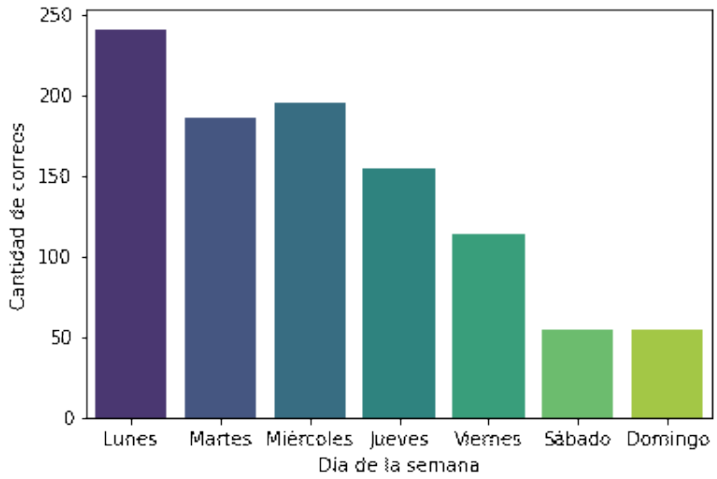

Figura 5.4: Cantidad de correos por día de la semana

El gráfico precedente permite verificar que, al menos para el muestreo elegido, existe una mayor cantidad de consultas los días iniciales de la semana, las cuales decrecen conforme avanza la misma.

5.2.1.2 *Atributo: semana_del_mes*

Se realiza un gráfico de barras, el cual no arroja grandes diferencias en torno a la semana del mes, salvo que en la quinta semana no aparecen demasiadas consultas. Esto se debe, seguramente, a que existen varios meses que sólo cuentan con cuatro semanas completas.

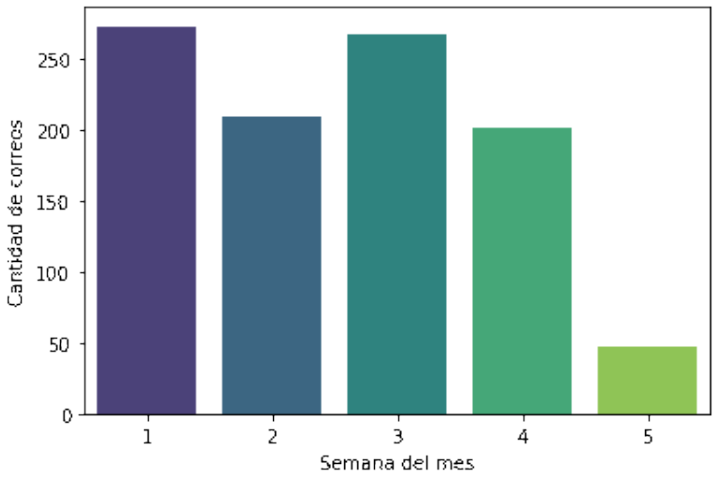

Figura 5.5: Cantidad de correos por semana del mes

5.2.1.3 *Atributo: mes*

Se realiza un gráfico de barras, verificando que gran cantidad de los correos de la muestra fueron recibidos durante los primeros meses del año, comenzando a partir de febrero puesto que la UNLu se encuentra en receso de verano durante el mes de enero.

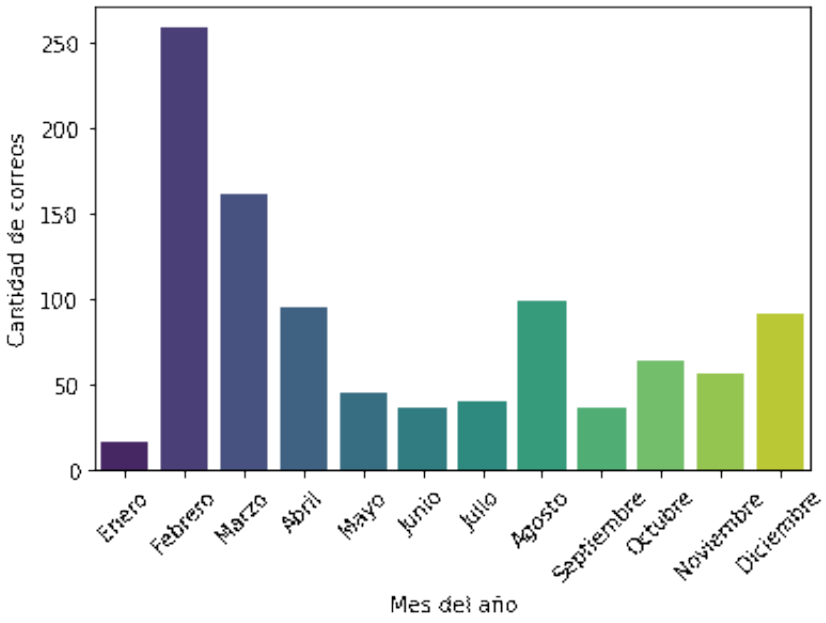

Figura 5.6: Cantidad de correos por mes

5.2.1.4 *Atributo: anio*

Se realiza un gráfico de torta para verificar la distribución de los correos de la muestra por **anio**, el cual permite verificar a 2015 y 2019 como los años que contienen la mayor cantidad de la muestra de las consultas.

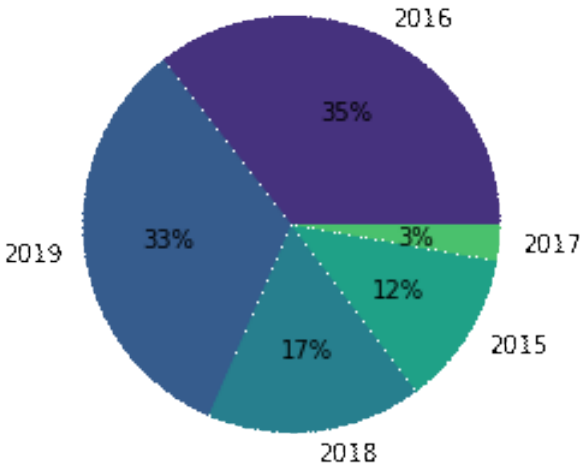

Figura 5.7: Cantidad de correos de la muestra por año

En este punto, se recupera la base de conocimiento completa con todas las consultas con el objetivo de verificar si la distribución de la llegada de correos es consecuente con lo observado en la Figura anterior.

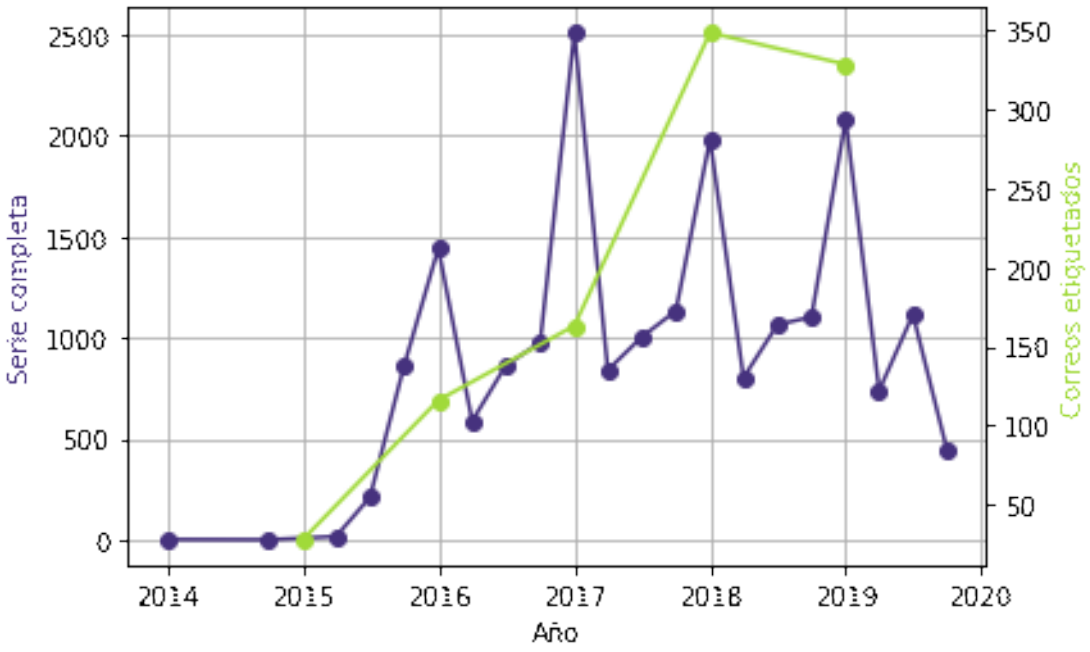

Figura 5.8: Cantidad de correos por fecha sobre el total de consultas y los correos etiquetados

Si bien la serie de la muestra de correos etiquetados y la que conforma el total de consultas resultan muy difíciles de comparar debido a la diferencia en la cantidad de instancias, pareciera que no existe una relación entre la distribución de la cantidad de correos por fecha entre una y otra muestra de datos.

5.2.2 *Análisis de los atributos categóricos*

A continuación se describen los atributos de tipo discreto que fueron generados para el análisis exploratorio así como su lógica:

- En función del atributo *horario*, se genera un atributo discreto con la siguiente lógica: 0-Mañana (6-12 hs), 1-Media-Tarde (12-16 hs), 2-Tarde (16-20 hs), 3-Noche (20-00 hs) y 4-Madrugada (00-6 hs).
- En función del atributo *documento*, se genera un nuevo atributo discretizado a partir de la estrategia de cuantiles, asumiendo que los valores más bajos de *documento* representan a estudiantes más longevos.
- En función del atributo *legajo*, se genera un nuevo atributo discretizado a partir de la estrategia de cuantiles, asumiendo que los valores más altos representan a estudiantes con menos antigüedad en la Universidad.
- Se extrae el código que determina la Carrera del campo carrera, el cual tenía originalmente el código y la denominación.
- Se obtiene el proveedor de la cuenta de correo a partir del campo *correo_electronico* ingresado en la consulta.
- A partir del atributo teléfono, se genera un atributo *dummy* que indica la existencia o no del teléfono en la consulta.
- Además, a partir del atributo *legajo*, se genera un segundo atributo que opera de la misma forma que para el campo *teléfono*.

Una vez calculados los atributos, se generaron los gráficos para observar el comportamiento, los cuales se observan en la Figura 5.9. De la misma, pueden sacarse algunas conclusiones que mejoran el entendimiento de los datos.

En primer lugar, si bien resulta bastante homogénea la distribución a lo largo del día del envío de consultas, el horario en el que más se reciben es el identificado con 1 (12-16 hs). Además, no existen consultas en el horario de 00-06 hs (valor 4). En cuanto a los atributos dni_discretizado y legajo_discretizado, no existen prácticamente variaciones en los diferentes agrupamientos dado que se utilizó la técnica de cuantiles. En relación a las Carreras a las que pertenecen los estudiantes que realizan las consultas, los valores con mayor cantidad de observaciones son: 3-Licenciatura en Administración, 5-Licenciatura en Trabajo Social, 54-Contador Público y en menor medida 43-Profesorado Universitario en Educación Física. En general, esta situación responde a que son las Carreras con mayor matrícula en la Universidad Nacional de Luján. El gráfico en torno al proveedor de correo electrónico, muestra que los mismos están monopolizados prácticamente en los valores 4 y 6 que corresponden a hotmail y gmail respectivamente. Por último, se observa que una gran cantidad de estudiantes remiten su número de teléfono ante la realización de las consultas, en torno al 90 %; y llama la atención que cerca del 30 % no conozcan o expliciten su número de legajo entre los datos provistos.

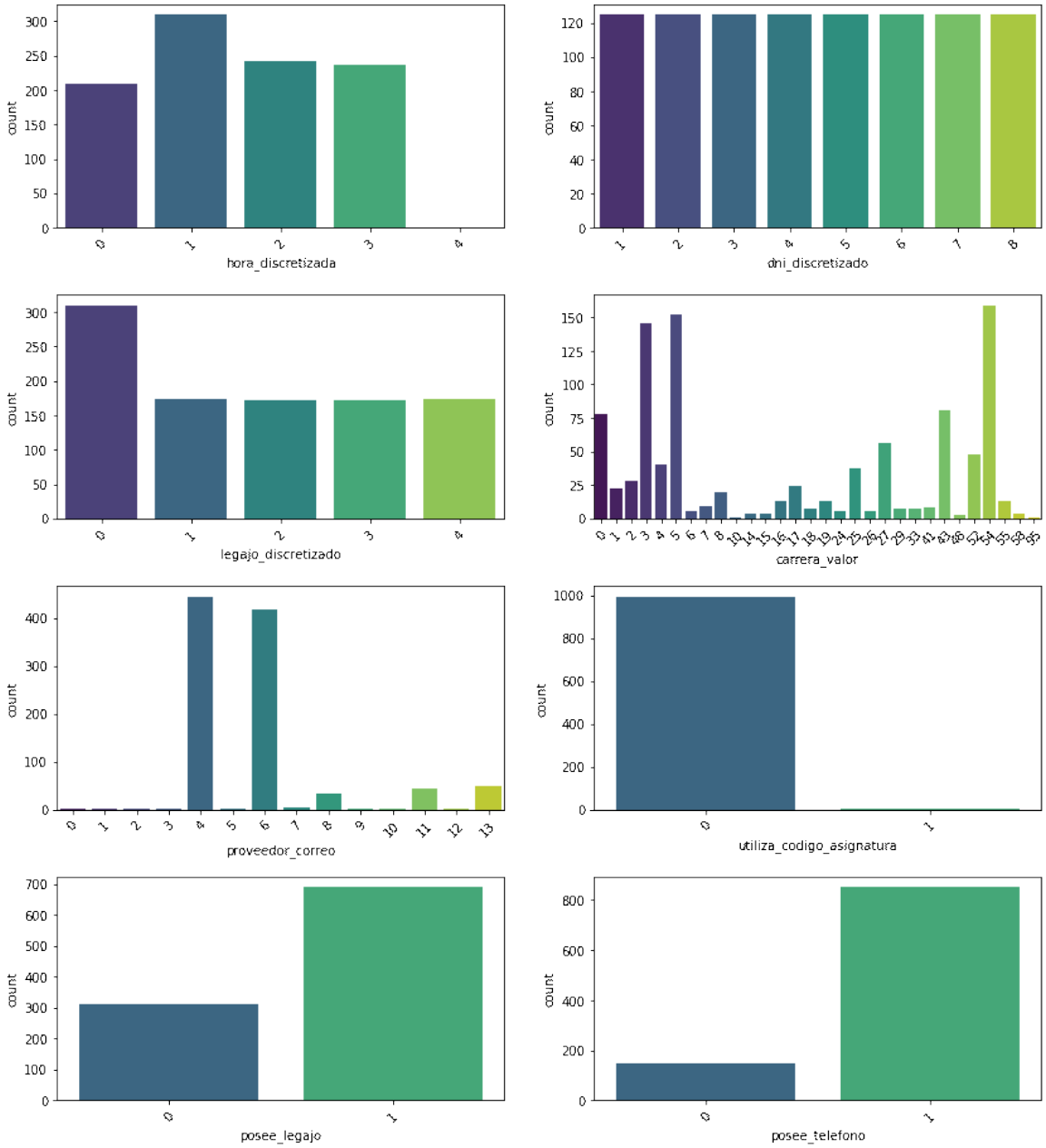

Figura 5.9: Resumen de atributos categóricos

5.2.3 *Análisis exploratorio del texto de la consulta*

En función de la caracterización introducida por Layton [56], se generaron un conjunto de características calculadas para explicar el comportamiento de las consultas realizadas por los estudiantes.

