



INSTITUTO POLITÉCNICO NACIONAL



CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Estudio de la intención de noticias y
estimación automática de su impacto

TESIS

QUE PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

Ing. Cesar Macias Sanchez

DIRECTORES DE TESIS:

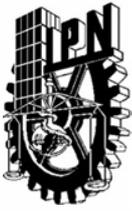
Dr. Francisco Hiram Calvo Castro

Dr. Joel Omar Juárez Gambino



CIUDAD DE MÉXICO

DICIEMBRE 2022



INSTITUTO POLITÉCNICO NACIONAL SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REGISTRO DE TEMA DE TESIS Y DESIGNACIÓN DE DIRECTOR DE TESIS

Ciudad de México, a 25 de agosto del 2021

El Colegio de Profesores de Posgrado del **Centro de Investigación en Computación** en su Sesión (Unidad Académica)

Ordinaria No. 05 celebrada el día 28 del mes mayo de 2021, conoció la solicitud presentada por el (la) alumno (a):

Apellido Paterno:	MACIAS	Apellido Materno:	SANCHEZ	Nombre (s):	CESAR
-------------------	--------	-------------------	---------	-------------	-------

Número de registro: A 2 1 0 2 3 4

del Programa Académico de Posgrado: **Maestría en Ciencias de la Computación**

Referente al registro de su tema de tesis; acordando lo siguiente:

1.- Se designa al aspirante el tema de tesis titulado:

"Estudio de la intención de noticias y estimación automática de su impacto"

Objetivo general del trabajo de tesis:

Desarrollar un modelo computacional que clasifique noticias de acuerdo con su intención y estime de manera automática su impacto en aquellos que las consulten.

2.- Se designa como Directores de Tesis a los profesores:

Director: **Dr. Francisco Hiram Calvo Castro** 2° Director: **Dr. Joel Omar Juárez Gambino**

No aplica:

3.- El Trabajo de investigación base para el desarrollo de la tesis será elaborado por el alumno en:

Centro de Investigación en Computación

que cuenta con los recursos e infraestructura necesarios.

4.- El interesado deberá asistir a los seminarios desarrollados en el área de adscripción del trabajo desde la fecha en que se suscribe la presente, hasta la aprobación de la versión completa de la tesis por parte de la Comisión Revisora correspondiente.

Director de Tesis

Dr. Francisco Hiram Calvo Castro

Aspirante

Cesar Macias Sanchez

2° Director de Tesis

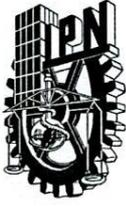
Omar Juárez Gambino
Dr. Joel Omar Juárez Gambino

Presidente del Colegio

Dr. Marco Antonio Moreno Ibarra



INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN
EN COMPUTACIÓN
DIRECCIÓN



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 16:00 horas del día 25 del mes de noviembre del 2022 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Posgrado de: Centro de Investigación en Computación para examinar la tesis titulada: “Estudio de la intención de noticias y estimación automática de su impacto “ del (la) alumno (a):

Apellido Paterno:	MACIAS	Apellido Materno:	SANCHEZ	Nombre (s):	CESAR
-------------------	--------	-------------------	---------	-------------	-------

Número de registro: A 2 1 0 2 3 4

Aspirante del Programa Académico de Posgrado: Maestría en Ciencias de la Computación

Una vez que se realizó un análisis de similitud de texto, utilizando el software antiplagio, se encontró que el trabajo de tesis tiene 09 % de similitud. **Se adjunta reporte de software utilizado.**

Después que esta Comisión revisó exhaustivamente el contenido, estructura, intención y ubicación de los textos de la tesis identificados como coincidentes con otros documentos, concluyó que en el presente trabajo SI NO SE CONSTITUYE UN POSIBLE PLAGIO.

JUSTIFICACIÓN DE LA CONCLUSIÓN: *(Por ejemplo, el % de similitud se localiza en metodologías adecuadamente referidas a fuente original)*
La mayoría de las coincidencias aparecen en frases cortas comunes en la literatura.

****Es responsabilidad del alumno como autor de la tesis la verificación antiplagio, y del Director o Directores de tesis el análisis del % de similitud para establecer el riesgo o la existencia de un posible plagio.**

Finalmente y posterior a la lectura, revisión individual, así como el análisis e intercambio de opiniones, los miembros de la Comisión manifestaron **APROBAR** **SUSPENDER** **NO APROBAR** la tesis por **UNANIMIDAD** o **MAYORÍA** en virtud de los motivos siguientes:
Cumple con todos los requisitos de una tesis de maestría .

COMISIÓN REVISORA DE TESIS

Dr. Francisco Hiram Calvo Castro
Director de Tesis

Dr. Alexander Gelbukh

M. en C. José Juan Guzmán Camacho

Dr. Joel Omar Juárez Gambino
2° Director de Tesis

Dr. Rolando Merchaca Méndez

Dr. Luis Manuel Vilches Blázquez

Dr. Francisco Hiram Calvo Castro
PRESIDENTE DEL COLEGIO DE PROFESORES
IPN-CIC



INSTITUTO POLITÉCNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA DE AUTORIZACIÓN DE USO DE OBRA PARA DIFUSIÓN

En la Ciudad de México el día 12 del mes de diciembre del año 2022, el (la) que suscribe Cesar Macias Sanchez alumno(a) del programa Maestría en Ciencias de la Computación con número de registro A210234, adscrito(a) a Centro de Investigación en Computación manifiesta que es autor(a) intelectual del presente trabajo de tesis bajo la dirección de Dr. Francisco Hiram Calvo Castro y Dr. Joel Omar Juárez Gambino y cede los derechos del trabajo intitulado "Estudio de la intención de noticias y estimación automática de su impacto" al Instituto Politécnico Nacional, para su difusión con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expresado del autor y/o director(es). Este puede ser obtenido escribiendo a las siguiente(s) dirección(es) de correo. cmaciass2021@cic.ipn.mx buick.macias.96@gmail.com. Si el permiso se otorga, al usuario deberá dar agradecimiento correspondiente y citar la fuente de este.


Cesar Macias Sanchez
Nombre completo y firma autográfica del (de la) estudiante

Resumen

Desde la invención de la Internet, y con la popularización de las redes sociales, la información está disponible para todos los usuarios al alcance de un click y en grandes cantidades. Una de las principales fuentes de información en las redes sociales son las noticias. De entre las posibles opciones en las que los usuarios expresan su opinión o comentan sobre algún tema de su interés, Twitter se ha convertido en una herramienta en la que pueden expresar sus pensamientos, lo que hace de los tuits una gran fuente de datos. Al analizar los comentarios de los usuarios sobre una noticia, es posible determinar su polaridad, pero si tenemos la polaridad del tuit, ¿de qué forma podemos utilizarla para predecir la controversia? Es entonces que surge la pregunta ¿Cómo podemos enseñarle a la computadora a decidir si una noticia es controversial o no? Más aún, ya que queremos clasificar noticias controversiales desde la perspectiva de sus lectores ¿podemos utilizar la polaridad de los comentarios de la noticia para predecir dicha controversialidad? Estas preguntas se responden a lo largo del presente trabajo. La controversia nos dice cómo es que la noticia ha afectado a sus lectores, ya que si las respuestas son controversiales entonces podemos inferir que la opinión de los lectores difirió de gran manera, y que no hubo un común acuerdo sobre la noticia indicando que en los comentarios se dió una acalorada discusión sobre las opiniones de quienes comentaron.

Abstract

Since the invention of the Internet, and with the popularization of social networks, information is available to all users at the click of a button and in large quantities. One of the main sources of information on social networks is news. Among the possible options for users to express their opinion or comment on a topic of their interest, Twitter has become a tool where they can express their thoughts, making tweets a great source of data. By analyzing user comments on a news item, it is possible to determine its polarity, but if we have the polarity of the tweet, how can we use it to predict the controversy? It is then that the question arises, how can we teach the computer to decide whether a news item is controversial or not? Moreover, since we want to classify controversial news from the perspective of its readers can we use the polarity of the news comments to predict such controversy? These questions are answered throughout this work. Controversy tells us how the news has affected its readers, since if the responses are controversial then we can infer that the opinion of the readers differed greatly, and that there was no common agreement on the news indicating that in the comments there was a heated discussion about the opinions of those who commented.

Agradecimientos

A mis padres y hermana, por su amor
y apoyo incondicionales.

A mis directores de tesis, por
su apoyo y guía durante el
desarrollo de este trabajo.

A mis mejores amigos MASH y MNBP,
gracias por estar siempre
que los necesité.

A mi grupo de trabajo MACM, MASH y MNBP,
por el intercambio de ideas y el apoyo
en nuestros trabajos.

Al CIC, por darme las herramientas y
conocimientos necesarios para
desarrollar mi investigación.

A CONACyT, por su apoyo
económico.

Índice general

Resumen	i
Abstract	ii
Agradecimientos	iii
Índice de tablas	viii
Índice de figuras	ix
1 Introducción	1
1.1 Antecedentes	2
1.2 Justificación	5
1.3 Hipótesis	5
1.4 Objetivos	6
1.4.1 Objetivo general	6
1.4.2 Objetivos específicos	6
1.5 Contribuciones	7
1.6 Estructura de la tesis	7
2 Estado del arte	9
2.1 Polaridad de sentimientos	9

2.1.1	Aprendizaje automático	9
2.1.2	Aprendizaje profundo	11
2.2	Árboles de retuits o de tuits citados	12
2.3	Agrupamiento y polaridad	13
2.4	Controversia	14
2.5	Corpus	16
2.5.1	COST	16
2.5.2	RepLab 2013	16
2.5.3	Corpus general del TASS	17
2.5.4	TAAS InterTAAS	17
3	Marco teórico	18
3.1	Inteligencia Artificial (IA)	18
3.1.1	Aprendizaje automático	19
3.1.2	Aprendizaje profundo	21
3.2	Procesamiento de lenguaje natural (PLN)	22
3.3	Modelos	24
3.3.1	Regresión logística (RL)	24
3.3.2	Naïve Bayes multinomial (NBM)	25
3.3.3	Máquinas de soporte vectorial (MSV)	27
3.3.4	Árboles de decisión	29
3.3.5	K-medias	30
3.3.6	Perceptrón	31
3.3.7	Perceptrón multicapa (MLP)	34
3.3.8	Propagación hacia atrás	35
3.3.9	Transformadores	36

4	Descripción general de nuestra propuesta	38
4.1	Modelo de polaridad	42
4.1.1	Selección del corpus	42
4.1.2	Manipulación de los datos	42
4.1.3	Preprocesamiento	45
4.1.4	Extracción de características	47
4.1.5	Modelos propuestos	48
4.1.6	Selección del mejor modelo	50
4.2	Desarrollo del corpus de controversia	51
4.2.1	Selección de las noticias	51
4.2.2	Encuesta	52
4.2.3	Proceso de etiquetado	52
4.2.4	Análisis del corpus	54
4.2.5	Selección de las etiquetas	55
4.2.6	Corpus de controversia	57
4.3	Rutina de obtención de hilos en Twitter	58
4.3.1	Descripción del funcionamiento	58
4.4	Preprocesamiento de los datos de polaridad	59
4.4.1	Descripción del preprocesamiento	60
4.5	Modelo de controversia	60
4.5.1	Modelos propuestos	60
5	Experimentos y resultados	62
5.1	Modelo de polaridad	62
5.1.1	Configuración de los experimentos	63
5.1.2	Resultados de la primera fase de experimentación	63

5.1.3	Resultados de la segunda fase de experimentación, pruebas de robustez	64
5.1.4	Comparación con el estado del arte	65
5.2	Rutina de obtención de hilos de Twitter	66
5.2.1	Agrupamiento y conteos de reacciones	66
5.2.2	Árboles de tuits citados	66
5.3	Modelo de controversia	68
5.3.1	Primera fase de experimentación	68
5.3.2	Segunda fase de experimentación	69
5.3.3	Discusión	72
6	Conclusiones y trabajo futuro	73
6.1	Conclusiones	73
6.2	Trabajo futuro	74
	Referencias	76

Índice de tablas

4.1	Distribución del corpus en los conjuntos de entrenamiento y prueba	44
4.2	Grado de acuerdo entre anotadores para las 4 clases	56
4.3	Grado de acuerdo entre anotadores para la primera redistribución de clases	56
4.4	Grado de acuerdo entre anotadores para la segunda redistribución de clases	57
5.1	Resultados para <i>LogisticRegression</i>	63
5.2	Resultados para <i>MultinomialNB</i>	63
5.3	Resultados para <i>SVC</i>	64
5.4	Resultados para BETO	64
5.5	Resultados para los modelos de aprendizaje automático tras la prueba de robustez, mín. TF = 1 y 2	65
5.6	Resultados para los modelos de aprendizaje automático tras la prueba de robustez, mín. TF= 3 y 5	65
5.7	Comparación con el estado del arte	65
5.8	Ejemplo del corpus de controversia completo	68
5.9	Resultados obtenidos en la primera etapa de experimentación, clasificación de controversia	69
5.10	Resultados para el perceptrón multicapa, clasificación de controversia	70
5.11	Resultados para el árbol de decisión, clasificación de controversia	71
5.12	Comparación de los modelos para clasificar controversia en noticias	72

Índice de figuras

3.1	Función logística estándar	25
3.2	Ejemplo de una SVM para dos clases	28
3.3	Ejemplo de un árbol de decisión	30
3.4	Neurona biológica	32
3.5	Neurona artificial (TLU)	32
3.6	El perceptrón	33
3.7	Perceptrón multicapa	35
3.8	Arquitectura del modelo transformador (Vaswani et al., 2017)	36
4.1	Ejemplo de la distribución de polaridad en el hilo de conversación de una noticia	40
4.2	Etapas para el clasificador de controversialidad	41
4.3	Distribución de las etiquetas de polaridad en el corpus general del TAAS	43
4.4	Distribución de las clases tras la manipulación del corpus general del TASS	44
4.5	Distribución de clases para el conjunto de entrenamiento	45
4.6	Distribución de clases para el conjunto de prueba	45
4.7	Distribución de participantes por grupo de edad	53
4.8	Distribución de participantes por grado máximo de estudios	53
4.9	Distribución de clases en el corpus de controversialidad tras la primera iteración de etiquetado	54

5.1	Conteos de respuestas por etiqueta de clase de los bloques de conversación analizados	67
5.2	Ejemplo de un árbol de tuits citados generado con la rutina de extracción de hilos de Twitter	67
5.3	Matriz de confusión para los resultados de la primera fase de experimentación, clasificación de controversia	69
5.4	Matriz de confusión para perceptrón multicapa, clasificación de controversia	71
5.5	Matriz de confusión para el árbol de decisión, clasificación de controversia	71

Capítulo 1

Introducción

El desarrollo de la Internet, y su rápida popularización, ha hecho que esta se convierta en una de nuestras principales herramientas para la consulta y diseminación de la información. Con el desarrollo y crecimiento exponencial de las redes sociales, la forma en la que los seres humanos nos relacionamos se ha visto afectada, ya que la forma en la que interactuamos por medio de estas se ha convertido en una tarea cotidiana. A tal grado que incluso podemos mantener relaciones interpersonales únicamente en línea con el uso de plataformas como Facebook, WhatsApp o Twitter. Además, la reciente pandemia de COVID-19 provocó una acelerada evolución de este fenómeno, ya que nos vimos forzados a interactuar utilizando plataformas en línea exclusivamente. Tal ha sido la popularización del empleo de las redes sociales, que de acuerdo con la publicación “*Digital 2021*”¹ aproximadamente 57% de la población mundial usa de forma activa las redes sociales como Twitter, y en promedio, invertimos alrededor de 2 h 27 min diariamente en esta actividad. Las plataformas en línea se han visto rápidamente involucradas en el discurso público, sus algoritmos ayudan a los ciudadanos a unirse a grupos sociales, a clasificar el ruido del discurso público e incluso mantenerse al tanto de los eventos actuales (Bastick, 2021).