- A partir del atributo *consulta* se generan un conjunto de atributos léxicos basados en caracteres:
  - **cantidad_caracteres:** Se calcula la cantidad de caracteres de la consulta.

  - **proporcion_letras:** Se computa la proporción de letras respecto a *cantidad_caracteres*.
  - **cantidad_tildes:** Se calcula la cantidad de tildes de la consulta.

- A su vez, también a partir del atributo *consulta*, se generan un conjunto de atributos léxicos basados en palabras:
  - **cantidad_palabras:** Se calcula la cantidad de palabras de la consulta en función de los espacios.
  - **cantidad_palabras_cortas:** Se computa la cantidad de palabras cortas (hasta 4 caracteres).
  - **cantidad_palabras_distintas:** A efectos de verificar la riqueza del lenguaje, se computa la cantidad de palabras diferentes.
- A partir de la consulta, se genera un atributo sintáctico, denominado *cantidad_signos_puntuacion* que evalúa el uso de los signos de puntuación por parte de los estudiantes.
- Por último, también tomando la variable *consulta*, se generan dos atributos estructurales: *cantidad_oraciones* con la cantidad de frases y *utiliza_codigo_asignatura* a efectos de verificar si el estudiante tiene en cuenta mencionar esta información.

Para su análisis, en primer lugar se incorporan histogramas de los atributos estáticos léxicos basados en caracteres.

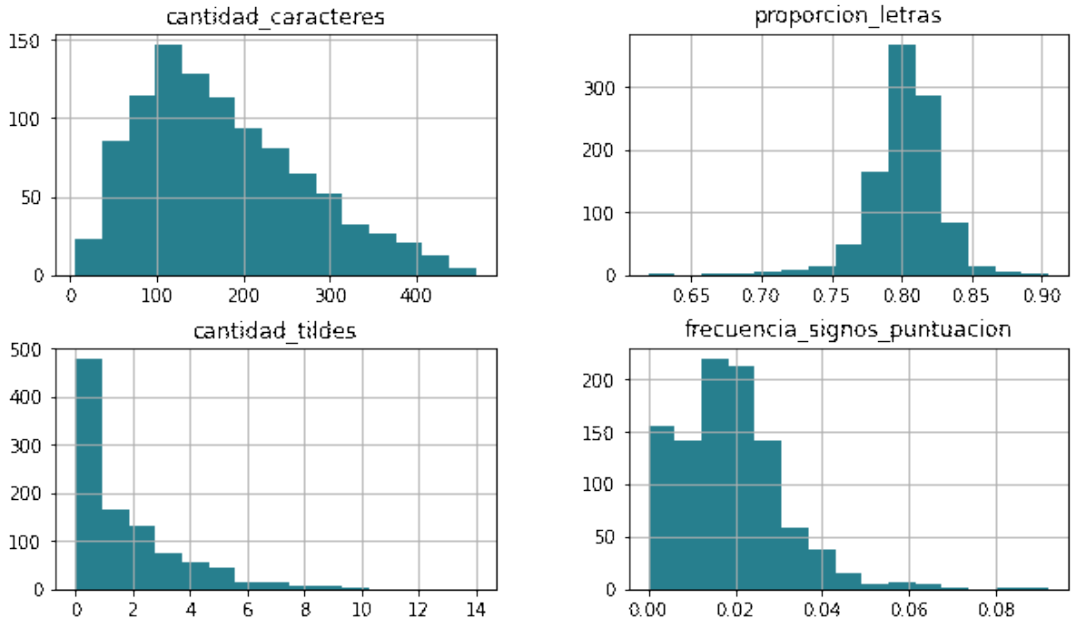

Figura 5.10: Histograma de los atributos léxicos basados en caracteres

De los recursos precedentes se observa que en promedio las consultas rondan los 110 caracteres aproximadamente, con una distribución que se asemeja a una normal. Esta exploración confirma la idea que los correos electrónicos de estas características en general constituyen textos cortos.

En relación al indicador proporcion_letras, se observa que estas rondan el 80 % de los caracteres que conforman las consultas. Por su parte, una gran cantidad de consultas no contienen ningún tilde y una muy baja proporción de signos de puntuación respecto a la cantidad de caracteres, lo cual habla de la informalidad de este tipo de correos.

A continuación, se analizan los atributos léxicos orientados a palabras. La mayoría de las consultas están entre las 20 y 40 palabras. A su vez, se verifica una muy alta proporción de palabras distintas puesto que prácticamente no hay consultas en que este valor sea menor al 70 %.

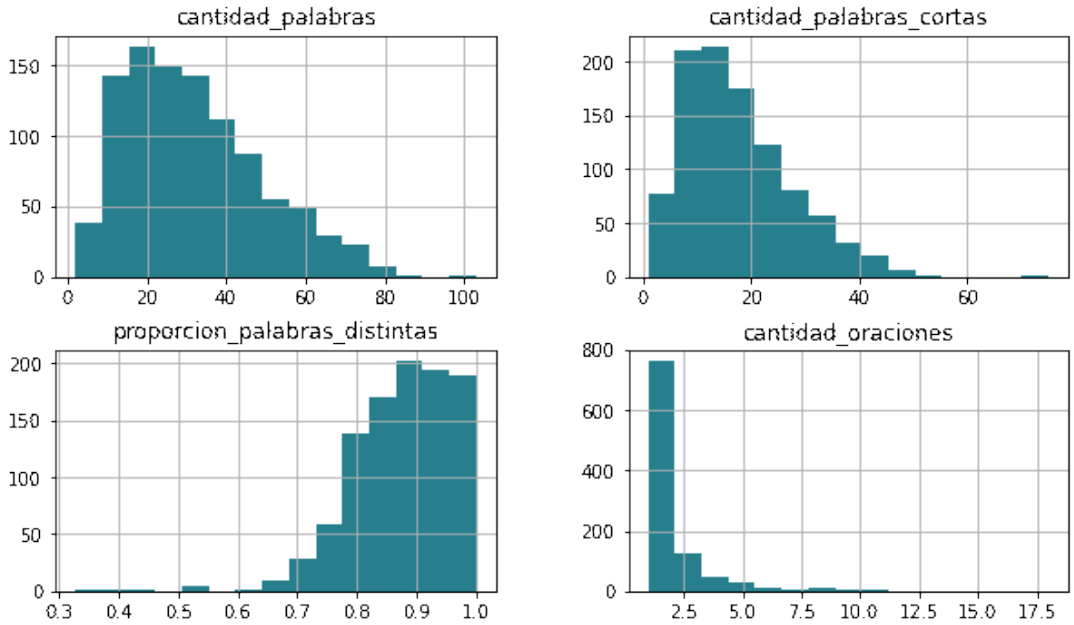

Figura 5.11: Histograma de los atributos léxicos basados en las palabras

En relación a la cantidad de oraciones, casi la totalidad de los correos cuenta con menos de tres oraciones. Asimismo, para dimensionar el atributo cantidad_palabras_cortas, se sitúa en términos de la proporción con respecto a la cantidad de palabras de cada consulta.

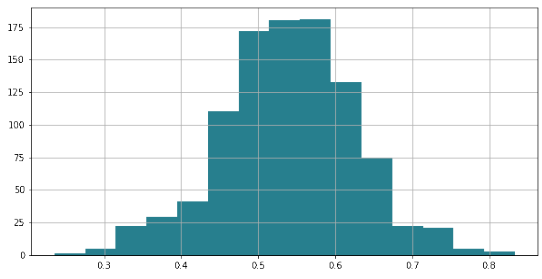

Figura 5.12: Histograma de la proporción de palabras cortas por correo

A partir de la Figura 5.12, se observa que la distribución responde a una normal con centro en torno al 0.55, lo cual muestra que, en promedio, cerca del 50 % de las palabras que conforman los correos electrónicos son cortas (poseen hasta cuatro caracteres).

Por último, se hace un análisis de la relación entre la cantidad de caracteres de los correos electrónicos etiquetados y las diferentes categorías.

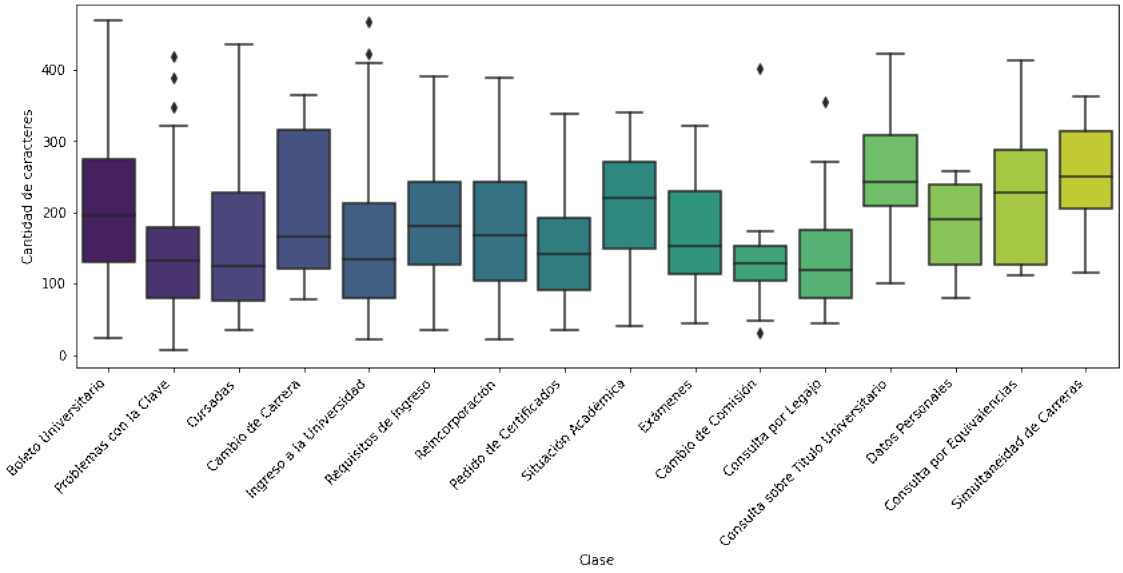

Figura 5.13: Diagramas de cajas de la cantidad de caracteres por consultas agrupados por clase

En este sentido, se observan algunas cuestiones interesantes. Se verifica que, en general, los correos electrónicos relacionados con los cambios de comisión poseen una extensión menor que el resto de las clases. A su vez, los correos con consultas sobre el boleto universitario son los que muestran mayor dispersión en torno a la cantidad de caracteres. Otra cuestión saliente es que el 75 % de los correos que consultan sobre el título universitario cuentan con más de 200 caracteres, algo similar ocurre para los correos sobre simultaneidad de Carreras.

5.3 SEPARACIÓN DEL CONJUNTO DE DATOS EN ENTRENAMIENTO Y EVALUACIÓN

Sobre la base de 24700 correos iniciales, luego del procesamiento inicial se consolidó una base de conocimiento con 20876 correos electrónicos preprocesados. En función de estos correos, se seleccionaron aleatoriamente 1000 interacciones que, cómo se expresó antes, fueron etiquetadas en torno al tema de la consulta por un experto del dominio.

A su vez, se decidió dividir el conjunto de datos etiquetado en una proporción de 80 % y 20 %, utilizando el primer conjunto de datos para el entrenamiento y validación con la estrategia de validación cruzada, o *cross validation*, que se introdujo en la Sección 3.4.6.3 y el segundo conjunto de datos para la evaluación de los modelos seleccionados[5].

Como ya se había introducido en el momento que se abordó el problema de la distribución de ejemplos por clases a partir de la Figura 5.3, las mismas están altamente desbalanceadas, lo cual hace más compleja la tarea de construcción de un clasificador automático sobre un conjunto de datos masivos. Por ello, a efectos de asegurar que tanto las instancias para entrenamiento y validación como las utilizadas para la evaluación de los modelos dispusieran de ejemplos de todas las clases, se realizó un muestreo estratificado por clase, como se observa en la Figura 5.14.

5 Los conjuntos de datos resultantes de la separación en entrenamiento y evaluación se encuentran en la carpeta **data** y se denominan **01-01-correos-train-80.csv** y **01-02-correos-test-20.csv**.