¹<https://datareportal.com/reports/digital-2021-october-global-statshot>

La facilidad de acceso a las redes sociales permite a los usuarios compartir sus publicaciones favoritas con cientos de sus contactos en un abrir y cerrar de ojos. En general, las redes sociales son una gran herramienta tanto para publicar como para opinar en los comentarios de una publicación acerca de la misma, y Twitter es un gran caso de estudio. De acuerdo con Kwak et al. (2010), al clasificar los temas en tendencia en Twitter, mostró que la gran mayoría de los temas tratados en los tuits (alrededor del 85 %) son titulares o noticias persistentes.

Las publicaciones en Twitter pueden ser sobre cualquier tema y además, existen pocas restricciones sobre el contenido de las publicaciones (p. ej. noticias, comentarios, etc.). Esta es una de las razones principales por la que los internautas deciden utilizar esta plataforma sobre las demás, ya que perciben libertad para expresar sus opiniones. Al analizar comentarios de Twitter, Li et al. (2019) descubrieron que el contenido de los comentarios usualmente se encuentra cargado con las emociones de quien los publica. Esta carga emocional es útil para la identificación de los puntos de vista de los usuarios (positivo, negativo o neutral), en otras palabras, podemos identificar la polaridad del texto publicado.

Las redes sociales nos dan la oportunidad de entender como es que los lectores reaccionan ante una variedad de temas, desde política hasta entretenimiento. Algunos de estos temas pueden controvertidos, si las personas debaten sobre el tema por un periodo de tiempo. Twitter es entonces una gran herramienta para analizar las respuestas a publicaciones sobre noticias en las que se puede identificar el debate en los comentarios publicados por los lectores al respecto.

1.1 Antecedentes

En general la tarea de clasificación se puede afrontar desde el paradigma supervisado y el no supervisado, algunos de los algoritmos más conocidos para realizar esta tarea son los clasificadores Bayesianos (Wang et al., 2020), las máquinas de

soporte vectorial (SVM) (Cortes & Vapnik, 1995), entre otros. Existen algoritmos basados en modelos matemáticos de las neuronas humanas conocidos como clasificadores neuronales (McCulloch & Pitts, 1943). Los clasificadores neuronales incluyen al perceptrón (Rosenblatt, 1958), estos modelos pertenecen al aprendizaje profundo.

El aprendizaje profundo comenzó con el primer modelo matemático de una neurona propuesto por McCulloch y Pitts (1943). Rosenblatt (1958) inventa el perceptrón, pero se presentó un periodo de inactividad en la investigación sobre la inteligencia artificial debido a la publicación de Marvin y Seymour (1969) en la que atacaron a la idea del perceptrón de Rosenblatt. Pero no fue hasta la década de los 80 que la inteligencia artificial tuviera su resurgimiento como campo de estudio con la publicación de Rumelhart et al. (1986) en la que se publica el algoritmo de propagación hacia atrás o *back-propagation* (BP).

El análisis de opinión es un campo de investigación que se encuentra estrechamente relacionado con la lingüística computacional y el procesamiento de lenguaje natural, su principal tarea es el determinar la opinión que una persona da sobre cierto tema o entidad al analizar el texto dónde se plasman los comentarios. Algunas veces el análisis de opinión se menciona como: análisis de subjetividad, minería de opiniones o extracción de valoraciones (Mejova, 2009). El análisis automático de opiniones forma parte del análisis de opinión, a su vez la polaridad de opinión es una subtarea de este campo (Gambino, 2019). La polaridad de opinión es una característica peculiar de los textos, dado un texto que refleje opinión se requiere clasificar entonces cómo esta opinión se encuentra polarizada, es decir, que tan positiva o negativa es. Distinguir si un texto es positivo o negativo es relativamente sencillo para los humanos, pero para las computadoras es una tarea bastante compleja (Pang et al., 2002). Uno de los enfoques propuestos para determinar la polaridad de los textos ha sido considerar la tarea como un problema de clasificación, usualmente las clases son la polaridad misma de las opiniones (p. ej. negativa o positiva). Existen dos metodologías principales para determinar la polaridad de un texto. En primer lugar, el aprendizaje automático, que utiliza un corpus o conjunto de datos para entrenar un clasificador, esta metodología forma parte del aprendizaje

supervisado. Por otro lado, la aproximación basada en la orientación semántica de las palabras, esta metodología no requiere de un entrenamiento previo, por lo tanto, es considerada como aprendizaje no supervisado.

El aprendizaje supervisado emplea algoritmos de aprendizaje automático o en inglés *Machine Learning* (ML) para realizar la clasificación, se logra al entrenar el algoritmo con ejemplos etiquetados, y luego probar su desempeño con datos no etiquetados que el algoritmo no haya visto. Uno de los primeros esfuerzos en efectuar la clasificación de polaridad en textos aplicando el enfoque supervisado fue en (Pang et al., 2002). Donde los autores trabajaron con un corpus de reseñas de películas en inglés para obtener la polaridad general de la reseña, durante la tarea de clasificación se utilizaron los algoritmos Naïve Bayes (NB), entropía máxima (ME, por sus siglas) y máquinas de soporte vectorial (SVM, por sus siglas) obteniendo una exactitud del 82 %. Para textos en español, uno de los primeros esfuerzos para hacer la clasificación de polaridad fue en (Cámara et al., 2011) dónde se usó un corpus en español de críticas de cine, para realizar la tarea de clasificación se emplearon los algoritmos Naïve Bayes y máquina de soporte vectorial, obteniendo una precisión de alrededor del 87 %. Más recientemente, se tienen los trabajos de (Vizcarra et al., 2018) y (Ochoa Luna & Ari, 2018).

El enfoque no supervisado intenta determinar la polaridad de los datos, sin la necesidad de tener conocimiento previo de los mismos. Uno de los primeros intentos fue el efectuado en (Turney, 2002) quien utilizó un corpus con reseñas de automóviles, películas, bancos y destinos turísticos, utilizó etiquetado gramatical y el algoritmo PMI-IR para obtener la orientación semántica de las frases usando la orientación media para clasificar, obteniendo un 74 % de exactitud. Para el aprendizaje no supervisado se han creado diccionarios léxicos que contienen la polaridad de las palabras, los algoritmos no supervisados emplean la información de la polaridad de las palabras del texto analizado que coincidan con las contenidas en el diccionario léxico para calcular la polaridad global del texto. Otros esfuerzos han sido combinar el uso de los diccionarios léxicos con el paradigma de aprendizaje supervisado como en (Martin-Valdivia et al., 2013).

La determinación de la controversia de un texto es una tarea poco usual dada la escasez de conjuntos de datos etiquetados, algunos trabajos que realizan la clasificación de la controversia son (Pennacchiotti & Popescu, 2010) y (Mejova et al., 2014).

1.2 Justificación

Los seres humanos dependemos cada vez más de las nuevas tecnologías como la internet. Esta dependencia nos ha hecho usar las redes sociales como fuente de consulta y diseminación de la información e intercambio de opiniones, con la inclusión de los algoritmos de aprendizaje automático y profundos, nos encontramos en una época en la que analizar información textual de forma masiva es cada vez más común. En ese sentido, el analizar la controversialidad de las noticias puede ser de gran ayuda para determinar el impacto que estas tienen en sus lectores.

La clasificación de la controversia de las noticias ha sido poco estudiada, por lo que indagar en este campo de estudio resulta útil, ya que, con la propuesta aquí descrita, se abren oportunidades para realizar investigación sobre la clasificación de controversialidad de noticias desde la perspectiva de los lectores. Para efectuar una clasificación confiable se debe desarrollar e implementar un modelo capaz de ayudarnos a visualizar como es que las noticias polarizan a sus lectores, y a su vez utilice esta información para determinar si las noticias son controversiales o no.

1.3 Hipótesis

La controversia de una noticia entre sus lectores indica cómo es que los ha impactado porque mientras exista controversia en los comentarios, el impacto que esta tendrá será negativo. Mientras más acalorada sea la discusión entre usuarios en los comentarios de una publicación mayor será el desacuerdo entre ellos. Por lo que se

puede usar un modelo que clasifique la controversia de las noticias a través de la polaridad de los comentarios de la misma.

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar un modelo computacional que clasifique noticias de acuerdo con su intención y estime de manera automática su impacto en aquellos que las consulten.

1.4.2 Objetivos específicos

- Revisar los trabajos relacionados con la clasificación de la polaridad de sentimientos en textos.
- Buscar y seleccionar un corpus de polaridad de sentimientos.
- Analizar y preprocesar el corpus.
- Desarrollar una rutina capaz de extraer la información del hilo de conversación de un tuit desde la API de Twitter.
- Evaluar y seleccionar los modelos de aprendizaje automático y basados en transformadores para la clasificación de polaridad basados en las métricas de desempeño.
- Implementar el mejor clasificador de polaridad en la rutina de extracción de información de tuits para obtener la polaridad de las publicaciones
- Crear un conjunto de datos de controversia con noticias en español en México.
- Desarrollar un modelo capaz de realizar la clasificación de controversia con los datos de polaridad de una conversación.
- Validar y evaluar el desempeño de los modelos de clasificación de controversia.

1.5 Contribuciones

Este trabajo aporta una nueva forma de clasificar la controversialidad de las noticias desde el punto de vista de sus lectores, las etapas aquí descritas y la forma en la que se unieron son una novedad para hacer esta clasificación. Además, se desarrolló una rutina capaz de extraer el hilo de conversación de un tuit, esta rutina se ve complementada con un clasificador de polaridad; con los datos obtenidos por la rutina se generan árboles de tuits citados, y la información de interés se almacena para su posterior utilización.

1.6 Estructura de la tesis

Una visión general del contenido de cada capítulo de esta investigación se provee a continuación.

El capítulo 2 habla sobre los trabajos relacionados con los temas de interés de esta investigación. Por lo tanto, este capítulo contiene referencias al estado del arte que describa la clasificación de polaridad de sentimientos con modelos de aprendizaje automático y profundo, los conjuntos de datos que han sido desarrollados para el análisis de la polaridad, la generación de árboles de tuits citados, la clasificación de controversialidad de textos, entre otros.

En el capítulo 3 se describe el marco teórico. En este capítulo se da una breve introducción a la inteligencia artificial, se cubre la teoría detrás de los modelos utilizados para el desarrollo de la solución y una visión general del procesamiento de lenguaje natural.

El capítulo 4 detalla la solución propuesta y cada una de sus etapas, el preprocesamiento realizado al corpus de polaridad seleccionado, y describe a detalle el proceso de desarrollo del corpus de controversia.

En el capítulo 5 se presentan los experimentos efectuados en cada una

de las etapas de la solución propuesta, la configuración para proceder con dichos experimentos y los resultados obtenidos tras la experimentación.

Finalmente, en el capítulo 6 se proveen las conclusiones de esta investigación y se establece el trabajo futuro.

Capítulo 2

Estado del arte

En este capítulo presentamos detalladamente los trabajos relacionados con nuestra propuesta.

2.1 Polaridad de sentimientos

2.1.1 Aprendizaje automático

Cámara et al. (2011), utilizaron un corpus en español de críticas de cine, con un total de 3,878 críticas. Estas críticas fueron escritas por usuarios de la web y se puntuaron en un rango de entre 1 y 5, siendo el 1 el valor asignado para la categoría muy mala y el 5 corresponde a la categoría muy buena, el 3 corresponde a neutral (ni mala, ni buena). Para el estudio no se consideró la puntuación de las críticas y las 5 categorías fueron reducidas a dos, positiva y negativa, dejando fuera la categoría neutral. Agruparon en la clase negativa las etiquetas 1 y 2; y en la clase positiva las etiquetas 4 y 5. Para la etapa de clasificación, los autores propusieron los siguientes algoritmos de clasificación: Máquina de soporte vectorial y Naïve Bayes. Usaron las técnicas *stopper* y *stemmer* como parte del preprocesamiento. Para la extracción

de características propusieron: TF-IDF, frecuencia de términos (TF), número de ocurrencias (TO) y ocurrencia binaria (BTO). Los mejores resultados los obtuvieron con la máquina de soporte vectorial, sin reducir las características y empleando la técnica *stopper* y *stemmer* con el método TF-IDF. Obtuvieron 87,71 % y 86,75 % de valor-F respectivamente.

En el artículo “*TASS: Detecting Sentiments in Spanish Tweets*” Saralegi y San Vicente (2012), utilizaron el conjunto de datos general del Taller de Análisis Semántico de la Sociedad Española de Procesamiento de Lenguaje Natural (TASS) que contiene alrededor de 7,000 tuits de entrenamiento clasificados en muy positivo (P+), positivo (P), neutral (NEU), negativo(N), muy negativo (N+) y sin sentimiento (NONE). Los autores crearon un lexicón para español usando dos recursos diferentes, un lexicón para el idioma inglés que fue traducido automáticamente al español con un diccionario bilingüe inglés-español, y las palabras del mismo corpus. Obtuvieron las palabras más representativas para cada etiqueta de polaridad al emplear la prueba de razón de verosimilitud (*Likelihood-ratio test*). Para realizar la clasificación, diseñaron un algoritmo de aprendizaje supervisado utilizando la configuración por defecto de la implementación de la máquina de soporte vectorial del software de minado de datos Weka o SMO. Todos los clasificadores fueron validados con la estrategia de validación cruzada *10-folds*. Los autores obtuvieron una exactitud de 71,1 % al evaluar el clasificador con el conjunto de prueba.

Posteriormente, Henriquez et al. (2017), emplearon el corpus de la competición SemEval 2016 para el español, los autores preprocesaron los textos del corpus aplicando lematización y etiquetado morfosintáctico, removieron las palabras auxiliares y la corrección ortográfica. Propusieron la extracción de aspectos del texto usando la ontología de dominio, que describe el vocabulario específico para un dominio (p. ej. bancos, hoteles, cine, etc.) en conjunto con un lexicón de similitud semántica, obteniendo una lista de aspectos. Para la etapa de clasificación, utilizaron máquinas de soporte vectorial (SVM) utilizando validación cruzada *10-folds*. La exactitud obtenida de los experimentos fue del 84 %.

Arias et al. (2022), crearon un conjunto de datos extrayendo tuits directamente de la API¹ de Twitter², para la clasificación de polaridad se extrajeron 8,000 tuits etiquetados como positivo, negativo o neutro. Para la tarea de clasificación, los autores emplearon los siguientes algoritmos: *Random Forest* (RF), *K-Nearest Neighbors* (KNN), Naïve Bayes (NB), *Gradient Boosting* (GB), *Stochastic Gradient Descent* (SDG), *Support Vector Classifier* (SVC) y *Extreme Gradient Boosting* (XGBoost). Del proceso de experimentación, el mejor algoritmo de clasificación fue el *Extreme Gradient Boosting*, que obtuvo una exactitud del 84 %.

2.1.2 Aprendizaje profundo

Ochoa Luna y Ari (2018), proponen un *pipeline* compuesto por *word embeddings*, una red neuronal convolucional y una red neuronal recurrente. La red convolucional y la red recurrente se encuentran conectadas en serie, y se utilizan los *embeddings* para vectorizar el texto y son estos vectores la entrada de la red convolucional. Utilizan tres *word embeddings* para codificar el texto, por lo que la entrada de la red convolucional es 3D, y se realizan convoluciones desde 1 hasta 5 palabras. La red recurrente recibe como entrada el vector de salida de la red convolucional, finalmente se tiene una capa densa o completamente conectada en la salida de la red que permite ejecutar la tarea de clasificación. Emplearon el corpus de InterTASS 2017, y tras ejecutar los experimentos correspondientes, el *pipeline* tuvo una exactitud del 60 %.

Para la competición de IBERLEF 2021, Abreu Salas et al. (2021-09), propusieron dos arquitecturas basadas en BETO que es un modelo BERT entrenado en un gran corpus en español. Las arquitecturas propuestas fueron un clasificador multi-clase que aprendió las 5 categorías de polaridad de forma simultánea, y una arquitectura de clasificadores en cascada, un ensamble de clasificadores binarios. Los mejores resultados fueron obtenidos con el ensamble de modelos obteniendo un resultado para la exactitud del 53 %.

¹<https://developer.twitter.com/>

²<https://twitter.com/>

En la competencia REST-MEX 2021 Vázquez et al. (2021), presentaron dos enfoques, un ajuste fino de BERT, y un ajuste fino de BERT en conjunto con vectores TF-IDF, su primer enfoque obtuvo una exactitud del 56 %, mientras que la combinación de BERT y los vectores TF-IDF obtuvieron una exactitud del 54 %. Para el segundo enfoque, BERT en conjunto con TF-IDF, obtuvieron dos vectores de características que concatenaron para hacer uno solo para entrenar un algoritmo de regresión logística que se utilizó para predecir la clase correspondiente al texto, utilizando como entrada los *embeddings* contextuales obtenidos con BERT.

Para REST-MEX 2022 Callejas-Hernández et al. (2022), utilizaron los modelos pre-entrenados BETO y RoBERTuito. Para el preprocesamiento emplearon el tokenizador de BERT para finalmente hacer un ajuste fino de ambos modelos con el corpus de entrenamiento. Para la tarea de análisis de sentimientos de la competición, el equipo obtuvo una exactitud del 75 %.

2.2 Árboles de retuits o de tuits citados

Los árboles de retuits han sido implementados con dos tareas principales en mente, estudiar la propagación de los tuits y sus retuits en Twitter y obtener los nodos más influyentes durante el proceso de propagación. En esta sección, los retuits y tuits citados serán entendidos como iguales, ya que con el desarrollo de la API V2 de Twitter, los retuits pasaron a ser una publicación en la cuenta de quien retuiteó sin información adicional. Los tuits citados suplieron a los retuits en los que los usuarios podían dar su opinión, en la nueva versión de Twitter, los tuits citados pueden contener la opinión de quién citó el tuit original, e incluso en el tuit citado se puede desarrollar una nueva conversación.

Kwak et al. (2010), utilizaron los árboles para determinar la velocidad a la que un tuit viaja en la red, y la profundidad que este alcanza. Desarrollaron un árbol de diseminación de información, al experimentar con este algoritmo descubrieron

que la mayoría de árboles de retuits alcanzan una profundidad máxima de primer o segundo nivel. Además, ningún árbol tenía una profundidad mayor a seis niveles de retuits. Los niveles de profundidad se refieren a los retuits, el primer nivel es el tuit original y sus comentarios, el segundo nivel corresponde a los retuits del tuit original, el tercer nivel corresponde a los retuits de los retuits y así sucesivamente. Su análisis mostró como es que un tuit puede propagarse a través de Twitter y cuántas publicaciones se ven involucradas en el proceso.

Bhowmick et al. (2018), construyeron árboles de “influencia” con secuencias temporales de retuits, estas secuencias mostraron un alto nivel de confianza al modelar de forma precisa el árbol de una cascada de retuits. CasCon, que es el modelo de aprendizaje no supervisado propuesto por los autores para construir árboles, es capaz de identificar los nodos de mayor influencia en un vecindario de “amistades”. Con los árboles de “influencia” propusieron dos metodologías: (a) descubrir la relación entre los intervalos de retuits y la formación de árboles representando cuan influyente es el nodo de mayor influencia y (b) proveer la información básica verdadera para la validación de los árboles construidos con el modelo CasCon.

Roth et al. (2022), diseñaron árboles de tuits citados para un tuit raíz, considerando cascadas recursivas de tuits. Los experimentos mostraron que el tamaño del árbol sigue una ley heterogénea donde el 75 % de todos los árboles generados tienen 2 o tres niveles a lo más, y al calcular la profundidad promedio de los árboles, los resultados mostraron que el 90 % de los árboles tienen una profundidad media de 1, indicando la ausencia de citas secundarias y citas de las citas.

2.3 Agrupamiento y polaridad

El análisis de sentimientos ha utilizado técnicas de agrupamiento en conjunto con el clasificador principal para mejorar los resultados del modelo de clasificación, o para poder observar el comportamiento o tendencia de los tuits.

Coletta et al. (2014), propusieron emplear una máquina de soporte vectorial para realizar la tarea de clasificación de polaridad en los tuits, posteriormente utilizaron un conjunto de *clusters* para afinar los resultados de la máquina de soporte vectorial e incrementar su calidad. Experimentaron con cuatro conjuntos de datos, de esta etapa de experimentación se observó que la combinación de la máquina de soporte vectorial y las técnicas de agrupamiento mejoran los resultados de la tarea de clasificación de polaridad en tuits.

Ahuja y Dubey (2017), utilizaron el lexicón AFINN en conjunto con TextBlob que es una librería de procesamiento de lenguaje natural para Python. Para crear dos grafos de datos dispersos, los autores unieron las puntuaciones de subjetividad y la polaridad obtenidas con TextBlob y la polaridad obtenida con el lexicón AFINN, finalmente los grafos se agruparon empleando técnicas de agrupamiento para determinar la clase a la que la polaridad de cada uno de los tuits pertenece.

Shamrat et al. (2021), hicieron uso de la librería TextBlob para calcular la polaridad y subjetividad de los tuits, ya que la puntuación de polaridad se había determinado, el algoritmo *K-Nearest Neighbor* (KNN) de aprendizaje supervisado fue utilizado para realizar la clasificación final de polaridad de los tuits, los datos de entrada del algoritmo fueron las puntuaciones de polaridad calculadas con TextBlob. La salida del algoritmo fue la clase de polaridad agrupada.

2.4 Controversia

Pennacchiotti y Popescu (2010) proponen un método para calcular la controversia de los tuits publicados por una celebridad (entidad) en un periodo de tiempo, el método hace uso de dos lexicones, uno de 7,590 términos sentimentales y uno de 750 términos controversiales. Para calcular la controversia de una publicación determinan la puntuación histórica de controversia que determina que tan controversial ha sido una entidad en la web. También calculan la puntuación temporal de con-

troversia que estima la controversia de la entidad analizada durante un periodo de tiempo. Además, calculan una puntuación general de controversia que es una combinación lineal de los anteriores. De sus experimentos, los autores obtuvieron una precisión promedio del 66 %.

Mejova et al. (2014) detectan la controversia en noticias en línea. Para lograr su objetivo, los autores desarrollaron un conjunto de datos de palabras, este conjunto de datos se compone de 2,000 términos de alta frecuencia y para etiquetarlos se les pidió a 7 anotadores que indicaran si la palabra que se les presentaba era fuertemente controversial, algo controversial, poco controversial o no controversial. Para la parte experimental, recopilaron todos los artículos que tuvieran el término controversial analizado, luego los artículos seleccionados se juntaron para formar un “súper”-artículo. El análisis de este súper-artículo se realizó con lexicones de sentimientos. Con la información de la polaridad proveída por los lexicones, obtuvieron la distribución de las palabras y su polaridad que más se relacionan con los términos controversiales. Finalmente, los autores concluyeron que normalmente los temas controversiales se enmarcan utilizando emociones negativas. Es decir, cuando un autor enfatiza en los aspectos negativos de la noticia, esta se torna controversial.

Para hacer la clasificación de controversia, Hessel y Lee (2019) crearon un conjunto de datos con discusiones extraídas de reddit³. A las publicaciones del conjunto de datos se les asignaron etiquetas de clase binarias: controversial y no controversial. El criterio que tomaron para asignar estas etiquetas de clase fue el siguiente: se descartaron las publicaciones que tuvieron pocos votos, acomodaron las publicaciones por su porcentaje de votos, para las publicaciones que con más del 50 % de votos positivos y que además estuvieran en el cuartil más alto se les asignó la etiqueta no controversial. A las publicaciones con un gran porcentaje de votos negativos (parecido al total de votos positivos) se les asignó la etiqueta controversial. Propusieron una gran cantidad de clasificadores binarios como una LSTM con una capa de salida de regresión logística, BERT, combinaciones de LSTM y BERT por mencionar algunos. Tras la fase experimental, el clasificador con mejor desempeño

³<https://www.reddit.com/>

fue BERT-MP-512, que obtuvo una exactitud promedio del 68 %.

2.5 Corpus

Navas-Loro y Rodriguez-Doncel (2020) realizaron una recopilación de los corpus en español para análisis de sentimientos disponibles hasta la fecha de publicación de este artículo. Los corpus que contienen tuits etiquetados con la polaridad del texto, son los siguientes.

2.5.1 COST

Ante la necesidad de un corpus de polaridad en sitios de *micro-blogging*, Martínez-Camara et al. (2015) crearon su propio corpus COST (*Corpus of Spanish Tweets*). Este corpus contiene un total de 34,634 tuits, 17,317 positivos y 17,317 negativos. Ya que el lenguaje usado en Twitter tiene atributos únicos, los tuits fueron preprocesados por los autores removiendo las siguientes características: retuits, menciones, hipervínculos y *hashtags*. Además, aplicaron los siguientes filtros al texto: remover líneas nuevas (los símbolos, para tener el texto completo en una sola línea), remoción de emoticonos que no proveyeran información útil al tuit, normalización de caracteres repetidos y normalización de la forma en la que se expresa la risa. El corpus contiene información sobre el usuario quién publicó, el texto y el valor de polaridad que se asignó directamente de la consulta realizada a la API de Twitter.

2.5.2 RepLab 2013

Amigo et al. (2013) desarrollaron un corpus con tuits en inglés y español con la siguiente distribución de etiquetas de clase: 63,442 positivos, 30,493 neutrales y 16,415 negativos. De estos tuits, únicamente 28,983 están escritos en español. Algunos ejemplos de la información contenida en el corpus son el *tweet id*, el tópico y

la entidad que lo publicó, el texto no forma parte del corpus, ya que se espera que quien utilice el corpus obtenga la información faltante sea obtenida directamente de la API correspondiente.

2.5.3 Corpus general del TASS

El corpus general del TASS⁴ se dio a conocer en la primera edición de la competición organizada en el TAAS en 2012 (Villena-Roman et al., 2013) y fue utilizado desde 2012 hasta 2017. Contiene 68,017 tuits obtenidos de celebridades de diferentes países y abarca un total de 10 temas (política, entretenimiento, economía, música, fútbol, películas, tecnología, deportes, literatura, entre otros). Este corpus contiene el *tweet id*, el identificador del usuario, la fecha de publicación, el tema de cada tuit y la etiqueta de polaridad asignada a cada tuit (P+, P, NEU, N, N+) y una etiqueta extra NONE para aquellos tuits que no reflejan sentimiento alguno.

2.5.4 TAAS InterTAAS

Este corpus fue elaborado para la edición de 2017 del taller TAAS (Camara et al., 2016), el corpus InterTAAS (*International TASS Corpus*) contiene 3,413 documentos con su respectiva etiqueta de polaridad (P, NEU, N, NONE). La distribución de etiquetas en el conjunto de datos es la siguiente, un 30-33% son tuits positivos, 40-43% de los tuits son negativos, entre 11-13% son tuits neutrales y el resto son tuits que no presentan sentimiento alguno.

⁴<http://tass.sepln.org/>

Capítulo 3

Marco teórico

En el presente trabajo, se implementó un *pipeline* de trabajo para clasificar la polaridad de las noticias desde la perspectiva del lector. El pipeline se compone por un vectorizador TF-IDF, algoritmo de clasificación de polaridad, un clasificador *k-means* y un perceptrón multicapa a la salida del *pipeline* utilizado para dar la clasificación de controversia de la noticia desde la perspectiva de los lectores. En esta sección se da un breve acercamiento a los fundamentos matemáticos de los modelos considerados para el desarrollo de la solución.

3.1 Inteligencia Artificial (IA)

Hablar sobre inteligencia artificial es remontarnos a la mitología griega en la que aparecen los autómatas, máquinas creadas por los dioses para realizar tareas que ellos encontraban aburridas. Uno de los primeros intentos por desarrollar inteligencias artificiales se da en 1843 con Lady Lovelace y Charles Babbage quienes trabajaron para poder hacer realidad la construcción del motor analítico de Babbage (Crank, 1963). Pero no fue sino hasta el verano de 1956, durante la conferencia de Dartmouth que el término “Inteligencia artificial” fue utilizado oficialmente (McCorduck et al., 1977).

A lo largo de la historia de la inteligencia artificial muchas han sido las definiciones dadas por diversos autores, pero no ha existido un común acuerdo en dar una definición oficial sobre la IA. En el presente trabajo, adoptaremos nuestra propia definición de inteligencia artificial. *La aplicación de las técnicas computacionales que hagan posible la interpretación de los datos de forma similar a como lo hacemos los seres humanos.*

3.1.1 Aprendizaje automático

El aprendizaje automático es un área relacionada tanto con la cibernética como con las ciencias de la computación. Sus orígenes se asocian con Rosenblatt y la creación del perceptrón (Rosenblatt, 1958) que se convirtió en el prototipo de las redes neuronales artificiales o *Artificial Neural Networks* (ANN) modernas. El encargado de acuñar el término “*Machine Learning*” fue Samuel (1959). El aprendizaje automático tiene como objetivo el dotar a las computadoras con la capacidad de aprender mediante el desarrollo de algoritmos capaces de reconocer patrones y generalizar comportamientos haciendo uso de ejemplos. El aprendizaje automático se divide comúnmente en tres paradigmas de aprendizaje, el aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo. Estos paradigmas difieren en la forma en la que los datos son presentados a los algoritmos.

Aprendizaje supervisado

El aprendizaje supervisado es el paradigma de aprendizaje más común en el aprendizaje automático. Como su nombre lo indica, necesita la supervisión del ser humano para lograr el aprendizaje. Se define por el uso de pares de datos entrada-salida o ejemplos etiquetados. Estos pares de datos normalmente forman parte de conjuntos de datos más grandes que sirven para entrenar los algoritmos de clasificación de forma precisa. El proceso de entrenamiento sigue dos sencillos pasos, el algoritmo es alimentado con los datos de entrada y los datos de salida correctos que permiten al

modelo aprender en el tiempo; con estos datos de entrada, el algoritmo realiza predicciones y los compara con los datos de salida, el algoritmo evalúa su exactitud con la función de pérdida y ajusta sus parámetros hasta que el error se haya minimizado lo suficiente.

El aprendizaje supervisado puede ser separado en dos tipos de problemas, la clasificación y la regresión.¹

- Clasificación: utiliza un algoritmo para asignar de forma precisa los datos de prueba a sus categorías. Algunos algoritmos de clasificación empleados comúnmente son las máquinas de soporte vectorial, árboles de decisión, K-vecinos cercanos (KNN, por sus siglas) y bosques aleatorios.
- Regresión: es usada para comprender la relación entre variables dependientes e independientes. La regresión lineal y logística son algunos ejemplos de algoritmos usuales.