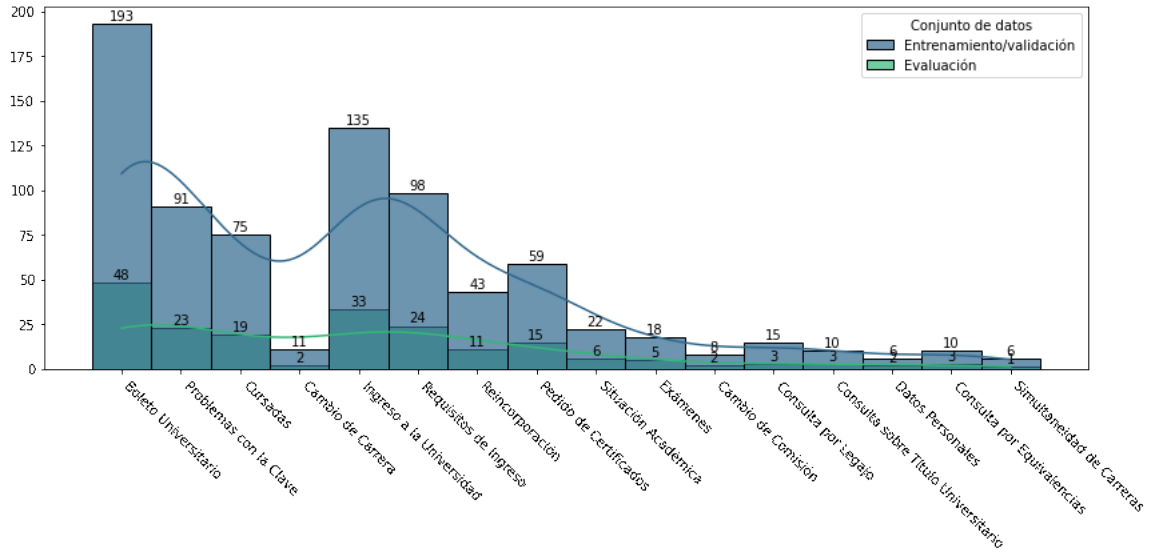

Figura 5.14: Frecuencia observada para las clases resultantes del etiquetado manual

5.4 EJECUCIÓN DE LOS EXPERIMENTOS

En primer lugar, se diseñaron y ejecutaron un conjunto de experimentos de caracter inicial, en función de tres estrategias diferentes de aprendizaje automático [29]:

1. **BoW+SVM**[6]**:** Esta estrategia de representación tiene como ventaja la simplicidad y a su vez la posibilidad de aplicar a la representación resultante cualquiera de las técnicas de clasificación existentes. Una de las usualmente utilizadas es máquina de vectores de soporte (SVM), presentada a mediados de 1990, fue ganando popularidad debido a algunas características atractivas y su rendimiento empírico.

2. **Word2Vec+LSTM**[7]**:** Una línea de investigación bastante actual comprende la utilización de información contextual junto con modelos simples de redes neuronales para obtener representaciones de palabras y frases en el espacio vectorial[101]. Uno de los modelos más populares es Word2Vec, el cual dispone de dos arquitecturas diferentes, a saber, CBoW y Skip-gram [70]. Estos modelos de incrustaciones de palabras usualmente se complementan con redes neuronales recurrentes como LSTM dado que proveen características que mejoran sustancialmente el rendimiento de las redes neuronales convencionales para el tratamiento de texto [2].

3. **BERT**[8]**:** Como una evolución a la estrategia anterior, en 2017, se propone una nueva arquitectura de red neuronal, más simple y paralelizable, denominada *Transformer* [96], basada únicamente en mecanismos de atención, prescindiendo por completo de recurrencia y convoluciones. Nace entonces lo que en la literatura se conoce como

6 Los experimentos realizados en base a esta estrategia se encuentran presentes en la notebook **03-bow+binario+svm.ipynb**
7 Los experimentos realizados en base a esta estrategia se encuentran presentes en la notebook **04-Word2Vec+LSTM.ipynb**
8 Los experimentos realizados en base a esta estrategia se encuentran presentes en la notebook **05-BERT.ipynb**

el estado del arte actual de los modelos de representación del lenguaje, denominado BERT *Bidirectional Encoder Representations from Transformers* [25].

Para el enfoque de aprendizaje basado en **BoW+SVM** se normalizó el texto, se eliminaron palabras vacías y se experimentó con diferentes variaciones de *n-gramas* de palabras y caracteres. En cambio, para los enfoques basados en **Word2Vec+LSTM** y **BERT** se utilizaron las secuencias de texto relativas a la consulta únicamente con el texto normalizado y se eliminaron palabras vacías solo para **Word2Vec+LSTM** puesto que para **BERT** en las pruebas experimentaba una baja de la performance. Para el enfoque **Word2Vec+LSTM** se utilizaron incrustaciones de palabras pre-entrenados disponibles para el idioma español [16]. En cuanto a **BERT**, se experimentó con dos modelos pre-entrenados, uno nativo para el idioma español (BETO) [18] y otro, denominado *Multilingual* [25], desarrollado para múltiples idiomas. Para la validación de los modelos se utilizó una validación cruzada con *5-fold* sobre el 80 % de las instancias en la etapa de entrenamiento mientras que luego se *testeó* el modelo mediante las 20 % restante de las instancias a partir de las métricas *accuracy*, *precision*, *recall*, *f1-score* y *Matthews correlation coefficient* o simplemente *MCC* [22].

5.4.1 *Primera iteración: Distribución de clases original*

Como se comentó anteriormente, se generó un primer lote de experimentos en función del etiquetado de los correos realizado originalmente. En todos los casos se realizó una búsqueda de los mejores hiperparámetros para cada estrategia, obteniendo los siguientes resultados:

Tabla 5.1: Resultados de los experimentos con las distintas estrategias de aprendizaje (Iteración #1).

Estrategia	Accuracy	Precision	Recall	F1-score	MCC
$BoW + SVM$	0.740	0.701	0.740	0.713	0.692
$Word2Vec + LSTM$	0.650	0.656	0.650	0.645	0.590
$BERT(Multilingual)$	0.685	0.624	0.685	0.626	0.000
$BERT(BETO)$	**0.760**	**0.734**	**0.760**	**0.741**	**0.717**

Los resultados obtenidos muestran que el abordaje más efectivo para la clasificación de este conjunto de datos es **BERT**, con el modelo pre-entrenado para el idioma español.

A partir de lo anterior, se hace un análisis de error en función de las matrices de confusión para los dos enfoques preponderantes: **BoW+SVM** y **BERT (BETO)**. El objetivo es interpretar en que contexto clasifican las instancias de forma errónea así como también cuales son las fortalezas de cada modelo.

Del análisis de la matriz de confusión de la estrategia **BoW+SVM**, se deduce que el modelo predijo de forma adecuada 47 de las 48 instancias para la clase 'Boleto Universitario', 12 de las 14 instancias de 'Pedido de Certificados', 18 de las 26 instancias de 'Requisitos

Figura 5.15: Matriz de confusión para la estrategia **BoW+SVM** (Iteración #1)

de Ingreso', 2 de las 3 consultas sobre 'Consulta por Equivalencias' y 'Consulta por Legajo' y 42 de las 47 consultas sobre 'Ingreso a la Universidad' aunque, como aspecto negativo, predijo 57 instancias en esta clase.

Como indicio negativo, el clasificador no fue capaz de asignar instancias para 6 de las 20 clases: 'Cambio de Carrera', 'Certificados Web', 'Datos Personales', 'Simultaneidad de Carreras', 'Situación Académica' y 'Vacunas Enfermería'.

Por su parte, se realiza el análisis de error del modelo entrenado con **BERT (BETO)**, a efectos de entender cuales son las falencias del clasificador.

En este caso, el clasificador predijo de forma acertada todas las instancias de la clase 'Boleto Universitario', 'Cambio de Comisión' y 'Vacunas Enfermería'. A su vez, clasificó correctamente 2 de las 3 instancias de 'Cambio de Carrera', las 3 de 'Consulta por Equivalencias', las 4 instancias de la clase 'Exámenes' y 42 de las 47 instancias de 'Ingreso a la Universidad' -aunque clasificó 11 instancias más en esta clase, erróneamente-.

Figura 5.16: Matriz de confusión para la estrategia **Bert (BETO)** (Iteración #1)

Existe cierta similitud en los patrones de error de ambos clasificadores puesto que en ambos casos las clases con mayor cantidad de errores son 'Ingreso a la Universidad' y 'Reincorporación', aunque la estrategia **Bert (BETO)** fue capaz de clasificar instancias en clases como 'Vacunas Enfermería' y 'Cambio de Carrera'. Sin embargo, persiste la incapacidad para clasificar consultas en las clases 'Certificados Web', 'Datos Personales', 'Simultaneidad de Carreras' y 'Situación Académica'.

A efectos de verificar la cantidad de aciertos y errores de clasificación en ambos modelos, se decide realizar la Tabla de resumen con las cantidades.

Es posible observar que la diferencia de performance entre la clasificación de las instancias realizadas por ambos modelos está en 4 instancias, en términos de valores absolutos, sin discriminar el análisis por clase.

Tabla 5.2: Clasificación de los modelos sobre instancias de evaluación (Iteración #1).

Estrategia	Bien Clasificados	Mal Clasificados
$BoW + SVM$	148	52
$BERT(BETO)$	152	48

A continuación, se hace un análisis del texto de las instancias mal clasificadas por ambos clasificadores. En este sentido, se verifica que en 36 instancias ambos clasificadores fallaron en simultáneo, coincidiendo en 18 de las mismas en la etiqueta que predijeron. Se realiza entonces un análisis de estas últimas instancias para indagar más sobre estas situaciones.

Tabla 5.3: Clasificación de los modelos sobre instancias de evaluación (Iteración #1)

Consulta	Clase	BoW+SVM	BERT
no es lo que esperaba, necesito volver a la carrera en la que estaba. gracias	Cambio de Carrera	Reincorporación	Reincorporación
me olvide la contraseña y la necesito para poder saber la fechas de finales	Requisitos de Ingreso	Problemas con la Clave	Problemas con la Clave
no recibo el mail para poder inscribirme a las materias.	Inscripción a Cursadas	Ingreso a la Universidad	Ingreso a la Universidad
hola quisieradescargar el certificado de alumna regular lo necesito por un tema médico .intentó descargarlo desde la web y me sale que soy interesante. me ayudan gracias	Certificados Web	Pedido de Certificados	Pedido de Certificados
no puedo ingresar como estudiante para obtener un certificado de alumno regular. ya me paso el año paso por lo que me acerque a la sede de lujan y supuestamente esta solucionado. aun sigo sin poder ingresar y necesito ese certificado	Reincorporación	Pedido de Certificados	Pedido de Certificados

Tabla 5.3: Continúa en la siguiente página.

Tabla 5.3: Continúa desde la página anterior.

Consulta	Clase	BoW+SVM	BERT
hola. quería consultarles la razón por la que perdí la regularidad. yo después de mucho tiempo sin poder ir a la universidad me re-inscribí a finales de 2017 y comencé en el 2018. pude cursar un par de materias pero las tuve que dejar por poco tiempo para estudio. espero su respuesta.si la respuesta es vía mail mejor para mi, ya que estoy trabajando de 8 a 18hs. muchas gracias. matias.	Reincorporación	Ingreso a la Universidad	Ingreso a la Universidad
necesito la certificación del programa análisis organizacional de la lic. en adm.,(20262), para completar el pedido de equivalencias en la univ. de gral. sarmiento, retiré el de contador, pero me falta este, quisiera saber si es factible retirarlo en san miguel, sin tener que ir a lujan, muchas gracias.	Pedido de Certificados	Consulta por Equivalencias	Consulta por Equivalencias
hola, me inscribí en la cede central el 18 de diciembre. quiero ingresar al perfil aspirante con mi numero de documento (para inscribirme al taller) y no me deja. quería saber por que no puedo entrar, ya que la inscripción a talleres cierra en unos días.	Inscripción a Cursadas	Ingreso a la Universidad	Ingreso a la Universidad
no puedo entrar a mi perfil,podrian solucionarlo.	Problemas con la Clave	Requisitos de Ingreso	Requisitos de Ingreso
como ago para conseguir mi numero d legajo?	Requisitos de Ingreso	Consulta por Legajo	Consulta por Legajo

Tabla 5.3: Continúa en la siguiente página.

Tabla 5.3: Continúa desde la página anterior.

Consulta	Clase	BoW+SVM	BERT
hice la pre-inscripcion online en fecha, luego fui a llevar los papeles a la universidad y me dijeron que de a partir del 5 de febrero podia elegir las asignaturas online, pero hoy 14 de febrero, cuando voy a hacer la inscripcion me dice &quot	Ingreso a la Universidad	Requisitos de Ingreso	Requisitos de Ingreso
cuando voy a ingresar a la pagina para accesos de aspirantes y ingreso mi documento, me aparece &quot	Requisitos de Ingreso	Ingreso a la Universidad	Ingreso a la Universidad
buenos días! en octubre hice la inscripción para la carrera de contador publico! la persona que me inscribio me dijo que masomenos en diciembre entrará como aspirante para así poder ver mi nro de comisión y el horario que me tocaría para el taller. intento ingresar como aspirante con mi nro de documento y me dice que no hay registros del aspirante o nro de documento ingresado	Cursadas	Ingreso a la Universidad	Ingreso a la Universidad
solicite durante el transcurso de este primer cuatrimestre que se rectifique mi caligicacion de la asignatura medios de pago del comercio internacional. promovi esa asignatura con un 8 , pero en mi perfil esta con calificacion 7 . y aun no se ha solucionado.	Situación Académica	Reincorporación	Reincorporación

Tabla 5.3: Continúa en la siguiente página.

Tabla 5.3: Continúa desde la página anterior.

Consulta	Clase	BoW+SVM	BERT
buenas noches. solicito información sobre como gestionar la reincorporación a la carrera ya que perdí la regularidad y no recibí el mail para poder gestionarla. saludos	Ingreso a la Universidad	Reincorporación	Reincorporación
buenos dias, reestablezco la clave para inscribirme en las materias y me notifica que ya fue enviada al mail, pero en el mail nuevo no tengo la clave. muchas gracias	Problemas con la Clave	Ingreso a la Universidad	Ingreso a la Universidad
estimados: necesito ingresar para tramitar mi titulo y me da clave incorrecta. por favor podria habilitar el ingreso a mi perfil? aprobé la ultima materia en los turnos de mayo. gracias, saludos juan jose nogueiro	Consulta sobre Título Universitario	Problemas con la Clave	Problemas con la Clave
el motivo de mi consulta es la siguiente: cuando quiero ingresar a mi perfil de estudiante el sistema me dice que no cumplo con los requisitos del sistema para ingresar, quisiera saber el motivo y que faltaría necesito ingresar para realizar la encuesta e inscribirme gracias, espero su respuesta	Ingreso a la Universidad	Requisitos de Ingreso	Requisitos de Ingreso

Tabla 5.3:Finalización de la Tabla.

A partir del análisis de la Tabla 5.3, se observa que existe un conjunto de instancias mal etiquetadas, las cuales habían sido bien clasificadas por alguno o los dos clasificadores. Por ejemplo, la consulta *"me olvide la contraseña y la necesito para poder saber la fechas de finales"* estaba etiquetada como 'Requisitos de Ingreso' y ambos clasificadores habían detectado, de forma adecuada, que correspondía a la clase 'Problemas con la clave'.

A su vez, se detectó que algunas de las clases contenían un solapamiento en su significado, lo cual también fue observado a partir de las clasificaciones. Por ejemplo, la consulta *"hola quisieradescargar el certificado de alumna regular lo necesito por un tema médico .intentó descargarlo desde la web y me sale que soy interesante. me ayudan gracias"* estaba etiquetada

como 'Certificados Web' y ambos clasificadores habían señalado a 'Pedido de Certificados' como la clase correspondiente.