Aprendizaje no supervisado

Este paradigma de aprendizaje utiliza algoritmos de aprendizaje automático para analizar y agrupar los datos sin la necesidad de la supervisión del ser humano². Estos algoritmos son capaces de descubrir patrones en los datos de entrada que se les proporcionan sin la necesidad de la intervención del ser humano, es decir que estos algoritmos son capaces de generar conocimiento sobre los datos que se le presentan, aunque estos no se encuentren etiquetados, simplemente al calcular la similitud que tienen los datos, y agruparlos utilizando esta similitud, o descubriendo patrones ocultos dada la distribución misma de los datos. Los métodos más usuales son:

- Agrupamiento: es una técnica que agrupa datos no etiquetados, con base en el cálculo de sus similitudes. Pueden ser exclusivos como en el algoritmo *K-means* que estipula que un dato solo puede existir en un grupo a la vez; probabilísticos

¹<https://www.ibm.com/cloud/learn/supervised-learning>

²<https://www.ibm.com/cloud/learn/unsupervised-learning>

como en los modelos de mezclas gaussianas, entre otros.

- Reglas de asociación: Es un método basado en reglas, que encuentra las relaciones entre las variables de un conjunto de datos.
- *Autoencoders*: Comprimen los datos al codificarlos, y los reconstruye, creando una nueva representación de los mismos en el proceso de decodificación.

Aprendizaje por refuerzo

Este paradigma de aprendizaje se enfoca en dar a agentes inteligentes las herramientas para que sean capaces de aprender y tomar las mejores acciones dentro del entorno en el que se encuentren. Los algoritmos de aprendizaje por refuerzo, penalizan al agente cuando comente errores, y lo premia cuando acierta, así el agente puede ajustar sus parámetros, para buscar la mayor recompensa posible por medio de la prueba y error.

3.1.2 Aprendizaje profundo

El aprendizaje profundo es una rama del aprendizaje automático, que a su vez forma parte de la inteligencia artificial. El aprendizaje profundo se distingue del aprendizaje automático clásico por el tipo de datos con los que trabaja y sus métodos de aprendizaje.³ Los algoritmos de aprendizaje automático generalmente requieren de una etapa de preprocesamiento de los datos para que se organicen en un formato estructurado. El aprendizaje profundo, en cambio, elimina pasos del preprocesamiento de datos porque estos algoritmos pueden ser alimentados con datos no estructurados para posteriormente procesarlos, estos algoritmos automatizan la extracción de características. Además, con los procesos conocidos como descenso del gradiente y propagación hacia atrás, los algoritmos de aprendizaje profundo son capaces de ajustar sus parámetros de forma automática buscando incrementar su exactitud. Esta automatización de procesos, permite a estos algoritmos realizar predicciones

³<https://www.ibm.com/cloud/learn/deep-learning>

de gran precisión sobre datos que no habían visto. El aprendizaje profundo intenta mimetizar el funcionamiento del cerebro humano.

3.2 Procesamiento de lenguaje natural (PLN)

Un lenguaje es un conjunto finito de sentencias construido con un conjunto finito de alfabetos. Como los conjuntos de sentencias y alfabetos son finitos, el lenguaje es finito. Pero, aunque nuestro alfabeto es finito, la cantidad de sentencias que podemos generar a partir de este es tan grande que normalmente el lenguaje se considera infinito. Para el estudio del lenguaje con las computadoras entonces debemos limitar el lenguaje, ya que las computadoras y sus recursos (hardware) tienen una capacidad finita.

El procesamiento de lenguaje natural (PLN) es una rama de la lingüística, las ciencias de la computación y la inteligencia artificial que se enfoca en la interacción entre el lenguaje humano y las computadoras. “El procesamiento del lenguaje natural (PLN) es la rama de la informática, y más concretamente, la rama de la inteligencia artificial o IA, que se ocupa de dotar a los ordenadores de la capacidad de entender textos y palabras habladas del mismo modo que los seres humanos.”-IBM.⁴

El PLN resulta ser esencial para analizar texto, y extraer sus características. Para realizar un correcto análisis, hace uso de la combinación de la lingüística computacional, modelos estadísticos, aprendizaje automático y profundo, con el objetivo de darle a la computadora la habilidad de “comprender” un texto (p. ej. el significado, la intención, el sentimiento, entre otros).

Dependiendo del tipo de tarea que se vaya a realizar, y la complejidad de los textos analizados, el proceso del PLN tendrá variaciones para que los resultados se ajusten lo mejor posible a lo esperado. En general, todos los procesos comienzan con la recopilación, manipulación y preparación de los datos. El siguiente paso es el

⁴<https://www.ibm.com/cloud/learn/natural-language-processing>

preprocesamiento de los datos obtenidos.

Algunas de las técnicas más utilizadas en el preprocesamiento de texto son:

- Tokenización. Divide un texto en entidades más pequeñas conocidas como tokens, que pueden ser palabras, caracteres, sub palabras e incluso, si la tarea así lo requiere, signos de puntuación.
- Lematización. Consiste en extraer la forma básica o lema de una palabra flexionada. El lema es el representante de todas las formas flexionadas, es de gran utilidad porque reduce todas las variantes morfológicas de una palabra a un lema.
- Etiquetado gramatical (POS *tagging*). Este proceso asigna a cada una de las palabras de un texto su etiqueta de categoría gramatical basado en su definición y contexto.

Los pasos arriba mencionados son los que se utilizan más frecuentemente durante la etapa de preprocesamiento de un texto en el PLN, pero estos pueden variar dependiendo de la tarea a resolver o de las características que se quieran extraer del texto.

Ya que se preprocesó el texto (limpiamos todo lo que desde nuestra perspectiva no es relevante para la tarea de clasificación, puesto que no nutre a las características), es necesario darle una representación que la computadora sea capaz de comprender, es decir, debemos encontrar la forma de representar el texto numéricamente. La forma más sencilla para realizar esta representación es la bolsa de palabras o *Bag of Words* (BoW), que representa el vocabulario del corpus que vamos a emplear para entrenar el modelo, la bolsa de palabras se utiliza para crear un vector de frecuencias de aparición de términos para cada uno de los textos del corpus analizado. Finalmente, con la bolsa de palabras y los textos se produce una matriz en la que las columnas son cada una de las palabras del vocabulario del corpus y las filas son cada uno de los textos analizados.

Algunas de las tareas del PLN son: Análisis de sentimientos, traducción

automática, clasificación de textos, extracción de información, resumen automático de textos, contestación automática de preguntas (Chowdhary, 2020).

3.3 Modelos

3.3.1 Regresión logística (RL)

La regresión logística es un modelo estadístico que se utiliza para la clasificación y el análisis predictivo. Este modelo estima la probabilidad de ocurrencia de un evento basado en un conjunto de datos de variables independientes. En la literatura, también se le conoce como clasificación de entropía máxima (MaxEnt) o *logit regression*. En este modelo, la probabilidad que describe cada una de las posibles salidas se modela utilizando la función logística que se describe en la siguiente ecuación.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.3.1)$$

Esta función mapea y como una función sigmoidea de x que se aprecia mejor al representarla gráficamente como se observa en la figura 3.1.

La regresión logística ha sido utilizada para clasificar en una gran cantidad de aplicaciones del procesamiento de lenguaje natural como se menciona en (Berger et al., 1996). La representación del modelo es la siguiente.

$$Pw(y|x) = \frac{\exp(w^T f(x, y))}{\sum_{y'} \exp(w^T f(x, y'))} \quad (3.3.2)$$

donde x representa el contexto (tuit), y denota la etiqueta de clase del contexto x , $w \in \mathbb{R}^n$ es el vector de pesos; la función vector $f(x, y) \in \mathbb{R}^n$ representa las características extraídas del contexto x y la etiqueta de clase y . La función de normalización es una función característica/clase para $f(x, y')$ y queda definida como

se muestra en la siguiente ecuación.

$$f(x, y') = \begin{cases} 1 & \text{si } n(x) > 0 \text{ y } y' = y \\ 0 & \text{de lo contrario} \end{cases} \quad (3.3.3)$$

Los vectores de pesos deciden la importancia de las características durante la clasificación. Un peso mayor indica que la característica es más representativa para la clase y vice versa.

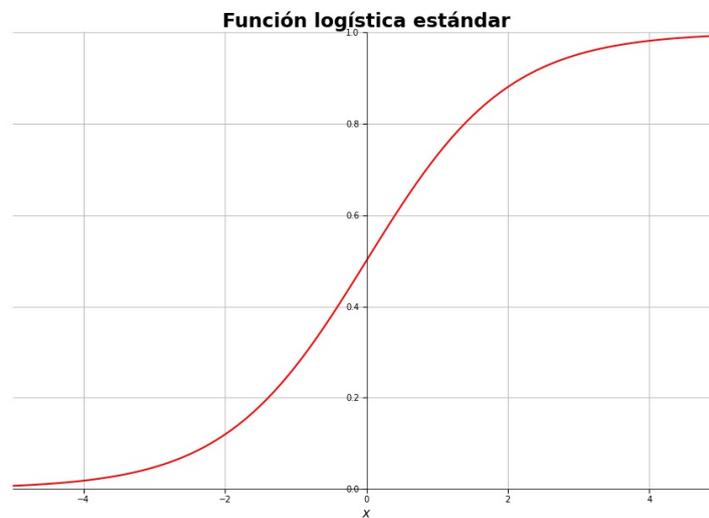


Figura 3.1: Función logística estándar

3.3.2 Naïve Bayes multinomial (NBM)

Los métodos Bayesianos son un conjunto de algoritmos de aprendizaje supervisado basados en la aplicación del teorema de Bayes 3.3.4 con la suposición Naïve de independencia condicional entre cada par de características dado el valor de la clase.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \quad (3.3.4)$$

donde $P(A|B)$ es la probabilidad de ocurrencia de A dado que B ha ocurrido (pro-

babilidad a posteriori); $P(A)$ es la probabilidad de que A ocurra (probabilidad a priori); $P(B)$ es la probabilidad de que B ocurra (probabilidad marginal) y $P(B|A)$ es la probabilidad de que B sea verdadero dado que A es verdadero.

Naïve Bayes multinomial (NBM) es un método probabilístico adecuado para realizar la tarea de clasificación con características discretas, hace uso del algoritmo Naïve Bayes (NB) para predecir la probabilidad de diferentes clases tomando en cuenta una gran cantidad de atributos, e indica la probabilidad de ocurrencia de un evento. NBM se implementa típicamente en la clasificación de texto, donde los datos se representan con vectores de conteo de palabras, ya que la probabilidad multinomial requiere de conteos enteros; pero se ha observado que en la práctica, conteos fraccionales como los generados con TF-IDF también funcionan. La probabilidad de que un documento d sea clasificado en c se calcula cómo:

$$P(c|d) = P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \quad (3.3.5)$$

donde $P(t_k|c)$ es la probabilidad condicional de que el término t_k ocurra en el documento (tuit) de clase c . $P(t_k|c)$ se interpreta como una medida de qué tanto la evidencia de t_k contribuye a que c pertenezca a la clase correcta. $P(c)$ es la probabilidad a priori de que un documento (tuit) ocurra en la clase c . Si los términos del documento no proveen evidencia clara para diferenciar una clase respecto de otra, la clase con mayor probabilidad es elegida. $\langle t_1, t_2, \dots, t_{n_d} \rangle$ son los tokens en d que forman parte del vocabulario utilizado para la clasificación, y n_d es la cantidad de esos tokens presentes en d . Si queremos conocer la mejor clase para el documento, la clase con la probabilidad a posteriori más alta o máxima es mapeada por la función c_{map} que define la polaridad del tuit y está dada por:

$$c_{map} = \arg \max_{c \in C} P(c|d) \quad (3.3.6)$$

3.3.3 Máquinas de soporte vectorial (MSV)

Las máquinas de soporte vectorial son algoritmos cuya tarea es encontrar un hiperplano de separación que defina la frontera de decisión entre las clases que se están separando. Por medio de este hiperplano, la SVM decide a que clase pertenecen los datos de entrada, y al ajustar el plano es capaz de definir la clase para datos nuevos mientras estén relacionados con los datos que se usaron para ajustar la SVM.

Una descripción más detallada de su funcionamiento es la siguiente. La SVM recibe datos de entrada, entonces emplea este conocimiento para encontrar los puntos más cercanos al hiperplano de separación entre clases. Estos puntos se conocen como vectores de soporte, con esta información, la SVM genera un hiperplano de separación; posteriormente se calcula la distancia entre el hiperplano y los vectores soporte, a esta distancia se le conoce como margen. La meta es maximizar el margen para que el hiperplano de separación se encuentre a una distancia óptima entre los puntos de las diferentes clases. Cuando los datos no pueden ser separados linealmente, se realiza una transformación a los datos a un nuevo espacio, de tal manera que se pueda hallar el hiperplano. La función utilizada para efectuar la transformación es conocida como kernel, y permite operar en espacios de alta dimensión.

La figura 3.2 muestra el hiperplano óptimo para un problema de clasificación binario. Como se puede observar, el margen entre los vectores soporte es máximo, y se puede decir que la frontera de decisión (hiperplano) se encuentra en su posición óptima. Las dimensiones del hiperplano dependerán de las dimensiones del vector de características que tiene como datos de entrada la máquina de soporte vectorial. Si tenemos un vector de dos características, entonces la SVM estará operando en un espacio 2-dimensional, por lo que el hiperplano de separación será una línea recta, si tenemos 3 dimensiones entonces el hiperplano de separación será un plano y así sucesivamente para espacios de mayor dimensión.

Matemáticamente, dados los vectores de entrenamiento $x_i \in \mathbb{R}^p$ y un vector, $y \in \{1, -1\}^n$ el objetivo es encontrar $w \in \mathbb{R}^p$ y $b \in \mathbb{R}$ tal que la predicción

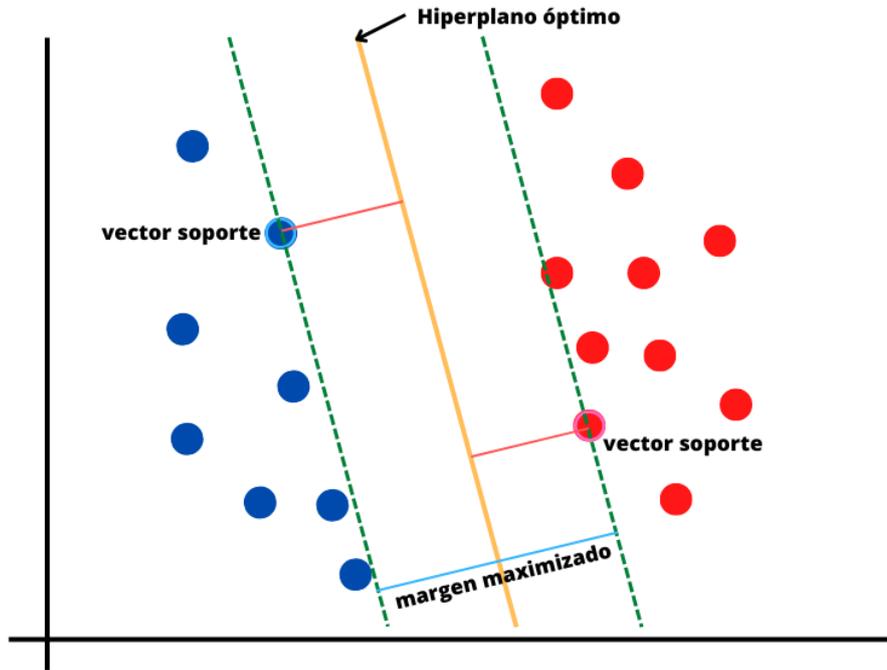


Figura 3.2: Ejemplo de una SVM para dos clases

dada por $\text{signo } w^T \phi(x) + b$ sea correcta para la mayoría de muestras. El problema a resolver es el siguiente.

$$\begin{aligned} \min_{w,b,\zeta} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^n \zeta_i \\ \text{sujeto a:} \quad & \\ & y_i (w^T \phi(x_i) + b) \geq 1 - \zeta_i, \\ & \zeta_i \geq 0, \quad i = 1, \dots, n \end{aligned} \tag{3.3.7}$$

La meta es maximizar el margen penalizando la SVM cuando una muestra no es correctamente clasificada o se encuentra entre los límites del margen. Ya que los problemas no siempre son perfectos, a algunas muestras se les permite estar a una distancia ζ_i del límite de su margen correspondiente. El término C define la penalización y controla la magnitud de la misma. Una vez que se ha resuelto el problema de optimización, la salida de la función de decisión para una muestra x es:

$$\sum_{i \in VS} y_i \alpha_i K(x_i, x) + b \quad (3.3.8)$$

y la clase predicha corresponde a su *signo*.