A partir de los problemas antes abordados, se trabajó, en conjunto con el experto del dominio, en la revisión de las etiquetas asignadas a cada instancia así como en la redefinición de las clases, fusionando las clases iniciales en las 16 clases definitivas que se presentan en la Sección 5.1.2[9,10]

5.4.2 *Segunda iteración: Redistribución de clases y corrección de etiquetas de clases*

A partir de los ajustes explicados en la Sección precedente, se generó una nueva iteración de los experimentos, ejecutando los mismos sobre las nuevas clases e instancias definidas, utilizando las dos estrategias que mejor funcionaron en la iteración anterior: **BoW+SVM** y **BERT (BETO)**.

Tabla 5.4: Resultados de los experimentos con las distintas estrategias de aprendizaje (Iteración #2).

Estrategia	Accuracy	F1-Score	Precision	Recall	MCC
$BoW + SVM$	0.810	0.809	0.829	0.810	0.770
$BERT(BETO)$	0.855	0.847	0.845	0.860	0.839

Puede observarse, a partir de las métricas de la Tabla 5.4, una mejora sustancial de los valores para todas las métricas de selección de modelos. En el caso de **BoW+SVM**, el *accuracy* supera en un 9.45 % a los resultados de la iteración 1 mientras que **BERT (BETO)** mejora aún más, alcanzando un *accuracy* 12.5 % mayor a la anterior ejecución.

El modelo que demuestra la mejor performance sigue siendo el entrenado a partir de **BERT (BETO)**, con un rendimiento un 5.55 % superior a la estrategia **BoW+SVM** en términos del *accuracy*.

A partir de estos resultados, se avanza en el diseño y ejecución de experimentos en función de la estrategia de etiquetado y clasificación semi-supervisada propuesta en el capítulo anterior.

9 Este trabajo de revisión del conjunto de datos está presente en la notebook **06-revision-etiquetado.ipynb**.
10 Los conjuntos de datos resultantes de la revisión se encuentran en la carpeta **data** y se denominan **02-01-correos-train-80.csv** y **02-02-correos-test-20.csv**.

5.5 IMPLEMENTACIÓN DE LA ESTRATEGIA DE APRENDIZAJE SEMI-SUPERVISADO

La implementación de la estrategia de aprendizaje semi-supervisado se realizó a partir de las mismas instancias etiquetadas con las cuales se entrenaron y evaluaron los modelos anteriores. El flujo de trabajo llevado a cabo, y que se describe a continuación, es el esquematizado en la Figura 4.1 *"Flujo de trabajo para el etiquetado semi-supervisado de correos electrónicos"*.

5.5.1 *Extracción de características*

En primer lugar, se realizó la extracción de las características representativas de cada clase a partir de las tres técnicas propuestas en el Capítulo anterior: la ponderación TF-IDF agrupada por clases, la valoración de palabras obtenida con el modelo para clasificación Sequential S3 (Smoothness, Significance, and Sanction), o simplemente SS3, y los coeficientes de las funciones de clasificación de la regresión logística para cada clase.

Las extracción de características se realizó con el conjunto de datos de entrenamiento, el cual constaba de 800 sobre las 1000 instancias etiquetadas. Las restantes 200 se reservaron para la evaluación de los modelos. A partir de estas instancias de entrenamiento, se ejecutaron las estrategias previamente presentadas.

En el caso de TF-IDF[11], se la introdujo como estrategia no supervisada de ponderación de términos y se aplicó en conjunto con una representación de documentos basada en el modelo vectorial, o bolsa de palabras [80].

Figura 5.17: Extracción de términos TF-IDF para clase 'Problemas con la Clave'

11 El proceso de extracción de características con TF-IDF se encuentra desarrollado en la notebook **07-seleccion-features-tf_idf.ipynb**

Los términos se ponderaron con TF-IDF[12] para cada término en cada documento y luego se agruparon los términos por clase aplicando la función de agrupación de promedio, quedando así una matriz con las clases como instancias y los términos (con el promedio observado de la ponderación TF-IDF en la intersección) como columnas. A efectos de seleccionar los términos más representativos para cada clase, se optó por recuperar los 20 términos con mayor ponderación de TF-IDF promedio por clase. En este sentido, en la Figura 5.17 se presentan, en un gráfico de nube de palabras, todos los términos representativos para la clase 'Problemas con la Clave' según la estrategia TF-IDF, sin limitar la cantidad.

Por su parte, se incorpora SS3[13] como una estrategia supervisada de extracción de características puesto que esta técnica genera un vocabulario para cada clase con una ponderación realizada a partir de un valor de confianza o *CV*. En este caso, se ajustó un modelo de clasificación con SS3 para las 800 instancias de entrenamiento y se obtuvieron los términos correspondientes al vocabulario de cada clase, utilizando los 20 mas representativos para la recuperación de correos. En la Figura 5.18, se muestran los términos más representativos obtenidos por SS3 para la clase 'Problemas con la Clave'.

Figura 5.18: Extracción de términos SS3 para clase 'Problemas con la Clave'

Por último, la técnica basada en la regresión logística[14] se introduce debido a que, además de ser una estrategia supervisada, permite identificar, además de los términos representativos para cada clase, un conjunto de términos que son nocivos para la elección de un tópico determinado.

12 Para ello, se utilizó la fórmula de TF-IDF por defecto de la clase *TfidfVectorizer* de la librería *sklearn* para Python.

13 El proceso de extracción de características con SS3 se encuentra desarrollado en la notebook **08-seleccion-features-ss3.ipynb**

14 El proceso de extracción de características a partir de la regresión logística se encuentra desarrollado en la notebook **09-seleccion-features-lr.ipynb**

Para esta estrategia de extracción de características, se representaron los documentos a partir del modelo vectorial, se ajustó un modelo[15] en función de las instancias de entrenamiento y se seleccionaron los 10 términos con ponderaciones mas altas -en valor absoluto- tanto para los ejemplos positivos como para los negativos.

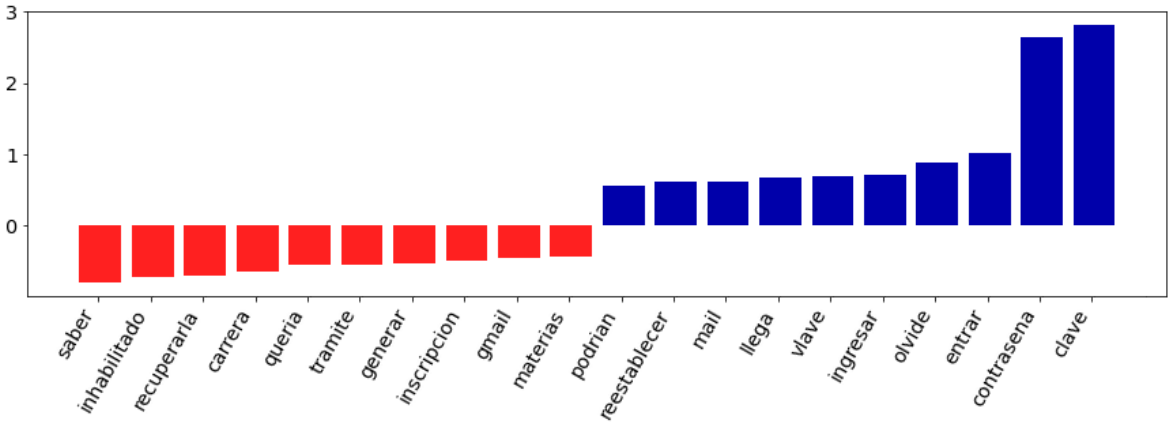

Figura 5.19: Extracción de términos LR para clase 'Problemas con la Clave'

La Figura 5.19 muestra los términos más representativos para la clasificación de consultas sobre la clase 'Problemas con la Clave' desde el centro hacia la derecha -en color azul- y, hacia la izquierda, los términos que, por el contrario, son representativos de las consultas que no corresponden a esta clase.

5.5.2 *Recuperación de correos electrónicos*

A partir de contar con los términos más representativos para cada clase -y su ponderación- según las tres técnicas de extracción de características abordadas, se recuperaron los correos electrónicos indexados en el motor de búsqueda *Elasticsearch*[16,17] a partir de dos diferentes estrategias.

Por un lado, se realizó una recuperación a partir de los términos para cada clase según su aparición o no, a modo de un *query* convencional. Por otro lado, se incorporó la ponderación de cada término según las distintas estrategias, lo que se denomina en el contexto de la herramienta como *boosting*. A su vez, para la estrategia basada en regresión logística se realizó un *boosting negativo* para los términos con coeficiente menor a cero, penalizando a los documentos en que aparecen. Se realizaron experimentos con lotes de 20, 50, 100 y 200 documentos para cada clase y cada una de las estrategias. Es importante hacer notar que el motor de búsqueda, además de la posibilidad de limitar la cantidad de resultados, provee un *score* para magnificar el nivel de similitud de la búsqueda en relación a cada documento recuperado.

15 Previamente, se realizó una búsqueda *grid* alternando diferentes valores del parámetro C de la clase *LogisticRegression* de la librería **sklearn** variando los esquemas de ponderación de los términos. El mejor valor se obtuvo con $C = 1$ para el esquema de ponderación binario.

16 En el directorio **elastic+kibana** del repositorio del proyecto, puede encotrarse una versión dockerizada de *elasticsearch* y el script de exportación de los correos de la herramienta jsonpyes.

17 Para este proceso se utilizaron los scripts **script-tfidf.py**, **script-ss3.py** y **script-lr.py**.

En la Tabla 5.5, a efectos explicativos, se presenta un resumen, para la estrategia de regresión logística, de la recuperación de 200 documentos por clase, comparando ambas estrategias de recuperación de documentos y un resumen de los *scores* obtenidos en cada clase en términos de similitud de estos con respecto a las características obtenidas para cada clase. En este sentido, las columnas *Coincidencias* y *Diferencias* muestran la cantidad de correos recuperados coincidentes y divergentes respecto de las estrategias de recuperación con y sin *boosting*; mientras que las columnas restantes lucen a modo de resumen en términos de los *scores* asignados por la herramienta *Elasticsearch* para cada clase.

Tabla 5.5: Recuperación con *Elasticsearch* (doc=200) con y sin boosting (LR)

Clase	Coincidencias	Diferencias	Max	Min	Avg
Boleto Universitario	141	59	23.22	12.10	14.21
Cambio de Carrera	138	62	14.99	7.62	9.09
Cambio de Comisión	156	44	20.10	8.38	10.52
Equivalencias	143	57	13.11	6.58	8.04
Consulta por Legajo	194	6	19.61	8.64	11.15
Título Universitario	149	51	16.71	7.59	9.28
Cursadas	162	38	13.45	8.04	9.25
Datos Personales	156	44	14.83	6.61	8.27
Exámenes	144	56	19.29	8.61	10.49
Ingreso a la Universidad	176	24	15.00	8.04	9.62
Pedido de Certificados	161	39	28.98	10.57	14.20
Problemas con la Clave	173	27	22.30	8.96	11.16
Reincorporación	147	53	14.25	7.38	8.96
Requisitos de Ingreso	177	23	18.33	10.87	12.86
Simultaneidad	131	69	21.09	7.89	10.34
Situación Académica	125	75	17.53	8.95	11.07
Total general	**2473**	**727**	**28.98**	**6.58**	**10.53**

5.5.3 *Construcción del Modelo de clasificación*

Para la etapa de construcción del modelo de clasificación, en primera instancia se utilizaron como conjunto de entrenamiento las instancias recuperadas para cada clase a partir de *Elasticsearch* en función de los términos generados por cada una de las técnicas de extracción de características y luego se adicionaron las instancias etiquetadas manualmente. En todos los casos se comparó la performance de los modelos generados a partir de la presencia de documentos etiquetados de forma automática con respecto a los modelos generados únicamente a partir de las instancias etiquetadas manualmente.

En cuanto a las técnicas de clasificación, se llevaron a cabo experimentos con máquinas de vector soporte (SVM) y el Clasificador Multiclase de *Bidirectional Encoder Representations from Transformer* (BERT). La razón de la elección de estas dos técnicas reside en que se busca

verificar el impacto de estas estrategias, tanto en técnicas tradicionales generales como SVM, así como en técnicas más recientes específicamente diseñadas en base a modelos de lenguaje como BERT.

Para el entrenamiento de los modelos, en el caso de **BERT**, se experimentó nuevamente con el modelo pre-entrenado nativo para el idioma español [18] y un conjunto de hiperparámetros utilizados con éxito en un trabajo anterior sobre los mismos datos, debido al tiempo de procesamiento necesario para la búsqueda de hiperparámetros en este tipo de modelos y la vasta cantidad de experimentos diseñados para esta investigación [29]. Por su parte, en el caso de SVM, se utilizó una búsqueda *grid* de los mejores hiperparámetros[18].

En cuanto a los datos utilizados para la etapa de entrenamiento, se diseñaron diferentes experiencias en función de la combinación de las instancias recuperadas mediante las estrategias de extracción de características y 800 de las etiquetadas manualmente.

Para la evaluación de los modelos, se reservaron nuevamente las 200 instancias restantes de las etiquetadas manualmente. Por último, el análisis de selección de los modelos generados se realizó a partir de las métricas de *accuracy*, *precision*, *f1-score* y *MCC* o *Matthews correlation coefficient* [22], especialmente indicativa en el caso de BERT.

Para los experimentos se utilizaron como conjuntos de entrenamiento las instancias etiquetadas manualmente así como las recuperadas con las tres estrategias de selección de características para 20, 50 y 100 instancias, con y sin boosting.

En primer lugar, a efectos de verificar la pertinencia de las instancias clasificadas automáticamente por las tres técnicas de selección de características, se entrenaron clasificadores a partir de las instancias etiquetadas manualmente por un lado, y se comparó la eficacia de estos modelos contra los modelos entrenados a partir de las instancias etiquetadas automáticamente, los cuales fueron detallados en la Sección 5.4.2. Los resultados obtenidos en términos del *accuracy* se presentan en la Tabla 5.6.

Tabla 5.6: Experimentos alternando las estrategias de extracción de características.