3.3.4 Árboles de decisión

Los árboles de decisión son un algoritmo de aprendizaje supervisado no paramétrico utilizado para tareas de clasificación. Los árboles de decisión son flexibles, ya que no incrementan la cantidad de parámetros a pesar de que se le agreguen más características, sus predicciones pueden ser categóricas o numéricas. se construyen con nodos y ramas (flechas). Los nodos definen el momento en el que se ha de tomar una decisión de entre la multitud de decisiones posibles, mientras mayor sea el número de nodos, mayor será la cantidad de decisiones y, por lo tanto, las posibles salidas del árbol serán mayores. Entre más complejo y detallado sea el problema, la cantidad de rutas posibles en el árbol incrementará y en consecuencia el análisis del mismo tendrá una complejidad mayor. Existen tres tipos de nodos: el nodo principal o nodo raíz, que es en donde se toma la primera decisión, que será aquella que divida mejor a los datos; los nodos interiores, que son aquellos en los que las variables se evalúan, pero no son los nodos en donde se realiza la predicción y los nodos hoja, en estos se realiza la predicción categórica o numérica. Las flechas indican las uniones entre cada uno de los nodos, y representan una decisión. La figura 3.3 ejemplifica un árbol de decisión.

Matemáticamente, un árbol de decisión se expresa de la siguiente manera. Dados los vectores de entrenamiento $x_i \in \mathbb{R}^n | i = 1, \dots, l$ y un vector de etiquetas de clase $y \in \mathbb{R}^l$, un árbol de decisión particiona de forma recursiva el espacio de características tal que las muestras con las mismas etiquetas de clase o valores objetivo similares sean agrupados conjuntamente.

Sea Q_m un dato en el nodo m con n_m características, para cada candidato

$\theta = (j, t_m)$ divide los datos en los subconjuntos $Q_m^{\text{izq}}(\theta)$ y $Q_m^{\text{der}}(\theta)$; donde j es una característica y t_m es un umbral. Esta separación se lleva a cabo considerando lo siguiente.

$$\begin{aligned} Q_m^{\text{izq}}(\theta) &= \{(x, y) | x_j \leq t_m\} \\ Q_m^{\text{der}}(\theta) &= Q_m \setminus Q_m^{\text{izq}}(\theta) \end{aligned} \quad (3.3.9)$$

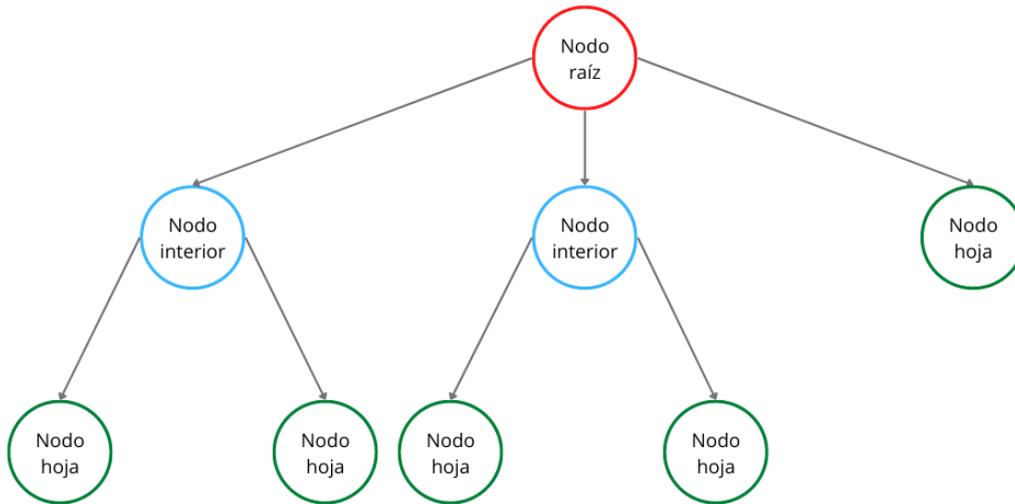


Figura 3.3: Ejemplo de un árbol de decisión

3.3.5 K-medias

K-medias es un algoritmo de agrupamiento cuyo paradigma de aprendizaje es el no supervisado en el que las muestras son extraídas de un conjunto de datos no etiquetado. Generalmente, los algoritmos de agrupamiento se utiliza para obtener patrones, procesos ocultos y realizar clasificación de muestras, su tarea principal es dividir la población en k grupos de tal forma que las muestras de la población compartan la mayoría de características con el grupo al que se han sido asignadas.

El algoritmo k-medias es un algoritmo de agrupamiento basado en centroides en el que se calcula la distancia entre cada muestra y el centroide al que han sido asignadas, la meta es identificar el número de grupos k y minimizar la distancia entre los centroides y los puntos de datos asignados a estos. El algoritmo puede

partirse en 5 pasos.

1. Elegir la cantidad de grupos. En este paso se definen la cantidad de grupos k en los que se agruparán los datos
2. Inicialización de los centroides. El centroide es el centro de un grupo, inicialmente se desconoce el centro de cada uno de los grupos, por lo que se seleccionan puntos aleatorios para definir los centroides.
3. Asignar los datos al centroide más cercano. Una vez inicializados los centroides, los datos se asignan a su centroide más cercano, para determinar dicho centroide, se calcula la distancia entre el punto x y los centroides C haciendo uso de la distancia euclidiana $d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$. El centroide ubicado a la menor distancia de cada punto será el asignado a dicho punto.
4. Reinicialización de los centroides. Los centroides se vuelven a inicializar calculando la media de todos los puntos de datos asignados al centroide. $C_i = \frac{1}{|N_i|} \sum x_i$.
5. Repetición de los pasos 3 y 4. Estos pasos se repetirán hasta que se encuentren los centroides óptimos y la asignación de los puntos de datos a sus grupos correctos no cambie.

3.3.6 Perceptrón

La neurona biológica puede ser considerada como la unidad básica de procesamiento del cerebro, esta célula especializada consta de un árbol de dendritas, el soma o cuerpo y un axón. Las conexiones entre neuronas se dan mediante la sinapsis, localizadas en las dendritas. Cada neurona recibe miles de conexiones que se “suman” a la entrada del axón, el resultado de esta “suma” es conocido como potencial de la membrana que si sobrepasa cierto umbral dado genera en la neurona un “disparo” (un impulso de voltaje) que será enviado a las neuronas conectadas a través de su axón.

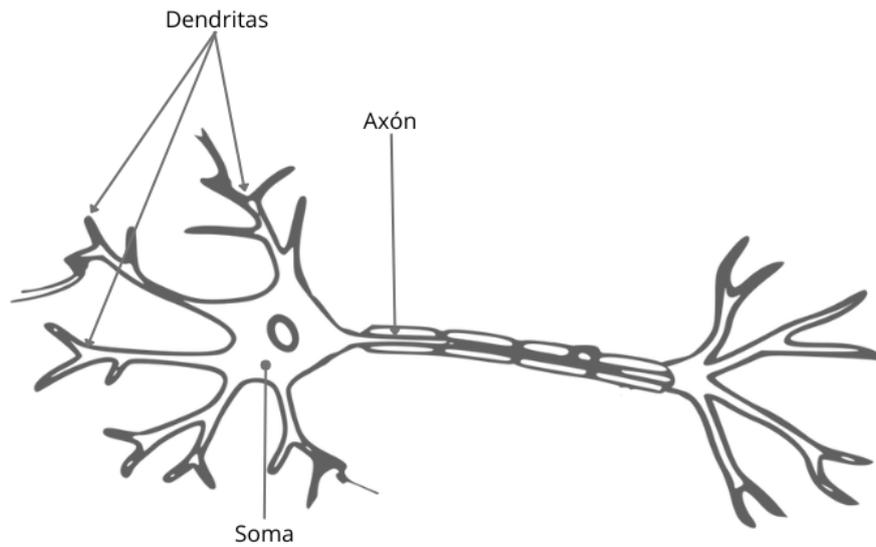


Figura 3.4: Neurona biológica

Inspirados en esta célula especializada, McCulloch y Pitts (1943) crearon una unidad de cálculo que intenta modelar el comportamiento de una neurona biológica, a esta unidad de cálculo se le llama unidad lógica de umbralado (TLU, por sus siglas del inglés). La forma en la que la TLU intenta modelar el comportamiento de una neurona biológica es la siguiente. Las sinapsis se modelan como números (pesos), cada uno de estos pesos multiplica a una entrada antes de ser enviada al equivalente del soma de la neurona real. En este lugar, todos los productos son sumados aritméticamente para dar como resultado la activación del nodo que se compara con un umbral predefinido. Si la activación excede este umbral, la TLU produce un 1, en caso contrario produce un 0. La figura 3.5 ejemplifica la neurona artificial.

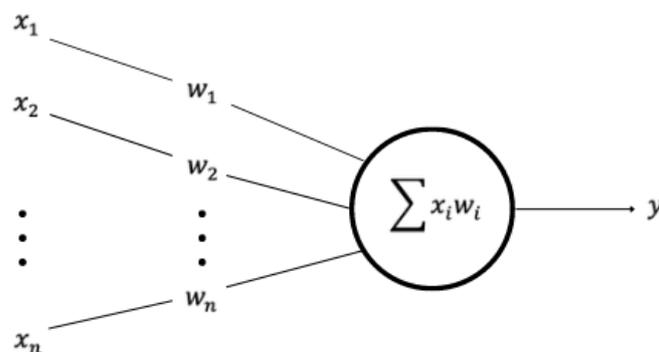


Figura 3.5: Neurona artificial (TLU)

Rosenblatt (1958) presenta el perceptrón que basó en la TLU de McCulloch y Pitts. Este algoritmo se compone por una capa de neuronas artificiales, es considerado como la red neuronal más simple, este modelo utiliza una matriz para representar las redes neuronales, actúa como un discriminador terciario que mapea las entradas x a un único valor de salida $f(x)$ utilizando dicha matriz. La función de activación se expresa como:

$$f(x) = \begin{cases} 1 & \text{si } w \cdot x - u > 0 \\ 0 & \text{de lo contrario} \end{cases} \quad (3.3.10)$$

donde w es el vector de pesos reales, $w \cdot x$ es el producto escalar del vector de pesos y el vector de entradas y u es el umbral que representa el grado de inhibición de la neurona. El umbral puede entenderse como la forma en la que se fija un nivel mínimo de actividad en la neurona para que la salida sea clasificada como activa (1) o inactiva (0). La figura 3.6

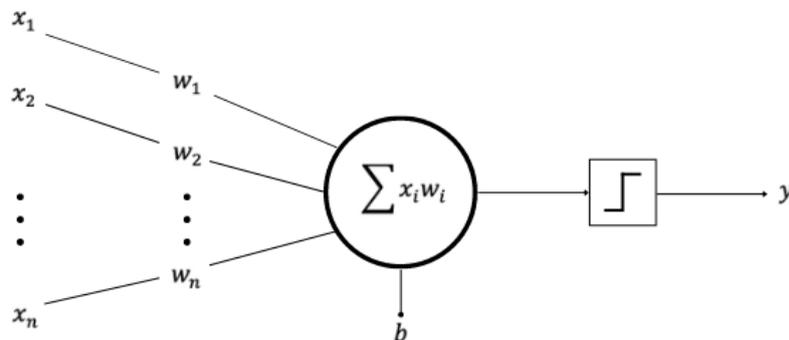


Figura 3.6: El perceptrón

La representación matemática del perceptrón es la siguiente.

$$f(x) = \text{signo}(\sum_{i=1}^n w_i x_i) \quad (3.3.11)$$

Cuando un punto se clasifica de manera incorrecta, es necesario ajustar los pesos para realizar su correcta clasificación. El algoritmo encargado de realizar estos ajustes es la regla de aprendizaje del perceptrón que se representa matemáticamente

como sigue.

$$w(t + 1) = w(t) + ex \quad (3.3.12)$$

donde $w(t + 1)$ es el peso nuevo, $w(t)$ es el peso anterior, e es el error de clasificación que se calcula como la diferencia entre el valor real de la etiqueta y el valor predicho por el perceptrón $e = t - a$, y x es la muestra clasificada.

3.3.7 Perceptrón multicapa (MLP)

El perceptrón multicapa se compone por la capa de entrada, una o más capas ocultas en la que cada una tiene muchas neuronas apiladas y una capa de salida, a diferencia del perceptrón en el que la función de activación fija un umbral de decisión, las neuronas del perceptrón multicapa pueden utilizar cualquier función de activación de manera arbitraria.

El MLP es un algoritmo de propagación hacia adelante, ya que de forma similar al perceptrón, cada neurona realiza el producto escalar de los pesos y entradas, pero la diferencia es que cada una de estas combinaciones se propaga a la capa siguiente. Cada una de las capas alimenta a las siguientes con los resultados de sus cálculos. La capa de entrada realiza la primera etapa de cálculos, la salida de cada una de las neuronas de la capa de entrada se convierte en la entrada de la primera capa oculta, este proceso se repite hasta alcanzar la capa de salida que recibe los valores de la última capa oculta y los transforma en valores de salida. Si se realiza una clasificación incorrecta de las muestras, entonces el perceptrón multicapa debe ajustar sus pesos, para lograrlo hace uso del algoritmo conocido como propagación hacia atrás. La figura 3.7 muestra el diagrama del perceptrón multicapa, la capa en el extremo izquierdo es la capa de entrada, las capas intermedias corresponden a las capas ocultas y la capa en el extremo derecho es la capa de salida.