Estrategia	Etiquetado Manual	N=20	N=50	N=100	N=20 boosting	N=100 boosting
LR+SVM	0.810	0.510	0.615	0.665	0.520	0.665
TF-IDF+SVM	0.810	0.560	0.630	0.680	0.550	0.690[19]
SS3+SVM	0.810	0.600	0.600	0.655	0.580	0.655
LR+BERT	0.855	0.655	0.665	0.610	0.625	0.645
TF-IDF+BERT	0.855	0.650	0.720	0.720	0.640	0.720
SS3+BERT	0.855	0.610	0.600	0.655	0.715	0.645

Si bien resulta evidente que los modelos generados a partir de las instancias etiquetadas de forma manual son más efectivos que los generados únicamente a partir de las estrategias de selección de características, algunos indicios que dan pistas que estas estrategias podrían contribuir al etiquetado de instancias, cómo las métricas obtenidas para la estra-

18 Los hiperparámetros que se alternaron a lo largo de los experimentos son *C*, *Gamma* y los *kernels* utilizados por el algoritmo. Estos valores son especificados en las *notebooks* de los experimentos

19 Con N=200 muestra un *accuracy* de 0.74 mientras que el resto de las estrategias decrementa su rendimiento.

tegia de $TF - IDF + SVM$ para N=100 con *boosting* o algunas de las configuraciones para $TF - IDF + BERT$. A partir de este primer paso, se analizó si la capacidad de representar el conocimiento de las clases fue deficiente en los 16 casos o los resultados varían en función de las clases y las 3 estrategias aplicadas. Para ello, se comparó el *accuracy* obtenido por clase a partir de la matriz de confusión obtenida para cada método, utilizando 100 documentos recuperados con boosting por clase para las técnicas de extracción de características.

Tabla 5.7: Accuracy observado por clase para las estrategias de etiquetado y SVM

Clase	Etiquetado manual	LR	TF-IDF	SS3	Instancias
Boleto Universitario	**0.98**	0.88	0.90	0.90	48
Cambio de Carrera	0.50	**1.00**	0.50	**1.00**	2
Cambio de Comisión	0.50	0.50	0.50	0.50	2
Consulta Equivalencias	0.67	0.67	0.67	0.67	3
Consulta por Legajo	0.67	0.33	0.67	0.67	3
Consulta sobre Título	0.33	0.67	**1.00**	0.33	3
Cursadas	**0.89**	0.79	0.63	0.53	19
Datos Personales	0.00	0.50	**1.00**	0.60	2
Exámenes	0.60	0.60	**0.80**	**0.80**	5
Ingreso a la Universidad	**0.76**	0.52	0.36	0.42	33
Pedido de Certificados	0.93	0.93	0.93	0.93	15
Problemas con la Clave	**0.96**	0.65	0.87	0.70	23
Reincorporación	**0.73**	0.36	0.18	0.18	11
Requisitos de Ingreso	**0.67**	0.42	0.62	0.62	24
Simultaneidad	0.00	0.00	0.00	**1.00**	1
Situación Académica	0.50	0.67	**0.83**	**0.83**	6
Average	**0.810**	0.665	0.690	0.650	200

Los resultados presentados en las Tablas 5.6 y 5.7 permiten verificar que, si bien este etiquetado automático de los ejemplos de entrenamiento no es suficiente para reemplazar un modelo entrenado con el etiquetado manual, nutren al modelo de información que permitiría enriquecer el conjunto de entrenamiento conformado por las instancias etiquetadas manualmente para lograr una mejora en los resultados obtenidos.

Prueba de ello es el hecho que sólo en 6 de las 16 clases la estrategia de etiquetado manual haya obtenido el mayor *accuracy*, dejando entrever que las estrategias de selección de características probablemente capten parte de las características de los correos que conforman cada clase. Un punto saliente de los resultados obtenidos es que para todas las clases en que el conjunto de datos de prueba posee menos de 10 instancias (5 % del total), los modelos basados en el etiquetado automático de documentos funcionan mejor que el basado en etiquetado manual.

A partir de esta evidencia, y como siguiente paso, se construyó un sistema de votación entre las instancias recuperadas mediante las estrategias de extracción de características,

Tabla 5.8: Experimentos a partir de un sistema de votación entre LR, TF-IDF y SS3.

Estrategia	# Instancias	Accuracy	Recall	Precision
$(LR \cap TFIDF) + SVM$	925	0.635	0.635	0.787
$(LR \cap SS3) + SVM$	797	0.685	0.685	0.877
$(SS3 \cap TFIDF) + SVM$	1284	0.680	0.680	0.840
$(LR \cap TFIDF \cap SS3) + SVM$	520	0.615	0.615	0.760
$(LR \cap TFIDF) + BERT$	925	0.680	0.680	0.760
$(LR \cap SS3) + BERT$	797	0.615	0.615	0.647
$(SS3 \cap TFIDF) + BERT$	1284	0.720	0.720	0.827
$(LR \cap TFIDF \cap SS3) + BERT$	520	0.690	0.660	0.716

consolidando un nuevo conjunto de datos de entrenamiento conformado por las instancias que habían sido recuperadas por al menos dos de las tres estrategias para los primeros 100 resultados de cada clase.

Se encuentran resultados dispares respecto de utilizar las estrategias de selección de características de una a la vez para la obtención de las instancias de entrenamiento. Sin embargo, resulta interesante observar los valores relativamente elevados observados para algunos modelos en la métrica precisión, lo cual permite inferir que si bien los modelos generados a partir de las instancias etiquetadas con las estrategias de selección de características no permiten captar toda la varianza de las instancias de evaluación, son muy precisas para identificar las que clasifican en determinadas clases.

A continuación, se propone verificar si estas estrategias son capaces de mejorar la performance de los modelos de clasificación generados a partir de las instancias etiquetadas manualmente sumando a este conjunto de entrenamiento las etiquetadas de forma automática a partir de cada una de las tres estrategias y las combinaciones entre las mismas.

Entonces, se generó un nuevo lote de experimentos en el cual se adicionaron a las instancias etiquetadas de forma automática, las instancias originalmente etiquetadas de forma manual por los expertos a efectos de verificar si esta metodología robustece la clasificación de los casos de prueba.

En función de los resultados observados en los experimentos de la Tabla 5.9, se puede afirmar que todas las combinaciones entre las estrategias de selección de características, así como uno de los modelos generados por ellas tomados por separado, contribuyen a aportar más varianza al conjunto de instancias y se obtienen mejores modelos que los entrenados únicamente por las instancias etiquetadas manualmente. En líneas generales, se observa que estas estrategias resultan efectivas utilizadas a partir de la combinación de las estrategias de etiquetado, donde todos los modelos superan a los entrenados manualmente.

En particular, la estrategia que obtuvo los mejores resultados para la técnica de clasificación más tradicional, como son las máquinas de vectores de soporte (SVM), resulta de la combinación de SS3 y TF-IDF, obteniendo mejoras en términos de todas las métricas entre

Tabla 5.9: Experimentos con las instancias etiquetadas manualmente incorporadas a las etiquetadas automáticamente mediante las estrategias TF-IDF, LR y SS3.

Estrategia	Accuracy	F1-Score	Precision
$Manual + SVM$	0.810	0.809	0.829
$Manual + LR + SVM$	0.795	0.806	0.843
$Manual + TFIDF + SVM$	0.790	0.797	0.819
$Manual + SS3 + SVM$	0.790	0.792	0.829
$Manual + (LR \cap SS3) + SVM$	**0.820**	**0.821**	**0.842**
$Manual + (LR \cap TFIDF) + SVM$	**0.815**	**0.809**	**0.829**
$Manual + (SS3 \cap TFIDF) + SVM$	**0.830**	**0.833**	**0.856**
$Manual + (LR \cap TFIDF \cap SS3) + SVM$	**0.835**	**0.831**	**0.849**
$Manual + BERT$	0.860	0.847	0.845
$Manual + LR + BERT$	**0.875**	**0.876**	**0.889**
$Manual + TFIDF + BERT$	0.825	0.827	0.841
$Manual + SS3 + BERT$	0.825	0.837	0.868
$Manual + (LR \cap SS3) + BERT$	**0.890**	**0.887**	**0.893**
$Manual + (LR \cap TFIDF) + BERT$	**0.870**	**0.867**	**0.874**
$Manual + (SS3 \cap TFIDF) + BERT$	**0.895**	**0.890**	**0.898**
$Manual + (LR \cap TFIDF \cap SS3) + BERT$	**0.885**	**0.873**	**0.875**

el 2 % y el 3 %. También se observan mejoras para los modelos basados en transformadores (BERT), obteniendo el mejor modelo para la estrategia resultante de combinar SS3 y TFIDF, con mejoras entre el 4 % y el 6 % para las métricas evaluadas.

Particularmente, la estrategia que más incrementó su eficacia fue la técnica de clasificación basada en BERT entrenada a partir de la combinación de las instancias etiquetadas manualmente y las etiquetadas mediante la combinación entre SS3 y TFIDF.

5.5.4 *Síntesis del trabajo experimental*

En este capítulo se plasmó el trabajo exploratorio y experimental realizado en el marco de esta investigación.

En primer lugar, se diseñaron y procesaron un conjunto de atributos estáticos para analizar la composición y características del conjunto de datos a partir de un análisis gráfico y cuantitativo. De este análisis del conjunto de datos, se observa que existe mayor cantidad de consultas en los primeros días de la semana y el mes con más interacciones es febrero. n. A su vez, los proveedores de correo electrónico más utilizados en las consultas son *Hotmail* y *Gmail* y se observa que una gran cantidad de estudiantes remiten su número de teléfono ante la realización de las consultas aunque llama la atención que una gran proporción de ellos no indica su número de legajo entre los datos provistos. Respecto a los textos de las consultas, gran cantidad de ellas no superan los 100 caracteres, lo cual en términos de palabras está en torno a 20 y gran parte de las consultas poseen 2 oraciones o menos.

Luego, en términos del trabajo experimental, se etiquetaron 1000 correos electrónicos sobre más de 20000 procesados, dando origen, luego de una depuración de errores de etiquetado, a 16 clases distintas indicativas de las temáticas de consulta de los correos electrónicos. El conjunto de datos resultante posee un alto desbalanceo en cuanto a la cantidad de instancias etiquetadas para cada clase, lo cual a todas luces hizo más complejo el proceso de entrenamiento de modelos de clasificación con una performance aceptable.

A su vez los correos etiquetados se separaron en dos conjuntos en proporciones de 80 % y 20 % para entrenamiento y evaluación de los modelos respectivamente utilizando un muestreo estratificado.

Para la clasificación automática de estos correos electrónicos, se seleccionaron como técnicas a SVM y BERT. En una segunda iteración, luego del proceso de corrección de las etiquetas, se obtuvieron modelos con un *accuracy* que alcanzaba a 0.86 para BERT y 0.81 para SVM.

A continuación, se propuso una nueva estrategia para abordar el problema del desbalanceo de clases, consistente en la recuperación automática de nuevas instancias a partir de los correos procesados sin etiquetar, utilizando para ello tres estrategias de selección de características como SS3, los coeficientes resultantes del entrenamiento de regresores logísticos y la ponderación TF-IDF agrupada por clases.

A partir de los experimentos realizados inicialmente con las instancias recuperadas por esta estrategia únicamente, es posible inferir que la misma capta algunas de las características más importantes de las consultas de cada clase, logrando una alta precisión en algunos modelos pero con baja exhaustividad.

Luego, se realizaron nuevos experimentos utilizando las instancias etiquetadas manualmente en combinación con las instancias etiquetadas de forma automática a través de estrategias de selección de características, logrando mejores modelos en comparación de los anteriores, con mejoras de hasta un 6 % con respecto a los modelos entrenados sólo con las instancias etiquetadas manualmente.

6

CONCLUSIONES Y TRABAJOS FUTUROS

6.1 CONCLUSIONES

El objetivo general de este trabajo consistió en estudiar y analizar el conocimiento existente sobre técnicas aprendizaje automático aplicadas a la clasificación automática de textos, particularmente de correos electrónicos, y generar un modelo que aborde un problema concreto.

En este sentido, el correo electrónico, tal como lo conocemos hoy en día, ha seguido un proceso evolutivo constante desde su aparición e independientemente del momento puntual del surgimiento, resulta evidente que el proceso de masificación del acceso a Internet a partir de la década del '90 y particularmente en el 2000, generaron que el correo electrónico se constituya como el canal de comunicación asincrónica más importante de estos tiempos.

A su vez, las cifras y proyecciones muestran que el correo electrónico seguirá siendo una parte central de la vida digital diaria dado que, como se ha introducido en esta investigación, puede afirmarse que más de la mitad de los habitantes de este planeta utilizan correo electrónico actualmente y la cantidad de correos enviados diariamente asciende a los 293 billones, 24 billones más que hace solo dos años con un incremento proyectado en casi 54 billones hacia el año 2023.

Una vez dimensionada la utilización del correo electrónico y el impacto en nuestras vidas, resulta evidente que muchas áreas del conocimiento se muestran interesadas en extraer conocimiento a partir de esta información. Por su parte, y de la mano de la necesidad de obtener conocimiento de todas las fuentes de información disponibles, aparece a mediados de la década del 90, un campo emergente denominado descubrimiento de conocimiento en bases de datos (KDD), cuyo núcleo es la minería de datos.

Como se introdujo en este documento, existe un área particular de la minería de datos, denominada Minería de Textos, donde el conocimiento es generado mediante la adopción de bases de datos exclusivamente textuales como fuentes de datos. El correo electrónico como fuente de datos textual, posee un conjunto de características particulares respecto de otros elementos de texto que hace que existan diferencias y problemáticas particulares entre la minería de textos tradicional y la minería de correos electrónicos, conocida como *email mining*.

En este trabajo se aborda el desafío de analizar, describir y sistematizar el estado del arte del *email mining*, puntualmente el área que aborda la clasificación automática de correos electrónicos.

En este sentido, se trabajó en la explicitación de un proceso general para el tratamiento y clasificación automática de correos electrónicos, intentando categorizar esta problemática dentro de la disciplina general de Minería de Textos, la cual abarca las características y particularidades que se originan en esta forma de comunicación.

Se abordó aquí la construcción de un clasificador automático, proceso que está constituido por un conjunto de etapas como el etiquetado de documentos, extracción de características, entrenamiento del modelo, evaluación del modelo, utilización del modelo.

Una de las primeras tareas que se deben llevar a cabo es la clasificación inicial de un conjunto de documentos que luego serán utilizados como conjuntos de entrenamiento y prueba para el entrenamiento y validación del clasificador. Sobre esta temática, y con el objetivo de buscar alternativas para mejorar los modelos entrenados, se indagó en el estado del arte de la clasificación semi-supervisada de documentos, logrando, como una de las principales contribuciones de este trabajo, proponer una nueva estrategia para mejorar la performance de clasificación de los modelos entrenados.

La hipótesis acuñada para la definición de esta estrategia es que la selección de características representativas de cada clase, combinada con enfoques de recuperación de información, constituyen un método semi-supervisado válido y sencillo para la clasificación automática de correos electrónicos. Desde allí, se generó un proceso para la clasificación semi-supervisada de correos electrónicos a partir de la identificación de características claves de cada clase, utilizando tres técnicas de selección de características como la regresión logística, TF-IDF y SS3 y la posterior recuperación de correos etiquetados automáticamente a partir de un enfoque de recuperación de información con un motor de búsqueda de propósito general como *Elasticsearch*.

A nivel experimental, se utilizó un conjunto de datos con las consultas que realizan los estudiantes al staff administrativo de la Universidad Nacional de Luján sobre temas de índole académica. Se procesaron estos correos electrónicos y sus respectivas respuestas, asegurando la calidad de los datos y etiquetando, a partir de especialistas en el dominio, un subconjunto de los mismos con las temáticas que abordan.

Este subconjunto de datos etiquetados, y el resto de los correos preprocesados sin etiquetar constituyen dos conjuntos de datos curados consolidados que conforman una base de conocimiento disponible, con un proceso de curado y enriquecimiento a partir de la generación de atributos estáticos de más de 20000 documentos.

Por otro lado, se entrenó un modelo para la clasificación automática de estos correos electrónicos, abordando de manera concreta esta problemática, alcanzando un *accuracy* del 86 % en el mejor de los casos.

Además, se pudo verificar que las estrategias propuestas para el etiquetado semi-supervisado permiten identificar los términos más representativos de cada correo electrónico, al mismo tiempo que es posible utilizar las ponderaciones definidas por cada técnica para valorar la representatividad de esos términos para cada clase.

Asimismo, se demostró para el conjunto de datos con que se trabajó que estas técnicas de selección de características, utilizadas como estrategias de etiquetado automático a partir de un enfoque de recuperación de información para la clasificación semi-supervisada,

mejoran la capacidad de los clasificadores cuando se incorporan las instancias etiquetadas automáticamente a las etiquetadas de forma manual para entrenar el modelo, alcanzando mejoras de entre el 2 % y el 6 % para muchas de las estrategias abordadas en el marco de esta investigación.

Por último, se presentó, como apéndice, nuevamente la estrategia como una estrategia de *oversampling* para aplicar al aprendizaje automático en ambientes de datos desbalanceados. En este sentido, nuevamente se demostró, al menos para los datos con que se evaluó, que esta nueva estrategia presentada es competitiva en relación a las estrategias de remuestreo para el balanceo de clases, tanto de *undersampling* como de *oversampling*, arrojando valores más altos para las métricas de selección de modelos utilizadas y el conjunto de datos sobre el que se realizaron los experimentos.

6.2 TRABAJOS FUTUROS

Existen diversas lineas de investigación que quedan abiertas a partir de los resultados de este trabajo:

- Los correos electrónicos en general, y este conjunto de datos en particular, tienen características que no benefician a los modelos de clasificación automática dada la informalidad y los errores de sintaxis recurrentes propios de la dinámica de este medio de comunicación, por lo cual se espera que en conjuntos de datos con textos más depurados se obtengan modelos que logren mejores resultados, para lo cual sería interesante incorporar un proceso de corrección ortográfica sobre el corpus de documentos para la depuración del lenguaje.

- En este trabajo se propone un enfoque semi-supervisado para el etiquetado y clasificación de documentos y se demuestra que el mismo es un enfoque válido y sencillo, al menos para esta colección de documentos. Queda como desafío futuro abordar ajustes en este proceso sobre los parámetros involucrados, ya sea en términos de las técnicas de selección de características como el ajuste fino del N con la cantidad de términos representativos para cada clase y el impacto de su calibración en la eficiencia y eficacia de esta estrategia.

- A su vez, para la consolidación de la estrategia de aprendizaje semi-supervisado propuesta, quedan pendientes nuevas pruebas empíricas en otros conjuntos de datos que permitan ratificar la utilidad de la misma, al mismo tiempo que ayude a identificar los contextos más propicios para su utilización.

- Asimismo, en este trabajo se propone enfoque anterior también como estrategia de *oversampling* para la clasificación automática en un ambiente de datos con clases desbalanceadas. Si bien en el presente estudio se ha limitado la experimentación al dominio de la clasificación automática de correos electrónicos, creemos que la estrategia propuesta es generalizable a otros dominios donde existan disponibles documentos de textos sin etiquetar y, como trabajo futuro, se propone realizar un

nuevo trabajo aplicado a un contexto más general de clasificación automática de textos.

- Por otro lado, se plantea la posibilidad de avanzar en la implementación de un modelo de clasificación automática en un entorno de aplicación real, con el fin de realizar una transferencia concreta para facilitar gestión de las respuestas a las consultas en el contexto de la Universidad Nacional de Luján así como también, analizar la posibilidad de adaptar esta solución -al menos en términos del proceso de construcción propuesto- a otras problemáticas de índole similar.

A

APRENDIZAJE AUTOMÁTICO A PARTIR DE CONJUNTOS DE DATOS DESBALANCEADOS

A.1 MARCO TEÓRICO

Todas las estrategias de clasificación automática de textos -en mayor o menor medida- son sensibles a los problemas de desbalanceo entre clases. El desequilibrio o desbalanceo de clases está presente en muchos conjuntos de datos de clasificación del mundo real y consiste en una desproporción del número de ejemplos de las diferentes clases en el problema. Esta situación dificulta el rendimiento de los clasificadores debido a su diseño orientado a la exactitud -o *accuracy*-, lo que generalmente hace que se pase por alto la clase minoritaria [28].

Aún más general, la mayoría de los algoritmos de aprendizaje automático funcionan mejor cuando los conjuntos de datos están equilibrados, pero el problema surge cuando los conjuntos de datos dados están muy desequilibrados por naturaleza [83].

La clasificación de estos conjuntos de datos desequilibrados es una tarea muy compleja para los clasificadores tradicionales, ya que en general tienden a favorecer las muestras de las clases mayoritarias. Como resultado de la distribución desigual de los datos, la clase mayoritaria domina significativamente a la clase minoritaria. Para hacer frente a este problema de aprendizaje desequilibrado, se han desarrollado una gran cantidad de técnicas [71] y herramientas [59], las cuales se pueden categorizar en cuatro categorías principales, dependiendo de cómo aborden la solución [28]:

- Enfoques al nivel de algoritmo (también llamados internos): que intentan adaptar los algoritmos de aprendizaje para clasificación existentes con el objetivo de sesgar el aprendizaje hacia la clase minoritaria.
- Enfoques al nivel de datos (o externos): apuntan a reequilibrar la distribución de clases mediante el remuestreo del espacio de datos.
- Enfoques sensibles a costos: permiten definir costos asociados a cada una de las clases a efectos de generar una ponderación en la clasificación.
- Enfoques basados en ensambles: consisten en una combinación entre un algoritmo de aprendizaje basado en ensambles y una de las técnicas anteriores.

Uno de los más utilizados es el enfoque a nivel de datos, el cual está conformado por técnicas de remuestreo que se utilizan para balancear los datos ya sea submuestreando (*undersampling*) o sobremuestreando (*oversampling*) el conjunto de datos [71].

En primer lugar, el submuestreo o *undersampling* es el proceso de disminuir la cantidad de instancias o muestras objetivo mayoritarias. Algunos de los métodos más utilizados de *undersampling* consisten en la utilización del algoritmo KNN, técnicas de clustering o de ensamble. En el caso del algoritmo KNN (*k-nearest neighbors*), el mismo se utiliza para eliminar los datos donde la clase objetivo no es igual a la mayoría de sus instancias «vecinas más cercanas» [71]. Por su parte, la utilización del método de agrupación *k-means* tiene como objetivo equilibrar las instancias de las clases desbalanceadas disminuyendo la cantidad de las instancias mayoritarias [63]. A su vez, en los métodos de submuestreo aleatorio [19], generalmente las instancias de las clases mayoritarias se muestrean aleatoriamente sin reemplazo de la etiqueta para crear un conjunto de entrenamiento totalmente equilibrado [64]. Por último, se encuentran los métodos de ensamble tales como *EasyEnsemble* [98] donde la clase mayoritaria se divide en varios subconjuntos donde el tamaño de cada subconjunto es igual al tamaño de una clase minoritaria.

En segundo lugar, el sobremuestreo o *oversampling* consiste en aumentar la cantidad de instancias o muestras de las clases minoritarias con la producción de nuevas instancias o la repetición de las pre-existentes. La técnica más común se conoce como SMOTE (*Synthetic Minority Over-sampling Technique*) [20], donde, para sobremuestrear, se toma una muestra del conjunto de datos y se consideran los k vecinos más cercanos en función del espacio de características, creando un punto de datos sintético a partir de la multiplicación de uno de los vectores de características y un valor aleatorio, generalmente entre 0 y 1. Otro ejemplo de métodos de sobremuestreo es Borderline-SMOTE [41] cuyo objetivo es identificar muestras minoritarias ubicadas cerca del límite de decisión y utilizarlas para el sobremuestreo, evitando los potenciales riesgos de generalización excesiva que se dan con SMOTE. Por su parte, RAMOBoost (*Ranked Minority Oversampling in Boosting*) [21] es una técnica que genera sistemáticamente muestras sintéticas utilizando una distribución de probabilidad de muestreo ordenada. También existen otros enfoques de generación de muestras sintéticas, como ADASYN [45] y MWMOTE [9] que han obtenido buenos resultados a partir de modificaciones en los mecanismos de generación de los datos sintéticos.

Por último, algunos estudios plantean que la combinación de métodos de sobremuestreo y submuestreo permiten lograr un mejor rendimiento de los clasificadores que los métodos utilizados de forma separada [20].

La cantidad de abordajes propuestos para la solución de estos problemas permiten inferir la importancia del tema para la evolución de las técnicas supervisadas de aprendizaje automático.

No obstante, las técnicas basadas en *undersampling* no son una alternativa cuando las clases minoritarias cuentan con muy pocas instancias identificadas ya que está demostrado que para caracterizar con precisión la efectividad de este tipo de sistemas, los mismos deben evaluarse a la escala operativa en la que se utilizarán en la práctica [52]. Por su parte, la mayoría de las técnicas de *oversampling* se basan en la generación de nuevas

instancias sintéticas que no son parte de las observaciones reales, lo cual a todas luces parece una limitación.

Sin embargo, fundamentalmente producto de la masificación del acceso internet, se generan millones y millones de datos cada día, no siendo una restricción la cantidad de datos disponibles para el entrenamiento de los algoritmos de clasificación [26]. Es evidente que las limitaciones están dadas por la capacidad de etiquetar los datos disponibles ya que la estrategia tradicional para el etiquetado de documentos consiste en que esta tarea sea realizada por un humano, de forma manual. Si bien estas etiquetas de expertos proporcionan la piedra angular tradicional para evaluar los modelos de aprendizaje automático, el acceso limitado o costoso a los expertos representa un cuello de botella [52].

En este sentido, en el Capítulo 4 se plantea una nueva alternativa para el etiquetado semi-supervisado de instancias. Sin embargo, esta estrategia también puede aplicarse como una estrategia de remuestreo de datos de tipo *oversampling* donde se generan nuevas muestras ya no de forma artificial sino a partir de su identificación de instancias sin etiquetar en el conjunto de datos original. En este apéndice, se presenta una nueva propuesta de *oversampling*, utilizada antes como estrategia de etiquetado semi-supervisado [30], que consiste en partir de un conjunto de datos etiquetados manualmente y, mediante estrategias de selección de características, extraer términos representativos de cada clase minoritaria para recuperar nuevas instancias desde un repositorio de datos no etiquetados y así balancear el conjunto de datos con ejemplos no sintéticos.

A.2 METODOLOGÍA DE LA INVESTIGACIÓN

En este apartado, la propuesta consiste en presentar la propuesta del Capítulo A como una nueva estrategia de balanceo de clases para clasificación automática de textos y evaluar su desempeño en relación a estrategias de *oversampling* y *undersampling* ampliamente utilizadas en la comunidad científica. En la Figura A.1 se puede ver el esquema del proceso desarrollado, como una adaptación de la Figura 4.1.

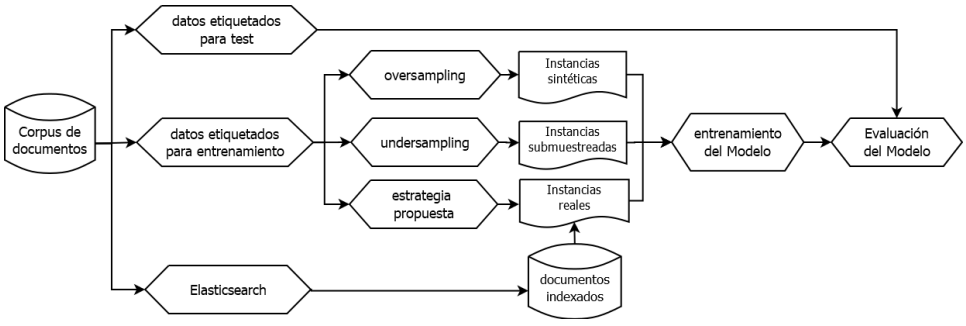

Figura A.1: Flujo de trabajo propuesto para el balanceo de clases

Para esta experiencia, se utilizaron los mismos conjuntos de datos de entrenamiento y validación que en el Capítulo 5.

Las estrategias de *oversampling* y *undersampling* se aplicaron directamente sobre el conjunto de entrenamiento para consolidar el conjunto de datos de entrenamiento.

Una vez consolidados los conjuntos de datos con las estrategias de balanceo de clases aplicadas, se entrenaron modelos de clasificación que se evalúan a partir del conjunto reservado para tal fin.

En las secciones siguientes se explican con mayor nivel de detalle las cuestiones más importantes relacionadas con el proceso desarrollado.

A.2.1 *Estrategias de balanceo de clases*

A continuación se presentan brevemente las tres estrategias utilizadas para el balanceo de clases previo al entrenamiento de los modelos para la generación de clasificadores automáticos.