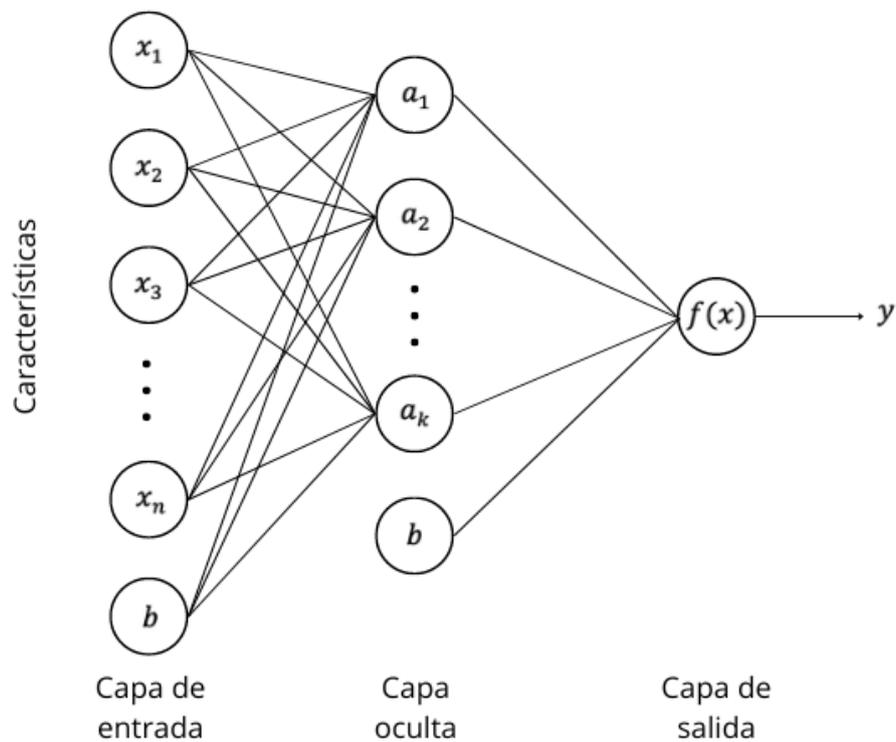


Figura 3.7: Perceptrón multicapa

3.3.8 Propagación hacia atrás

El algoritmo de propagación hacia atrás fue inventado por Rumelhart et al. (1986), es el mecanismo de aprendizaje que permite al perceptrón multicapa ajustar los pesos de la red, su objetivo es minimizar la función de costo. El algoritmo toma la información de la función de costo (p. ej. el error de clasificación de las muestras) y lo propaga hacia las capas anteriores, de ahí el nombre del algoritmo. La función de optimización típica es el descenso del gradiente. Una vez que las sumas ponderadas son propagadas hacia adelante a través de todas las capas, se calcula el gradiente del error cuadrático medio de la capa de salida, utilizando el gradiente el error es propagado hacia atrás en la red, actualizando los pesos de todas las capas hasta alcanzar la capa de entrada. Este proceso se repite hasta optimizar el valor de la función de costo.

3.3.9 Transformadores

Los transformadores son un tipo de arquitectura de red neuronal artificial que se utiliza para resolver el problema de la transducción o transformación de secuencias de entrada en secuencias de salida en aplicaciones de aprendizaje profundo. Fueron introducidos por Vaswani et al. (2017) en el artículo “*Attention is all you need*”. La arquitectura propuesta por el equipo se muestra en la figura 3.8

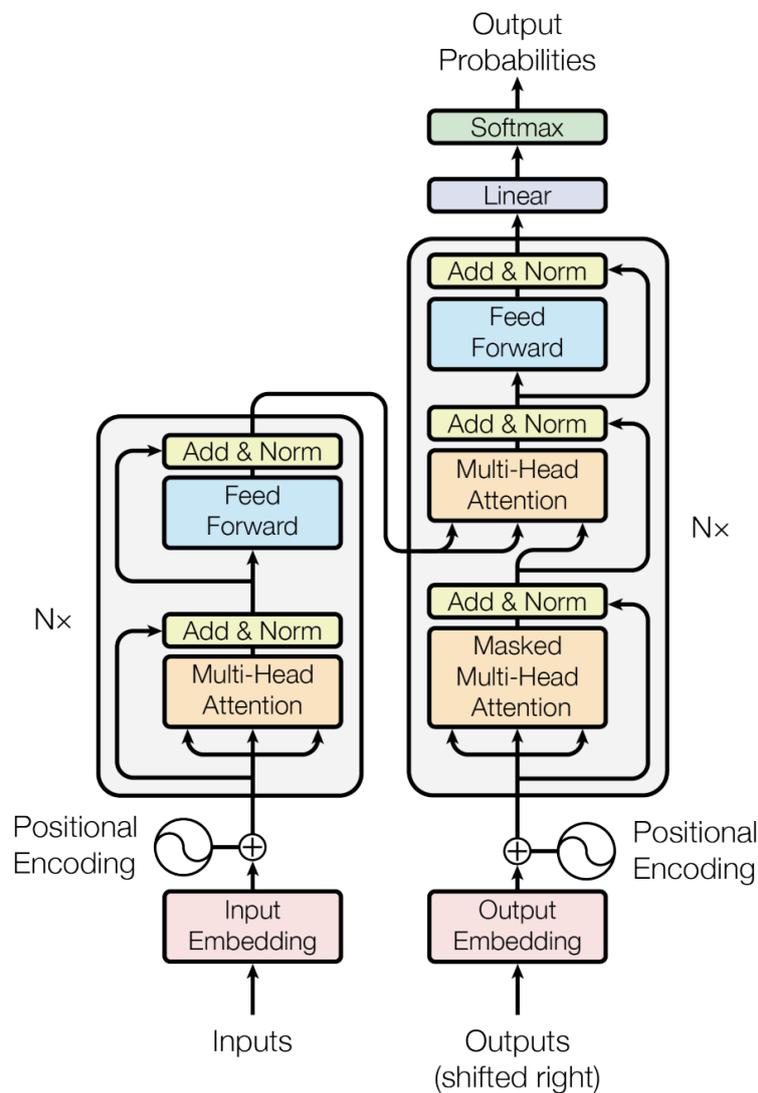


Figura 3.8: Arquitectura del modelo transformador (Vaswani et al., 2017)

La arquitectura se basa en la estructura codificador-decodificador, el codificador (porción izquierda) mapea las entradas a una secuencia de representaciones

continuas que toma el decodificador (porción derecha) para generar una salida. El transformador además utiliza capas completamente conectadas de atención y punto a punto tanto para el codificador como para el decodificador. Las capas de atención mapean una consulta y un conjunto de parejas valor-llave a una salida calculada como una suma ponderada de los valores en la que el peso de cada valor se calcula con una función de compatibilidad entre la consulta y su llave correspondiente.

La codificación posicional dota al modelo con la capacidad de utilizar el orden de la secuencia de los datos de entrada, ya que de alguna forma se debe alimentar la información de la posición relativa o absoluta de los tokens en la secuencia, debido a que el modelo no es recurrente ni realiza convoluciones sobre los datos de entrada para aprender esta posición. Las funciones matemáticas utilizadas para realizar la codificación posicional son las siguientes.

$$\begin{aligned} PE_{pos} &= \sin(pos)/10000^{2i/d_{\text{model}}} \\ PE_{pos,2i+1} &= \cos(pos)/10000^{2i/d_{\text{model}}} \end{aligned} \tag{3.3.13}$$

donde pos es la posición del token e i es la dimensión.

Capítulo 4

Descripción general de nuestra propuesta

En este capítulo se da una descripción general de la propuesta planteada. En primer lugar, se brindan aspectos importantes sobre la novedad de la propuesta, posteriormente se da una descripción detallada de cada uno de los pasos seguidos para el desarrollo de la propuesta.

El objetivo principal de nuestra propuesta es estimar el impacto que tienen las noticias sobre sus lectores, la forma en la que abordamos este problema fue utilizando la controversia, ya que es un indicador del nivel de acuerdo que hubo en los comentarios, si una noticia es controversial significa que hubo un acalorado debate en los comentarios de una publicación. Teniendo un impacto negativo sobre sus lectores. Para dar solución al problema de la predicción de la controversia desde la perspectiva de los lectores se planteó un *pipeline* compuesto por varios modelos de clasificación. Las etapas que componen el *pipeline* se describen brevemente a continuación, una descripción más detallada de cada una de estas etapas se da en las secciones siguientes de este capítulo.

1. Entrenamiento y evaluación de un modelo para clasificar la polaridad del texto de los tuits.

2. Desarrollo de un corpus de controversia.
3. Desarrollo e implementación de una rutina capaz de extraer el hilo de la conversación completo desde la API V2 de Twitter, que en conjunto con el modelo de clasificación de polaridad con mejor desempeño, obtengan la polaridad media de cada una de las conversaciones (respuestas a la publicación de la noticia) en el hilo de conversación.
4. Preprocesamiento de los datos de polaridad extraídos del hilo de conversación con el algoritmo k-medias para realizar una asignación de clase final y realizar un conteo de respuestas por cada clase de polaridad para posteriormente utilizarlos en conjunto con el corpus de controversia para entrenar un modelo de clasificación de controversia.
5. Entrenamiento y evaluación del modelo de controversia.

El proceso completo con cada una de las etapas se muestra en la figura 4.2.

Luego de revisar el estado del arte relacionado con la predicción de controversia, nos damos cuenta que han sido pocas las propuestas al respecto y aunque en todas se utiliza el análisis de sentimientos de los textos, los objetivos que se persiguen son distintos, ya que uno de los autores propone el uso de lexicones de sentimientos y de controversia para determinar que tan controversial ha sido una entidad en Twitter, otro de los autores utiliza el análisis de sentimientos para determinar que mientras mayor sea la polaridad negativa del texto de una noticia redactada por un autor, la controversia de la misma aumenta, y por último se utilizan los votos positivos y negativos (polaridad) de discusiones en reddit para usarlos como un indicador de controversialidad. Ninguno de los autores ha implementado una secuencia de tareas como la que se plantea en este trabajo, además tampoco se han utilizado algoritmos de aprendizaje automático para realizar la tarea de la clasificación de polaridad en los tuits de noticias y tampoco han utilizado un conteo del total de reacciones positivas negativas y neutras a los tuits para determinar la controversialidad de una noticia. Además, los trabajos mencionados en el estado del

arte que resuelven el problema de controversialidad fueron realizados sobre textos en inglés. Desde nuestro conocimiento, no existe trabajo previo que resuelva este problema de forma similar a lo que se plantea en nuestra propuesta.

Supongamos que tenemos una noticia sobre política, y en su hilo de conversación nos encontramos con lo siguiente, hay 7 comentarios en respuesta a esa noticia, y un tuit citado en el que el usuario que citó el tuit da su opinión y además 2 personas respondieron a esa cita. Tendríamos un total de 10 comentarios posteados por quienes leyeron la noticia o incluso solo la información proporcionada en el cuerpo del tuit de la noticia. Ahora, supongamos que tras analizarlas respuestas, las 7 respuestas al comentario original fueron negativas, la opinión de quién citó el tuit fue neutra, y las respuestas a la cita fueron positivas, como se observa en la figura 4.1.

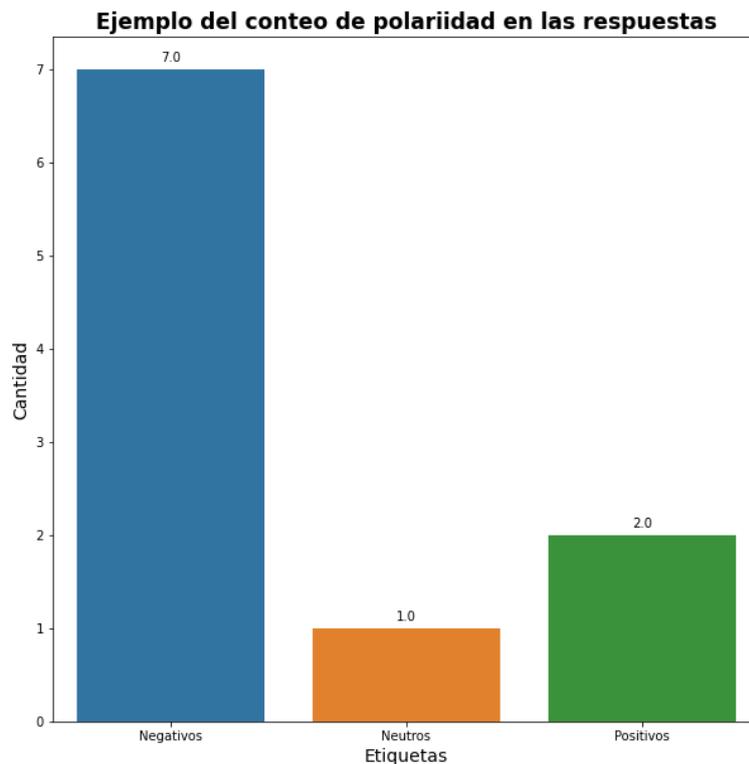


Figura 4.1: Ejemplo de la distribución de polaridad en el hilo de conversación de una noticia

Al analizar esta figura nos damos cuenta de que aunque la mayoría de comentarios fueron negativos, esa negatividad de parte de los lectores sería un indicador de que la noticia es controversial, ya que expresaron su desacuerdo con la misma.

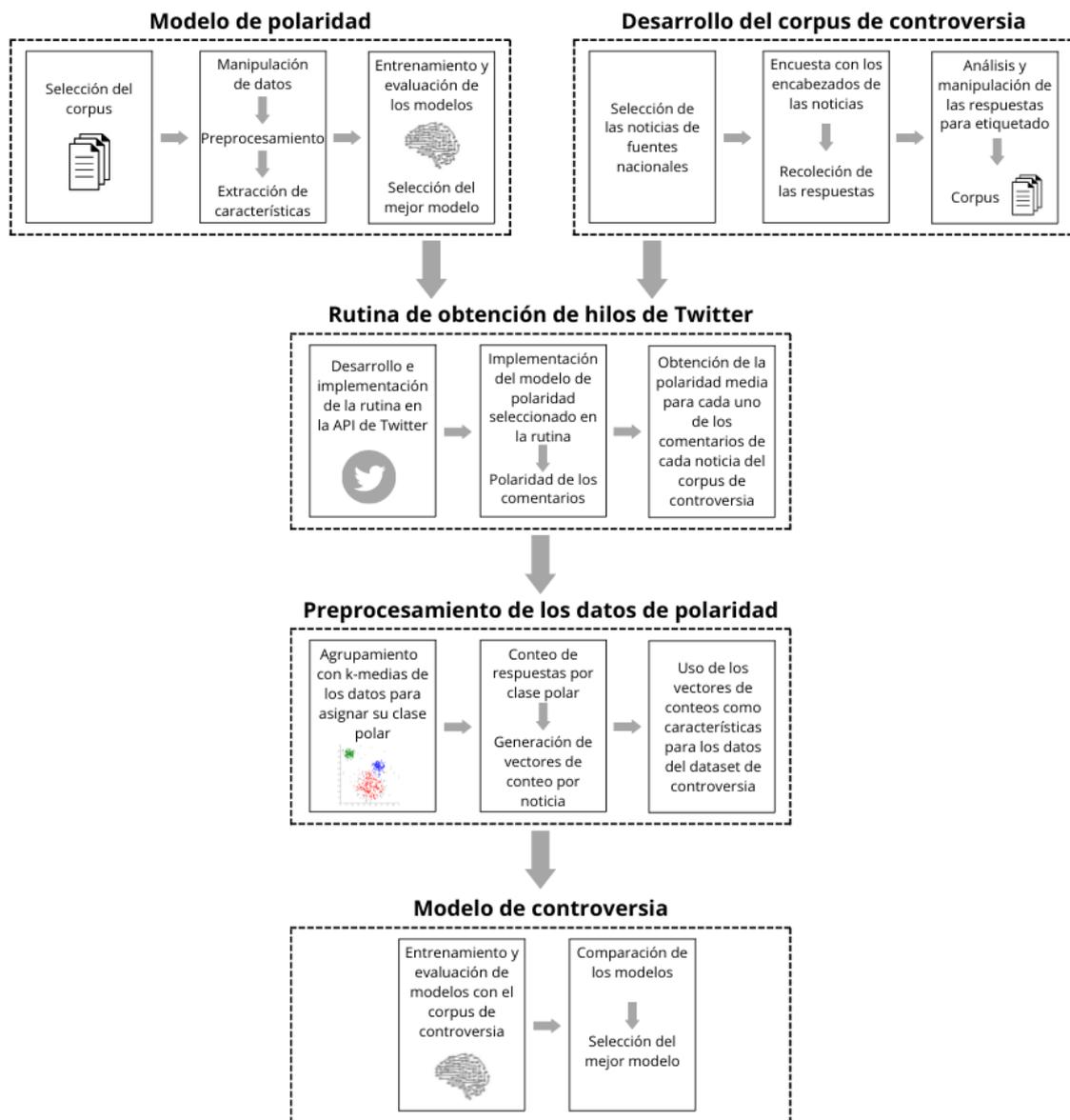


Figura 4.2: Etapas para el clasificador de controversialidad

4.1 Modelo de polaridad

En nuestra propuesta hacemos uso de la polaridad de las respuestas contenidas en el hilo de conversación de Twitter a una noticia publicada por fuentes nacionales para que tras una serie de pasos la controversialidad de la noticia en cuestión pueda ser clasificada. Es por eso que se propusieron varios modelos para realizar la tarea de clasificación, de entre los cuales se seleccionará el mejor para implementarlo en etapas posteriores.