A.2.1.1 *Estrategias de Oversampling.*

Las estrategias implementadas[1] fueron *RandomOverSampler*, *SMOTE*, *ADASYN* y *BorderSMOTE*. La primera estrategia, también conocida como ROSE (por el acrónimo en inglés de *random over sampling examples*), consiste en generar nuevas muestras mediante un muestreo aleatorio con reemplazo de las muestras actuales disponibles y se apoya en una base teórica respaldada por las propiedades de los métodos kernel [69]. Por su parte, SMOTE (*Synthetic Minority Over-sampling Technique*) es una de las estrategias más reconocidas de sobremuestreo, donde, a grandes rasgos, la clase minoritaria se sobremuestrea introduciendo ejemplos sintéticos en función de sus k vecinos más cercanos dependiendo de la cantidad de sobremuestreo requerida [20]. En este sentido, Borderline-SMOTE [41] es una variante de SMOTE que básicamente intenta determinar las instancias de las clases minoritarias que se encuentran en los límites y generar instancias sintéticas a partir de ellas. Por último, la idea esencial de ADASYN (*Adaptive Synthetic Sampling*) [45] es utilizar una distribución ponderada para los diferentes ejemplos de clases minoritarias según su nivel de dificultad en el aprendizaje, donde se generan más datos sintéticos para los ejemplos de clases minoritarias que son más difíciles de aprender.

A.2.1.2 *Estrategias de Undersampling.*

Las estrategias implementadas fueron *RandomUnderSampler*, *ClusterCentroids* y *EditedNearestNeighbours*. La primera estrategia elimina arbitrariamente instancias de la clase mayoritaria en el conjunto de datos de entrenamiento [43] mientras que en el caso de las estrategias basadas en *clustering* [63], se emplea un método de submuestreo basado en el reemplazo o eliminación de instancias por los centroides de las instancias de las clases minoritarias para reducir el número de muestras de datos de la clase mayoritaria.

Por su parte, la estrategia *Edited Nearest Neighbours* [99] aplica el algoritmo de vecinos más cercanos y «edita» el conjunto de datos eliminando las muestras que no coinciden «suficientemente» con su vecindad.

1 Las implementaciones se realizaron con la librería **Imbalanced-learn** para Python.

A.2.1.3 *Estrategia propuesta.*

La estrategia fue presentada inicialmente como una estrategia de clasificación semi-supervisada en el Capítulo 4 [30]. Para esta experiencia e utilizaron dos de las tres técnicas de selección de características explicadas allí, las cuales son TF-IDF y SS3 debido a que fueron las que mejores resultados arrojaron para técnicas de clasificación aptas para las estrategias de balanceo de clases abordadas.

Luego de recuperar los términos representativos por clase con ambas estrategias, con la base de conocimiento completa indexada en un motor de búsqueda de propósito general como *Elasticsearch*, se recuperan documentos de cada clase en función de las características detectadas por cada técnica y se consolida un nuevo conjunto de datos en función de las instancias que fueron recuperadas por ambas estrategias de selección de características.

Estas instancias son complementadas por las instancias de *dataset* de entrenamiento previo al entrenamiento del modelo de clasificación con el objetivo de balancearlo.

A.2.2 *Construcción del Modelo de clasificación*

En cuanto a las técnicas de clasificación, se utilizaron las máquinas de vector soporte (SVM) por su alto rendimiento para datos vectorizados dado que para las estrategias de remuestreo a implementar en general es necesario contar con datos vectorizados.

Para la validación de los modelos, se reservaron las 200 instancias restantes de las etiquetadas manualmente. Por último, el análisis de selección de los modelos generados se realizó a partir de las métricas de *accuracy*, *precision* y *f1-score*.

A.3 EXPERIMENTOS

Para los experimentos[2] se utilizó el conjunto de entrenamiento con las 800 instancias en todos los casos. Previo al entrenamiento, se vectorizaron las consultas utilizando 3-4-gramas de caracteres y una ponderación TF-IDF en todos los casos y luego se aplicaron las estrategias de balanceo de clases.

Es importante aclarar que en el caso de la estrategia propuesta, se recuperaron 200 instancias por cada clase y técnica de selección de características de la base de datos indexada en *Elasticsearch*, lo cual resultó en una limitación debido a que la cantidad de instancias resultantes del entrecruzamiento entre las instancias recuperadas por las dos técnicas hizo que en algunas clases no se alcance la cantidad de «equilibrio» requerida para el balanceo aunque disminuyó el desequilibrio existente. Se optó por esta opción por sobre la de recuperar un mayor número de instancias, con un *score* de coincidencia menor, para no introducir ruido en el conjunto de entrenamiento. Para paliar esta situación, se incorpora como variante de la estrategia propuesta la definición de un N alternativo menor de instancias, tal como el promedio disponible por clase, a efectos de reducir la distorsión.

2 Experimentos disponibles en `github.com/jumafernandez/imbalanced_data`

A continuación, se entrenaron clasificadores a partir de los conjuntos de datos balanceados a partir de las diferentes estrategias y se evaluó la performance de los modelos con las 200 instancias reservadas a tal fin, aplicando las métricas de *accuracy*, *F1-score* y *precision*. Los resultados se presentan en el Cuadro A.1.

Tabla A.1: Experimentos con técnicas de balanceo de clases

Estrategia	Accuracy	F1-Score	Precision
SVM (sin balanceo de clases)	0.810	0.80	0.82
RamdomOverSampler	0.810	0.80	0.81
SMOTE	0.805	0.79	0.81
ADASYN	0.810	0.80	0.81
BorderSMOTE	0.805	0.79	0.81
RamdomUnderSampler	0.660	0.68	0.73
ClusterCentroids	0.645	0.68	0.75
EditedNearestNeighbours	0.665	0.60	0.61
Estrategia propuesta	**0.820**	**0.83**	**0.85**
Estrategia propuesta (n = media = 115)	**0.820**	**0.83**	**0.85**

En función de los experimentos anteriores, se puede afirmar que ninguna de las técnicas preexistentes, ya sea de *oversampling* como de *undersampling*, pudieron mejorar los resultados obtenidos con el conjunto de datos original con las clases altamente desbalanceadas. Por su parte, se observa que la estrategia propuesta mejoró todas las métricas en sus dos variantes por igual.

A su vez, otra de las ventajas de la estrategia propuesta, al incorporar instancias no sintéticas al conjunto de datos de entrenamiento, reside en la posibilidad de utilizarla para los nuevos enfoques de clasificación basados en redes neuronales, ya sea de aquellos de *deep learning* así como los basados en *transformers*, limitación que sí se observa en las estrategias de balanceo basadas en ejemplos sintéticos en general. A continuación, se transcriben los resultados de ejecutar los experimentos en BERT (*Bidirectional Encoder Representations from Transformers*)[3] [25].

Tabla A.2: Experimentos con técnicas de balanceo de clases con BERT

Estrategia	Accuracy	F1-Score	Precision
BERT (sin balanceo de clases)	0.860	0.847	0.845
Estrategia propuesta	**0.865**	**0.865**	**0.878**
Estrategia propuesta (n = media = 115)	0.840	0.837	0.854

En el Cuadro A.2 puede observarse que la estrategia propuesta sigue siendo efectiva pero sólo para el abordaje convencional. En el caso de la variante por la media de instancias por clase los resultados son inferiores para las métricas *accuracy* y *F1-score*, entre un 1 % y 2 % y superior en proporciones similares para la *precision*.

3 Para el entrenamiento de los modelos, se experimentó con un modelo pre-entrenado nativo para el idioma español [18] y un conjunto de hiperparámetros utilizados con éxito en un trabajo anterior [29].

A.4 REFLEXIONES FINALES

Este apartado presenta una nueva estrategia para el balanceo de clases en problemas de clasificación automática de textos a partir del sobremuestreo de clases a partir de la recuperación de nuevas instancias no etiquetadas desde un repositorio de datos de la misma naturaleza que los datos etiquetados.

El hecho que las instancias para el remuestreo provengan de instancias reales se presenta como una ventaja sobre las estrategias que generan muestras sintéticas. En principio, puede aparecer como una debilidad el hecho de tener que contar con un repositorio adicional de datos para la experimentación, sin embargo, en problemas de escala real es normal que exista un repositorio de datos amplio -aunque no etiquetado- disponible.

Otra de las ventajas de la estrategia propuesta, al incorporar instancias no sintéticas al conjunto de datos de entrenamiento, reside en la posibilidad de utilizarla para los nuevos enfoques de clasificación basados en redes neuronales, ya se aquellos de *deep learning* como los basados en *transformers*, limitación que sí se observa en las estrategias de balanceo basadas en ejemplos sintéticos en general.

En función de los resultados obtenidos, puede concluirse que esta nueva estrategia es competitiva con respecto a otras estrategias de remuestreo utilizadas ampliamente en la comunidad científica, ya sea para enfoques de clasificación tradicionales, como el propuesto para SVM, como para los nuevos enfoques basados en *transformers*, como BERT.

Por último, si bien en el presente estudio se ha limitado la experimentación al dominio de los correos electrónicos, creemos que la estrategia propuesta es generalizable a otros dominios donde existan disponibles documentos de textos sin etiquetar y, como trabajo futuro, se propone realizar un nuevo trabajo aplicado a un contexto más general de clasificación automática de textos.

BIBLIOGRAFÍA

[1] Ashutosh Adhikari, Achyudh Ram, Raphael Tang y Jimmy Lin. «Docbert: Bert for document classification». En: *arXiv preprint arXiv:1904.08398* (2019).

[2] Charu C Aggarwal y col. «Neural networks and deep learning». En: *Springer* 10 (2018).

[3] Alyaa Alfalahi, Gunnar Eriksson y Eriks Sneiders. «Shadow answers as an intermediary in email answer retrieval». En: *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer. 2015, págs. 209-214.

[4] Ahmed Alghoul, Sara Al Ajrami, Ghada Al Jarousha, Ghayda Harb y Samy S Abu-Naser. «Email classification using artificial neural network». En: *ACM* (2018).

[5] Rayan Salah Hag Ali y Neamat El Gayar. «Sentiment Analysis using Unlabeled Email data». En: *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*. IEEE. 2019, págs. 328-333.

[6] Vicente Trigo Aranda. «Historia y evolución de Internet». En: *Autores Cientifico-tecnicos y académicos* (2004), págs. 22-32.

[7] Prafulla Bafna, Dhanya Pramod y Anagha Vaidya. «Document clustering: TF-IDF approach». En: *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. IEEE. 2016, págs. 61-66.

[8] Dzmitry Bahdanau, Kyunghyun Cho y Yoshua Bengio. «Neural machine translation by jointly learning to align and translate». En: *arXiv preprint arXiv:1409.0473* (2014).

[9] Sukarna Barua, Md Monirul Islam, Xin Yao y Kazuyuki Murase. «MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning». En: *IEEE Transactions on knowledge and data engineering* 26.2 (2012), págs. 405-425.

[10] Thomas Bayes. «LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S». En: *Philosophical transactions of the Royal Society of London* 53 (1763), págs. 370-418.

[11] Yoshua Bengio, Réjean Ducharme, Pascal Vincent y Christian Janvin. «A neural probabilistic language model». En: *The journal of machine learning research* 3 (2003), págs. 1137-1155.

[12] Pranjal S Bogawar y Kishor K Bhoyar. «Email mining: a review». En: *International Journal of Computer Science Issues(IJCSI)* 9.1 (2012).

[13] Ronald J Brachman y Tej Anand. «The process of knowledge discovery in databases. Advances in Knowledge Discovery and Data Mining». En: *American Association for Artificial Intelligence*. 1996, págs. 37-57.

[14] Max Bramer. *Principles of data mining*. Vol. 180. Springer, 2016.

[15] Sergio G Burdisso, Marcelo Errecalde y Manuel Montes-y Gómez. «A text classification framework for simple and effective early depression detection over social media streams». En: *Expert Systems with Applications* 133 (2019), págs. 182-197.

[16] Cristian Cardellino. *Spanish Billion Words Corpus and Embeddings*. Ago. de 2019. URL: https://crscardellino.github.io/SBWCE/.

[17] Marina E Cardenas, Julio J Castillo, Martin Navarro, Nicolás Hernández y Marisa Velazco. «Herramientas para el desarrollo de sistemas de análisis de textos no estructurados». En: *XXI Workshop de Investigadores en Ciencias de la Computación (WICC 2019, Universidad Nacional de San Juan)*. 2019.

[18] José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang y Jorge Pérez. «Spanish Pre-Trained BERT Model and Evaluation Data». En: *PML4DC at ICLR 2020*. 2020.

[19] Nitesh V Chawla. «Data mining for imbalanced datasets: An overview». En: *Data mining and knowledge discovery handbook* (2009), págs. 875-886.

[20] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall y W Philip Kegelmeyer. «SMOTE: synthetic minority over-sampling technique». En: *Journal of artificial intelligence research* 16 (2002), págs. 321-357.

[21] Sheng Chen, Haibo He y Edwardo A Garcia. «RAMOBoost: Ranked minority oversampling in boosting». En: *IEEE Transactions on Neural Networks* 21.10 (2010), págs. 1624-1642.

[22] Davide Chicco y Giuseppe Jurman. «The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation». En: *BMC genomics* 21.1 (2020), págs. 1-13.

[23] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk y Yoshua Bengio. «Learning phrase representations using RNN encoder-decoder for statistical machine translation». En: *arXiv preprint arXiv:1406.1078* (2014).

[24] Cyril W Cleverdon, Jack Mills y E Michael Keen. «Factors determining the performance of indexing systems,(Volume 1: Design)». En: *Cranfield: College of Aeronautics* 28 (1966).

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee y Kristina Toutanova. «Bert: Pre-training of deep bidirectional transformers for language understanding». En: *arXiv preprint arXiv:1810.04805* (2018).

[26] Fanny Fanny, Yohan Muliono y Fidelson Tanzil. «A comparison of text classification methods k-NN, Naive Bayes, and support vector machine for news classification». En: *Jurnal Informatika: Jurnal Pengembangan IT* 3.2 (2018), págs. 157-160.

[27] Usama Fayyad, Gregory Piatetsky-Shapiro y Padhraic Smyth. «From data mining to knowledge discovery in databases». En: *AI magazine* 17.3 (1996), págs. 37-37.

[28] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk y Francisco Herrera. *Learning from imbalanced data sets*. Vol. 10. Springer, 2018.

[29] Juan M Fernandez, Nicolás Cavasin y Marcelo Errecalde. «Classic and recent (neural) approaches to automatic text classification: a comparative study with e-mails in the Spanish language». En: *Short Papers of the 9th Conference on Cloud Computing, Big Data & Emerging Topics*. 2021, pág. 20.