4.1.1 Selección del corpus

En el estado del arte se dio una explicación superficial de los corpus disponibles para ejecutar la tarea de clasificación de polaridad. De los corpus ahí mencionados, el que contiene la mayor cantidad de tuits en español etiquetados con su polaridad es el corpus general del Taller de Análisis Semántico de la Sociedad Española de Procesamiento de Lenguaje Natural (TASS) (Villena-Roman et al., 2013). Este corpus contiene 60,798 tuits etiquetados como: muy positivo (P+), positivo (P), neutral (NEU), negativo (N), muy negativo (N+) y una etiqueta extra para aquellos tuits que no reflejan ningún sentimiento (NONE). La distribución de etiquetas es la que se observa en la figura 4.3.

La cantidad de tuits contenidos en el corpus es la razón por la que seleccionó este corpus, ya que en comparación, el segundo corpus con la mayor cantidad de tuits es el corpus COST con 34,634 tuits, pero este corpus no tiene la clase neutral, una razón más por la que se descartó.

4.1.2 Manipulación de los datos

Una vez seleccionado el corpus, los datos se manipularon para que el formato final de los mismos resultara útil para entrenar los modelos planteados.

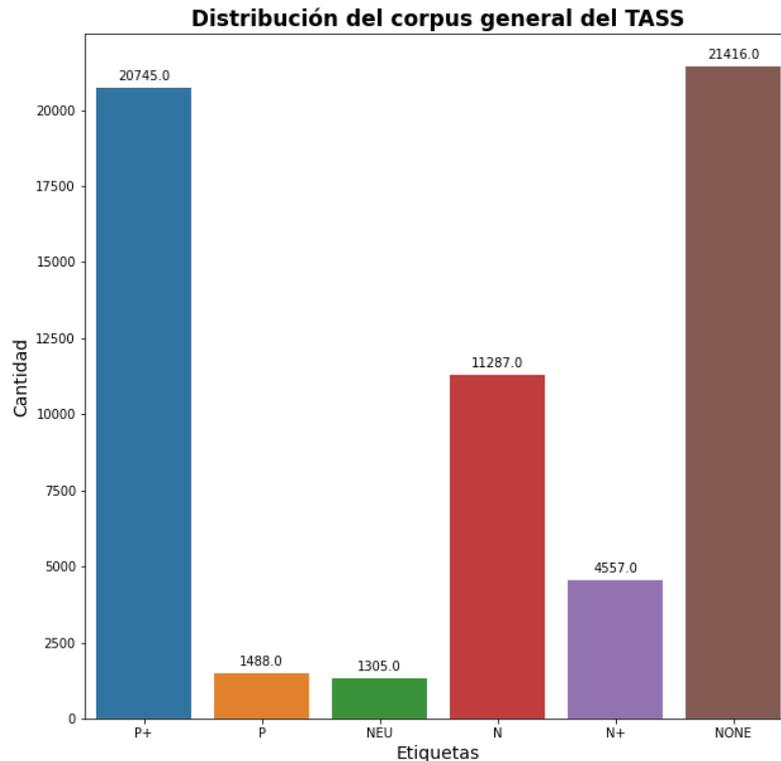


Figura 4.3: Distribución de las etiquetas de polaridad en el corpus general del TAAS

Para darle el formato necesario a los datos, se tomaron en cuenta las siguientes consideraciones.

- Ya que solo nos interesa conocer la polaridad básica de los tuits, es decir, que tan positivos, negativos o neutrales son, los datos con etiquetas (P+) y (P) se unieron para formar una nueva clase positiva (P), de igual forma los datos etiquetados como (N) y (N+) se fusionaron en una sola clase negativa (N).
- Los tuits etiquetados como (NONE) no expresan ningún sentimiento, por lo que se consideró que estos tuits se mantienen neutrales, por lo que las clases (NEU) Y (NONE) se fusionaron y se les asignó la etiqueta de clase neutral (NEU).

Tras este primer paso de manipulación, los datos quedaron distribuidos como se observa en la figura 4.4.

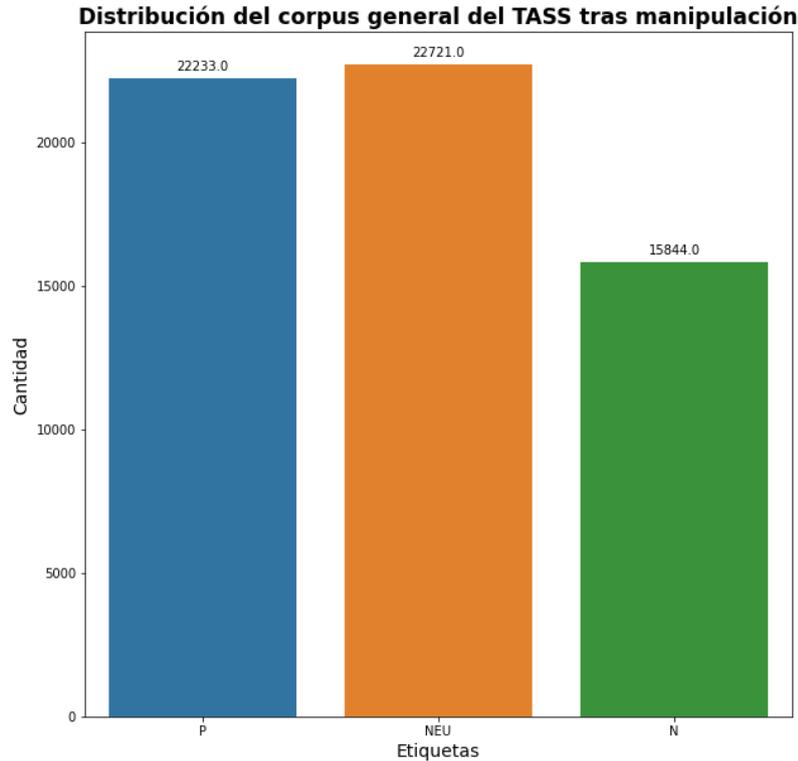


Figura 4.4: Distribución de las clases tras la manipulación del corpus general del TASS

Además, los datos se partitionaron en dos conjuntos, el de entrenamiento y el de prueba. El resultado de esta partición se muestra en la tabla 4.1. De un total de 60798 documentos, 48,638 se asignaron al conjunto de entrenamiento y 12,160 al de prueba.

Tabla 4.1: Distribución del corpus en los conjuntos de entrenamiento y prueba

Conjunto	Cantidad de documentos
Entrenamiento	48,638
Prueba	12,160
Total	60,798

La distribución de clases para los conjuntos de entrenamiento y prueba es la siguiente. Entrenamiento: 12,664 documentos pertenecen a la clase negativa (N), para la clase neutral (NEU) 18,171 documentos y para la clase positiva (P) se cuenta con un total de 17,803 documentos. Prueba: 3,180 pertenecen a la clase negativa,

4,550 pertenecen a la clase neutral (NEU) y 4,430 documentos se asignaron a la clase positiva (P). La distribución de clases para los conjuntos de entrenamiento y prueba se muestra gráficamente en las figuras 4.5 y 4.6 respectivamente.

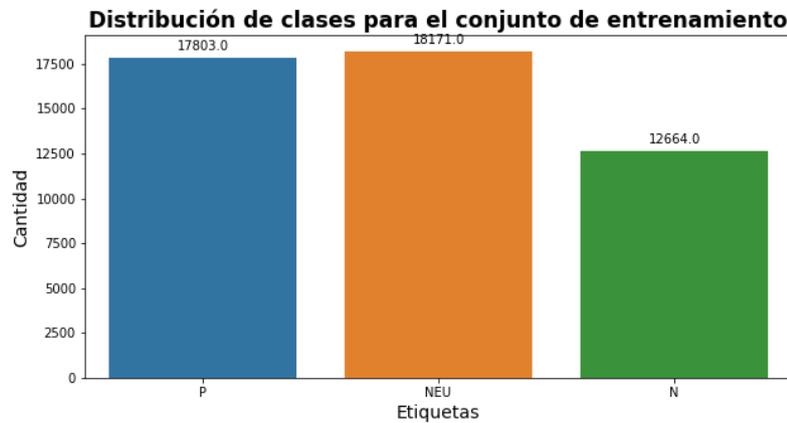


Figura 4.5: Distribución de clases para el conjunto de entrenamiento

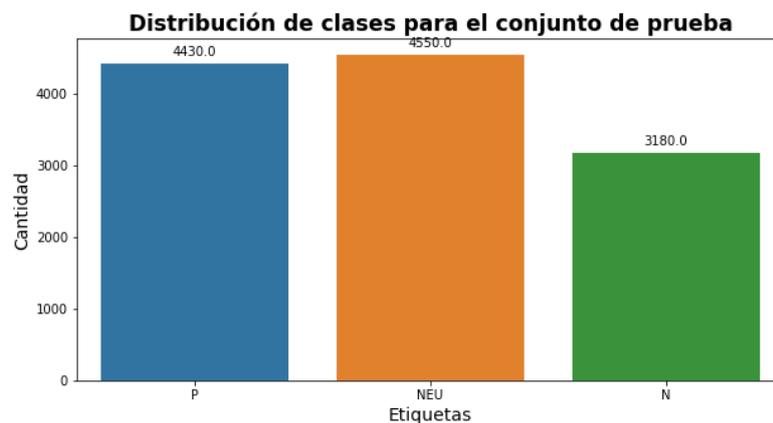


Figura 4.6: Distribución de clases para el conjunto de prueba

4.1.3 Preprocesamiento

Los autores del corpus general del TASS realizaron un preprocesamiento básico a los textos, removieron los emojis, los nombres de usuario de Twitter identificados como *@usuario* se les asignó el token *_USER_*, a los enlaces web se les asignó el token *_URL_* y a los *hashtags* identificados con el token *etiqueta* se les asignó el token *_HASHTAG_*. Además, todas las palabras fueron transformadas a minúsculas.

Algunos ejemplos de los textos contenidos en el corpus se dan a continuación.

- venga y vamos a apuntarnos todos rt *_USER_* : ya hay fecha ! marcar el jueves 15 de diciembre *_HASHTAG_* *_URL_*
- . *_USER_* nada más peligroso que un presidente que cree que sólo responde ante la historia (y no ante sus ciudadanos) .

Además, se tomó en cuenta que se debe realizar el preprocesamiento a textos extraídos directamente de Twitter, que contiene emojis, enlaces web, nombres de usuario, *hashtags*, entre otros elementos únicos de Twitter, algunos ejemplos de estos textos se dan a continuación.

- Se condonará impuesto predial por dos años a quienes paguen 2022.
La medida, para apoyar estrategia presidencial. Anuncia la mandataria @Claudiashein jornadas del bienestar para llevar alimentos y servicios a zonas marginadas.<https://bit.ly/38mMqnL>
- U+1F692 Las llamas consumieron una planta química en la ciudad de Omaha. Las autoridades instaron a evacuar a algunos residentes, aunque no se reportaron heridos. Mira las impresionantes imágenes.

Los textos crudos de Twitter contienen mucho texto que no aporta nada relevante al texto a la hora de efectuar la clasificación, es por esto que se procedió con los pasos descritos a continuación para preprocesar estos textos.

- Nombres de usuario: Los nombres de usuario identificados por un @ o *_USER_* al inicio de la palabra fueron removidos.
- Enlaces web: Todos los enlaces web fueron removidos.
- Entidades HTML: Las entidades propias del lenguaje HTML como “<p></p>” fueron removidas.
- Texto en mayúsculas: Todos los textos fueron convertidos a minúsculas.
- Caracteres repetidos: Las palabras en las que un caracter se repite más de dos veces (p.e. Holaaaaaaa) fueron recortadas a dos repeticiones del caracter (p.e. Holaa).

- Palabras auxiliares: Se removieron las palabras auxiliares (*stop words*) utilizando el corpus de palabras auxiliares en idioma español de la librería NLTK.
- Caracteres numéricos: Los números y las palabras que contenían caracteres numéricos fueron removidos del texto.
- Caracteres especiales: Todos los caracteres especiales fueron removidos del texto.
- Signos de puntuación: La mayoría de signos de puntuación no enriquecen el texto por lo que se removieron del texto.
- Emojis: En las redes sociales, muchas personas emplean los emojis para reflejar sentimientos, acciones, lugares, etc., se consideró que especialmente los emojis que se utilizan para reflejar sentimientos enriquecen el texto, por lo que, se decidió transformar los emojis a texto. Se empleó un diccionario de emojis que contiene el código del emoji y el texto en inglés, y se tradujo dicho texto al idioma español.

Al finalizar el preprocesamiento, los textos obtenidos fueron los siguientes.

- venga vamos apuntarnos fecha marcar jueves diciembre
- mas peligroso presidente cree solo responde historia ciudadanos
- condonara impuesto predial anos quienes paguen medida apoyar estrategia presidencial anuncia mandataria jornadas bienestar llevar alimentos servicios zonas marginadas
- camion bomberos llamas consumieron planta quimica ciudad omaha autoridades instaron evacuar algunos residentes aunque reportaron heridos mira impresionantes imagenes

4.1.4 Extracción de características

Para que un algoritmo de clasificación de aprendizaje automático tenga un buen desempeño, es necesario obtener un conjunto de características que representen adecuadamente las instancias. Para extraer las características se decidió usar el método

TF-IDF.

El método TF-IDF emplea métodos estadísticos para representar la relevancia de cada una de las palabras de un documento, considerando su presencia en la colección de documentos que está siendo analizada. Primero se genera una bolsa de palabras con el vocabulario de todos los documentos del corpus. A continuación se realiza el conteo de las veces que cada uno de los términos aparece en cada documento y se calcula la frecuencia de término (TF) al dividir ese conteo entre el total de palabras del documento. Luego se calcula la frecuencia inversa del documento (IDF) al calcular el logaritmo del producto inverso de la cantidad de documentos analizados y la cantidad de documentos en los que el término aparece. Obteniendo un vector con la representación numérica de los tokens para cada uno de los documentos. Como se analizan varios documentos al mismo tiempo, entonces el resultado final será una matriz TF-IDF en la que las filas representan cada uno de los documentos del corpus, las columnas el vocabulario de todos los corpus y cada una de las entradas corresponde al valor TF-IDF calculado para cada término de cada documento.

4.1.5 Modelos propuestos

Para la tarea de clasificación de polaridad, se propuso utilizar algoritmos de aprendizaje automático y de aprendizaje profundo, cabe mencionar que el modelo de aprendizaje profundo propuesto solo se utilizó para realizar la comparación de que tan buen desempeño tiene el aprendizaje automático en comparación con el aprendizaje profundo. Otro punto importante es que para la fase de entrenamiento los modelos fueron entrenados utilizando los parámetros que vienen definidos por defecto para que ningún modelo tenga ventaja sobre los demás.