[30] Juan M Fernandez y Marcelo Errecalde. «Multi-class e-mail classification with a semi-supervised approach based on automatic feature selection and information retrieval». En: *Full Papers of the 10th Conference on Cloud Computing, Big Data & Emerging Topics*. 2022, pág. 20.

[31] Raúl Fernández Regalado. «El teorema de Bayes y su utilización en la interpretación de las pruebas diagnósticas en el laboratorio clínico». En: *Revista cubana de investigaciones biomédicas* 28.3 (2009), págs. 158-165.

[32] Edgardo Ferretti, Marcelo L Errecalde, Maik Anderka y Benno Stein. «On the use of reliable-negatives selection strategies in the pu learning approach for quality flaws prediction in wikipedia». En: *2014 25th International Workshop on Database and Expert Systems Applications*. IEEE. 2014, págs. 211-215.

[33] Edgardo Ferretti, Donato Hernández Fusilier, Rafael Guzmán Cabrera, Manuel Montes y Gómez, Marcelo Errecalde y Paolo Rosso. «On the Use of PU Learning for Quality Flaw Prediction in Wikipedia». En: *CEUR Workshop Proceedings*. Vol. 1178. 2012.

[34] Internet Engineering Task Force. *RFCs.* https://www.ietf.org/standards/rfcs/. 2019. (Visitado 02-08-2021).

[35] George Forman y col. «An extensive empirical study of feature selection metrics for text classification.» En: *J. Mach. Learn. Res.* 3.Mar (2003), págs. 1289-1305.

[36] Jerome H Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. springer open, 2017.

[37] Alec Go, Richa Bhayani y Lei Huang. «Twitter sentiment classification using distant supervision». En: *CS224N project report, Stanford* 1.12 (2009), pág. 2009.

[38] Cyril Goutte y Eric Gaussier. «A probabilistic interpretation of precision, recall and F-score, with implication for evaluation». En: *European conference on information retrieval*. Springer. 2005, págs. 345-359.

[39] The Radicati Group. *Email Statistics Report, 2019-2023.* url: http://www.radicati.com. 2019. (Visitado 02-11-2021).

[40] Itisha Gupta y Nisheeth Joshi. «Real-time twitter corpus labelling using automatic clustering approach». En: *International Journal of Computing and Digital Systems* 10 (2021), págs. 519-532.

[41] Hui Han, Wen-Yuan Wang y Bing-Huan Mao. «Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning». En: *International conference on intelligent computing*. Springer. 2005, págs. 878-887.

[42] Jiawei Han, Jian Pei y Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[43] Mohamed Hanafy y Ruixing Ming. «Improving imbalanced data classification in auto insurance by the data level approaches». En: *International Journal of Advanced Computer Science and Applications* 12.6 (2021).

[44] Bhat S Harish, Devanur S Guru y Shantharamu Manjunath. «Representation and classification of text documents: A brief review». En: *IJCA, Special Issue on RTIPPR (2)* (2010), págs. 110-119.

[45] Haibo He, Yang Bai, Edwardo A Garcia y Shutao Li. «ADASYN: Adaptive synthetic sampling approach for imbalanced learning». En: *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE. 2008, págs. 1322-1328.

[46] Martin Henkel, Erik Perjons y Eriks Sneiders. «Examining the potential of language technologies in public organizations by means of a business and IT architecture model». En: *International Journal of Information Management* 37.1 (2017), págs. 1507-1516.

[47] Laura Igual y Santi Seguí. «Introduction to Data Science». En: *Introduction to Data Science*. Springer, 2017, págs. 1-4.

[48] Md R Islam, Morshed U Chowdhury y Wanlei Zhou. «An innovative spam filtering model based on support vector machine». En: *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*. Vol. 2. IEEE. 2005, págs. 348-353.

[49] Thorsten Joachims. «Optimizing search engines using clickthrough data». En: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002, págs. 133-142.

[50] Thorsten Joachims y col. «Transductive inference for text classification using support vector machines». En: *Icml*. Vol. 99. 1999, págs. 200-209.

[51] Guillermo Jorge-Botana, Jose A Leon, Ricardo Olmos e Inmaculada Escudero. «Latent semantic analysis parameters for essay evaluation using small-scale corpora». En: *Journal of Quantitative Linguistics* 17.1 (2010), págs. 1-29.

[52] Hyun Joon Jung y Matthew Lease. «Evaluating classifiers without expert labels». En: *arXiv preprint arXiv:1212.0960* (2012).

[53] Giuseppe Jurman, Samantha Riccadonna y Cesare Furlanello. «A comparison of MCC and CEN error measures in multi-class prediction». En: (2012).

[54] Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee y Khairullah Khan. «A review of machine learning algorithms for text-documents classification». En: *Journal of advances in information technology* 1.1 (2010), págs. 4-20.

[55] Miroslav Kubat. *An introduction to machine learning*. Springer, 2017.

[56] Robert Layton, Paul Watters y Richard Dazeley. «Recentred local profiles for authorship attribution». En: *Natural Language Engineering* 18.3 (2012), págs. 293-312.

[57] Junseok Lee, Ji-Ho Kang, Sunghae Jun, Hyunwoong Lim, Dongsik Jang y Sangsung Park. «Ensemble modeling for sustainable technology transfer». En: *Sustainability* 10.7 (2018), pág. 2278.

[58] Barry M Leiner, Vinton G Cerf, David D Clark, Robert E Kahn, Leonard Kleinrock, Daniel C Lynch, Jon Postel, Lawrence G Roberts y Stephen Wolff. «Una breve historia de Internet». En: *Revista Novática. Números* 130 (1999), pág. 131.

[59] Guillaume Lematre, Fernando Nogueira y Christos K Aridas. «Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning». En: *The Journal of Machine Learning Research* 18.1 (2017), págs. 559-563.

[60] Elisabeth Lex. «Content Facets for Individual Information Needs in Media». En: (2011).

[61] Wenjuan Li, Weizhi Meng, Zhiyuan Tan y Yang Xiang. «Design of multi-view based email classification for IoT systems via semi-supervised learning». En: *Journal of Network and Computer Applications* 128 (2019), págs. 56-63.

[62] Zhixing Li, Zhongyang Xiong, Yufang Zhang, Chunyong Liu y Kuan Li. «Fast text categorization using concise semantic analysis». En: *Pattern Recognition Letters* 32.3 (2011), págs. 441-448.

[63] Wei-Chao Lin, Chih-Fong Tsai, Ya-Han Hu y Jing-Shang Jhang. «Clustering-based undersampling in class-imbalanced data». En: *Information Sciences* 409 (2017), págs. 17-26.

[64] Bin Liu y Grigorios Tsoumakas. «Dealing with class imbalance in classifier chains via random undersampling». En: *Knowledge-Based Systems* 192 (2020), pág. 105292.

[65] Bing Liu, Xiaoli Li, Wee Sun Lee y Philip S Yu. «Text classification by labeling words». En: *AAAI*. Vol. 4. 2004, págs. 425-430.

[66] Sisi Liu e Ickjai Lee. «Email sentiment analysis through k-means labeling and support vector machine classification». En: *Cybernetics and Systems* 49.3 (2018), págs. 181-199.

[67] Luciana Mariñelarena-Dondena, Marcelo Luis Errecalde y Alejandro Castro Solano. «Extracción de conocimiento con técnicas de minería de textos aplicadas a la psicología». En: *Revista Argentina de Ciencias del Comportamiento* 9.2 (2017), págs. 65-76.

[68] Andrew McCallum, Kamal Nigam y col. «A comparison of event models for naive bayes text classification». En: *AAAI-98 workshop on learning for text categorization*. Citeseer. 1998, págs. 41-48.

[69] Giovanna Menardi y Nicola Torelli. «Training and assessing classification rules with imbalanced data». En: *Data mining and knowledge discovery* 28.1 (2014), págs. 92-122.

[70] Tomas Mikolov, Kai Chen, Greg Corrado y Jeffrey Dean. «Efficient estimation of word representations in vector space». En: *arXiv preprint arXiv:1301.3781* (2013).

[71] Roweida Mohammed, Jumanah Rawashdeh y Malak Abdullah. «Machine learning with oversampling and undersampling techniques: overview study and experimental results». En: *2020 11th international conference on information and communication systems (ICICS)*. IEEE. 2020, págs. 243-248.

[72] I Peter. *The history of email. Internet History Project*. 2004.

[73] Kenneth Price, Rainer M Storn y Jouni A Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[74] Anita Rácz, Dávid Bajusz y Károly Héberger. «Multi-level comparison of machine learning classifiers and their performance metrics». En: *Molecules* 24.15 (2019), pág. 2811.

[75] Sebastian Raschka. «Model evaluation, model selection, and algorithm selection in machine learning». En: *arXiv preprint arXiv:1811.12808* (2018).

[76] Jonathon Read. «Using emoticons to reduce dependency in machine learning techniques for sentiment classification». En: *Proceedings of the ACL student research workshop*. 2005, págs. 43-48.

[77] J Russell Stuart y Peter Norvig. *Artificial intelligence: a modern approach*. Prentice Hall, 2009.

[78] Soumyabrata Saha, Suparna DasGupta y Suman Kumar Das. «Spam mail detection using data mining: A comparative analysis». En: *Smart Intelligent Computing and Applications*. Springer, 2021, págs. 571-580.

[79] Gerard Salton y Christopher Buckley. «Term-weighting approaches in automatic text retrieval». En: *Information processing & management* 24.5 (1988), págs. 513-523.

[80] Gerard Salton, Anita Wong y Chung-Shu Yang. «A vector space model for automatic indexing». En: *Communications of the ACM* 18.11 (1975), págs. 613-620.

[81] Susan Schreibman, Ray Siemens y John Unsworth. *A new companion to digital humanities*. John Wiley & Sons, 2015.

[82] Fabrizio Sebastiani. «Machine learning in automated text categorization». En: *ACM computing surveys (CSUR)* 34.1 (2002), págs. 1-47.

[83] Mayuri S Shelke, Prashant R Deshmukh y Vijaya K Shandilya. «A review on imbalanced data handling using undersampling and oversampling technique». En: *Int. J. Recent Trends Eng. Res* 3.4 (2017), págs. 444-449.

[84] Nadia Felix F Da Silva, Luiz FS Coletta y Eduardo R Hruschka. «A survey and comparative study of tweet sentiment analysis via semi-supervised learning». En: *ACM Computing Surveys (CSUR)* 49.1 (2016), págs. 1-26.

[85] Steven S Skiena. *The data science design manual*. Springer, 2017.

[86] Eriks Sneiders. «Review of the main approaches to automated email answering». En: *New advances in information systems and technologies*. Springer, 2016, págs. 135-144.

[87] Eriks Sneiders, Jonas Sjöbergh y Alyaa Alfalahi. «Automated email answering by text-pattern matching: Performance and error analysis». En: *Expert Systems* 35.1 (2018), e12251.

[88] Ge Song, Yunming Ye, Xiaolin Du, Xiaohui Huang y Shifu Bie. «Short text classification: A survey». En: *Journal of multimedia* 9.5 (2014), pág. 635.

[89] Statista. *Most popular global mobile messenger apps as of July 2019, based on number of monthly active users (in millions).* url: http://www.statista.com/. 2019. (Visitado 02-11-2021).

[90] Chi Sun, Xipeng Qiu, Yige Xu y Xuanjing Huang. «How to fine-tune bert for text classification?» En: *China National Conference on Chinese Computational Linguistics*. Springer. 2019, págs. 194-206.

[91] Guanting Tang, Jian Pei y Wo-Shun Luk. «Email mining: tasks, common techniques, and tools». En: *Knowledge and Information Systems* 41.1 (2014), págs. 1-31.

[92] Gabriel H Tolosa y Fernando RA Bordignon. *Introducción a la Recuperación de Información*. 2008.

[93] Ray Tomlinson. «The first network email». En: *Site de Ray Tomlinson* (2009).

[94] Mrityunjay Upadhyay, Divakar Radhakrishnan y Madhusudhanan Natarajan. *Summarization and processing of email on a client computing device based on content contribution to an email thread using weighting techniques*. US Patent 10,102,192. 2018.

[95] Antonio Usai, Marco Pironti, Monika Mital y Chiraz Aouina Mejri. «Knowledge discovery out of text data: a systematic review via text mining». En: *Journal of knowledge management* (2018).

[96] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser e Illia Polosukhin. «Attention is all you need». En: *arXiv preprint arXiv:1706.03762* (2017).

[97] Augusto Villa Monte. «Resumen de tesis: Generación automática inteligente de resúmenes de textos con técnicas de soft computing». En: *XXII Workshop de Investigadores en Ciencias de la Computación (WICC 2020, El Calafate, Santa Cruz)*. 2020.

[98] Tao Wang, Changhua Lu, Wei Ju y Chun Liu. «Imbalanced heartbeat classification using EasyEnsemble technique and global heartbeat information». En: *Biomedical Signal Processing and Control* 71 (2022), pág. 103105.

[99] Dennis L Wilson. «Asymptotic properties of nearest neighbor rules using edited data». En: *IEEE Transactions on Systems, Man, and Cybernetics* 3 (1972), págs. 408-421.

[100] Rüdiger Wirth y Jochen Hipp. «CRISP-DM: Towards a standard process model for data mining». En: *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. Vol. 1. Springer-Verlag London, UK. 2000.

[101] Lingfei Wu, Ian EH Yen, Kun Xu, Fangli Xu, Avinash Balakrishnan, Pin-Yu Chen, Pradeep Ravikumar y Michael J Witbrock. «Word mover's embedding: From word2vec to document embedding». En: *arXiv preprint arXiv:1811.01713* (2018).

[102] Savaş YILDIRIM y Tuğba YILDIZ. «A Comparison of Different Approaches to Document Representation in Turkish Language». En: *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi* 22.2 (2018), págs. 569-576.

[103] Jing Zhao, Xijiong Xie, Xin Xu y Shiliang Sun. «Multi-view learning overview: Recent progress and new challenges». En: *Information Fusion* 38 (2017), págs. 43-54.

[104] Rong Zheng, Jiexun Li, Hsinchun Chen y Zan Huang. «A framework for authorship identification of online messages: Writing-style features and classification techniques». En: *Journal of the American society for information science and technology* 57.3 (2006), págs. 378-393.

[105] Zhi-Hua Zhou, De-Chuan Zhan y Qiang Yang. «Semi-supervised learning with very few labeled training examples». En: *AAAI*. Vol. 675680. 2007.