Aprendizaje automático

En la literatura, los modelos para realizar la tarea de clasificación de polaridad más utilizados son modelos de aprendizaje automático. En el presente trabajo se propuso la utilización de tres modelos de aprendizaje automático de la librería de Python Scikit-learn, (Pedregosa et al., 2011) los cuales se enlistan a continuación.

- Regresión logística (*LogisticRegression*)
- Naïve Bayes multinomial (*MultinomialNB*)
- Máquina de soporte vectorial (*SVC*)

Entre paréntesis aparece el nombre del algoritmo de Scikit-learn. Además, ya que nuestro problema es multi-clase, se utilizó la estrategia uno contra todos (*OneVsRestClassifier*) de Scikit-learn. Esta estrategia resuelve el problema multi-clase al ajustar un clasificador por cada una de las clases, en nuestro caso, se ajustaron tres clasificadores. Para cada clasificador, se realiza una comparación de su clase contra todas las demás, asignando la etiqueta de clase a aquella que sea mayoritaria. Una de las ventajas de esta estrategia es su eficiencia computacional y su interpretabilidad, ya que como cada clase es representada por un clasificador, es posible obtener conocimientos sobre esa clase al inspeccionar su respectivo clasificador. Es una de las estrategias más utilizadas para la clasificación multi-clase.

Aprendizaje profundo

Adicionalmente, se probó el hacer la tarea de clasificación de polaridad con métodos de aprendizaje profundo, para tal efecto se propuso hacer uso del transformador BETO (Cañete et al., 2020) al cual se le realizó un ajuste fino con el corpus de entrenamiento. BETO es un modelo BERT (Devlin et al., 2018) entrenado con un gran corpus en idioma español, es de tamaño similar a BERT *base* que cuenta con un total de 12 capas en la pila de codificación y un total de 110 millones de parámetros, y fue entrenado utilizando la técnica de enmascaramiento de palabras completas (*whole word masking*).

Ya que BETO modelo pre-entrenado para realizar tareas generales en español, es necesario ajustar los parámetros de este modelo para que aprenda una tarea particular, en nuestro caso, la clasificación de polaridad. Las aplicaciones principales del ajuste fino son el acelerar el proceso de entrenamiento aprovechando los conocimientos previamente aprendidos por la red y superar el problema de entrenamiento con conjuntos de datos pequeños.

4.1.6 Selección del mejor modelo

El enfoque del presente trabajo son los algoritmos de aprendizaje automático mencionados en la sección anterior. Aunque se experimentó con el modelo de aprendizaje profundo BETO, los resultados de esos experimentos se utilizaron para tener un panorama del desempeño de los algoritmos de aprendizaje automático y de aprendizaje profundo.

Para seleccionar el mejor modelo, es se evaluó el desempeño de los mismos con datos que no vieron durante la etapa de entrenamiento (conjunto de prueba) para observar que tan buenos son los modelos con estos datos. Ya que al implementar el modelo en la rutina de obtención de hilos de conversación en Twitter, todos esos datos serán nuevos para el mismo.

Las métricas propuestas para evaluar el desempeño de los modelos se enlistan a continuación.

- Precisión (*precision*). La precisión evalúa la calidad del modelo al medir cuántos del total de clasificados como positivos son en verdad positivos y se calcula como: $precision = \frac{verdaderos\ positivos}{verdaderos\ positivos + falsos\ positivos}$
- Exhaustividad (*recall*). La exhaustividad indica la cantidad de verdaderos positivos que el modelo es capaz de recuperar y se calcula con la siguiente expresión: $recall = \frac{verdaderos\ positivos}{verdaderos\ positivos + falsos\ negativos}$
- Valor-F (*F1-score*). El valor-F es una función que relaciona la precisión y la exhaustividad, mide el balance que existe entre estas otras dos métricas y se

calcula con la siguiente fórmula: $F1 = \frac{precision \cdot recall}{precision + recall}$

Por lo tanto, el modelo se seleccionará tomando en consideración los resultados obtenidos para estas métricas durante el proceso de evaluación con el conjunto de datos de prueba.

4.2 Desarrollo del corpus de controversia

Debido a la falta de corpus que contengan la información necesaria para realizar la clasificación de controversia, se decidió desarrollar un corpus propio, seleccionando noticias de fuentes nacionales, el proceso que se llevó a cabo para el desarrollo del corpus se describe paso a paso en esta sección.

4.2.1 Selección de las noticias

En Twitter existen muchas fuentes nacionales de noticias, pueden ser noticieros que tienen su propia página web, y que postean sus artículos en la plataforma, pero lo más común es que sean periódicos que tienen una plataforma en línea y hacen uso de Twitter para difundir enlaces a sus artículos. Otro aspecto a considerar es el contenido que publican, ya que algunos se especializan en deportes o en finanzas, ese tipo de periódicos quedó descartado, y nos centramos en aquellos que publican información más general, tienen página en línea y además tienen una cuenta en Twitter. Tras el proceso de selección, se eligieron El Universal¹, La Jornada², CNN en Español³, BBC News Mundo⁴, Milenio⁵, Reforma⁶ y Proceso⁷.

De estas fuentes, un total de 60 noticias fue seleccionado para crear el cor-

¹<https://www.eluniversal.com.mx/>

²<https://www.jornada.com.mx/>

³<https://cnnespanol.cnn.com/>

⁴<https://www.bbc.com/mundo>

⁵<https://www.milenio.com/>

⁶<https://www.reforma.com/>

⁷<https://www.proceso.com.mx/>

pus de controversia. El criterio para elegir las noticias fue elegir aquellas que traten temas que desde nuestro punto de vista eran polémicos como política, economía o seguridad, y temas poco polémicos como la ciencia y tecnología o la cultura.

4.2.2 Encuesta

Una vez seleccionadas las 60 noticias, se generó una encuesta en la que los participantes tenían cuatro opciones para elegir enlistadas a continuación.

- Nada controversial
- Poco controversial
- Controversial
- Altamente controversial

En la encuesta únicamente se incluyó el contenido del tuit del que fueron extraídas, porque se consideró que para tener una respuesta fiel sobre la opinión de quienes leen el encabezado de una noticia, la mejor opción sería proveer a los encuestados con los datos que leerían al ver una publicación en Twitter.

4.2.3 Proceso de etiquetado

Para etiquetar la colección de noticias, la encuesta se envió a múltiples posibles etiquetadores, en total participaron 31 personas en el proceso de etiquetado. Para contestar la encuesta, a los participantes se les dieron las siguientes instrucciones.

1. De acuerdo con los siguientes encabezados de noticias, elija una opción que describa mejor para usted el nivel de controversialidad (polémica) de la noticia dada la siguiente definición. Controversialidad: discusión de opiniones contrapuestas entre dos o más personas.
2. Para este caso en particular, la controversialidad se da en la división de opiniones de las personas en los comentarios o respuestas a la noticia. Es decir,

entre más divididos crea usted que serán los comentarios, el nivel de controversialidad será mayor.

La distribución de los participantes por grupos de edad y grado máximo de estudios se muestra en las figuras 4.7 y 4.8 respectivamente.

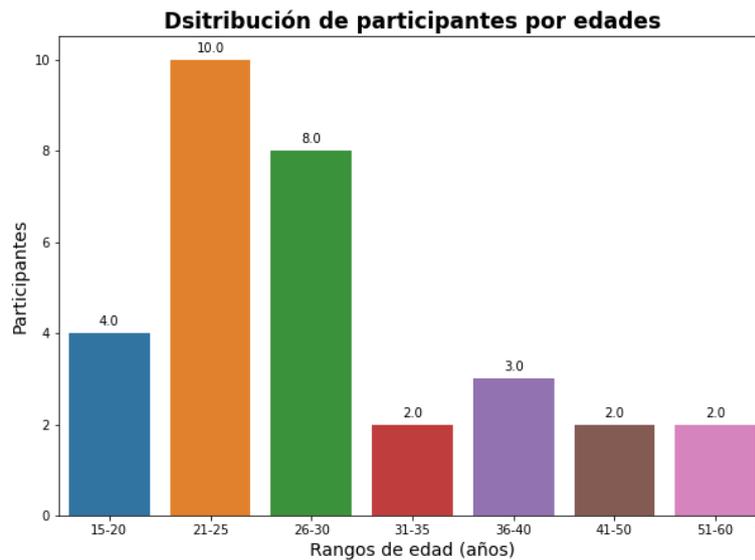


Figura 4.7: Distribución de participantes por grupo de edad

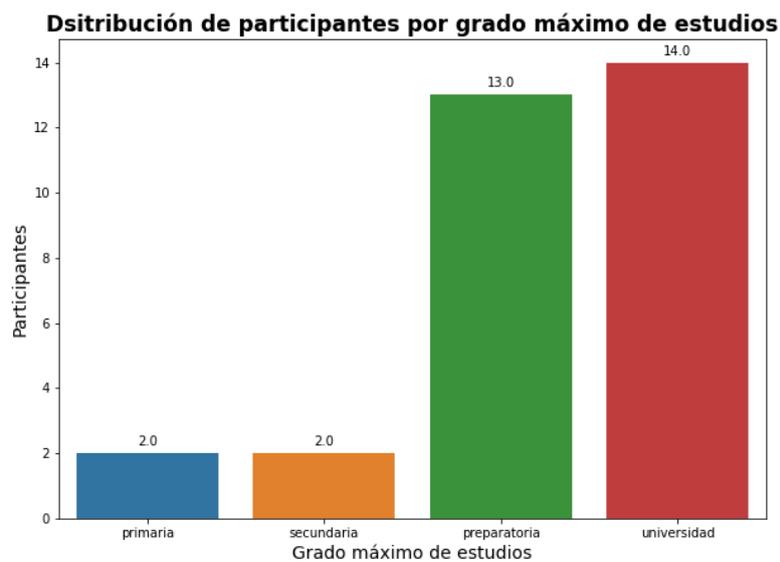


Figura 4.8: Distribución de participantes por grado máximo de estudios

4.2.4 Análisis del corpus

Habiendo finalizado el proceso de encuestado, se procedió a recolectar las respuestas de los participantes para manipularlas y poder analizarlas. La opción con mayor número de respuestas fue asignada como etiqueta de clase durante una primera etapa. La distribución de clases durante la primera iteración de etiquetado se muestra en la figura 4.9 en la que podemos observar que la mayoría de noticias fueron clasificadas en poco controversial y controversial, y muy pocas fueron consideradas como nada controversial y altamente controversial.

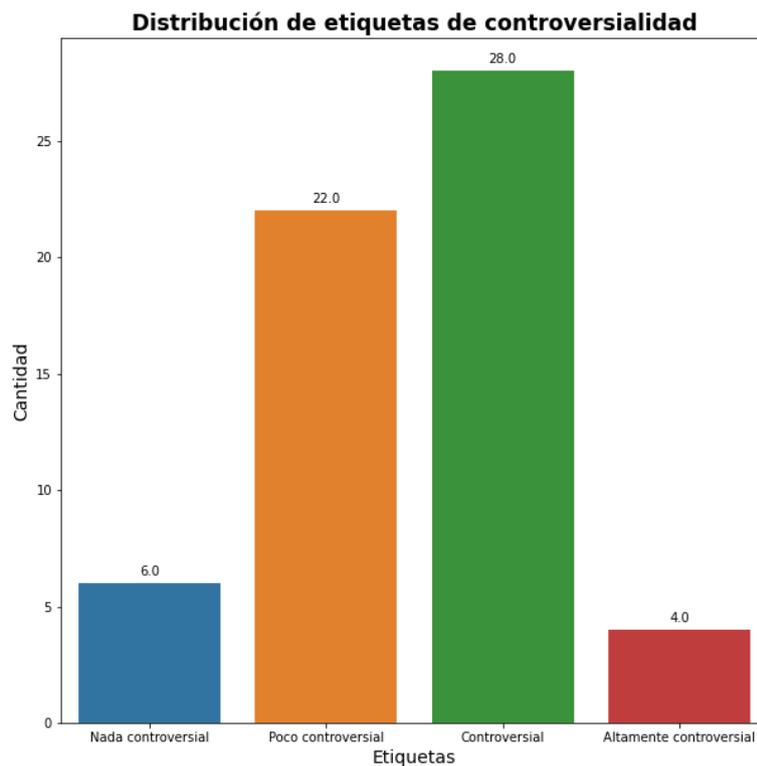


Figura 4.9: Distribución de clases en el corpus de controversialidad tras la primera iteración de etiquetado

4.2.5 Selección de las etiquetas

Para asignar las etiquetas finales al conjunto de datos se debe de tener un buen indicador de que tan buenas fueron las respuestas para las clases. El grado de acuerdo o desacuerdo entre anotadores es un buen indicador de que tan confiable es utilizar las clases para entrenar un algoritmo de aprendizaje automático. Para evaluar el grado de acuerdo entre anotadores, se utilizó el índice kappa de Cohen. El algoritmo se implementó con la librería Scikit-learn. Este algoritmo es una estadística que mide el acuerdo entre los anotadores de un corpus. Se define como:

$$\kappa = (p_o - p_e)/(1 - p_e) \quad (4.2.1)$$

donde p_o es la probabilidad empírica de acuerdo en la etiqueta asignada a cualquier muestra; p_e es el acuerdo esperado del proceso de asignación de etiquetas de los anotadores. Esta métrica es un número en el rango $[-1, 1]$ y los resultados se interpretan de la siguiente forma, cuando se tiene un grado de acuerdo mayor a 0,8 generalmente se puede considerar como un buen grado de acuerdo, una puntuación igual o menor a 0 indica que no hubo acuerdo y se puede decir que las etiquetas fueron asignadas aleatoriamente. Esta métrica solo se puede calcular para el caso binario (es decir para dos anotadores), ya que tenemos 31 anotadores, se realizó el cálculo del índice kappa para cada una de las posibles combinaciones de anotadores. Durante este proceso, no se consideró el grado de acuerdo del anotador con el mismo, y las combinaciones que ya se habían calculado (es decir, el grado de acuerdo entre los anotadores 1-3 es el mismo que 3-1) se iban descartando, se utilizó un algoritmo basado en el cálculo de las diagonales únicas de un polígono y además se consideraron los vértices para determinar las operaciones que se tenían que efectuar.

En primera instancia, se consideraron las clases tal como se reportaron en la sección anterior y se obtuvo la media, desviación estándar, y grado máximo y mínimo de acuerdo, estas métricas se pueden observar en la tabla 4.2.

Tabla 4.2: Grado de acuerdo entre anotadores para las 4 clases

Métrica	Valor
Media	0.074
Desviación estándar	0.096
Acuerdo mínimo	0.386
Acuerdo mínimo	-0.179

Como podemos observar, los valores para la media y el grado de acuerdo máximo son muy bajos, lo que nos indica que no hubo acuerdo entre los anotadores y se puede decir que el proceso de etiquetado en sí mismo fue controversial. Es por eso que se tomó la decisión de binarizar el corpus para ver si de alguna forma el grado de acuerdo se podía ver incrementado al reducir el problema de clasificación a un problema binario. Se realizó el cálculo del índice kappa entre anotadores para los siguientes casos.

En primer lugar, se consideró redistribuir las clases nada controversial y poco controversial en una sola clase no controversial y las clases controversial y altamente controversial en la clase controversial. Los valores para la media, desviación estándar y grados máximo y mínimo de acuerdo se muestran en la tabla 4.3.

Tabla 4.3: Grado de acuerdo entre anotadores para la primera redistribución de clases

Métrica	Valor
Media	0.143
Desviación estándar	0.155
Acuerdo mínimo	0.596
Acuerdo mínimo	-0.358

Como se puede observar en la tabla anterior, el grado máximo de acuerdo y la media incrementaron su valor, esto nos motivó a realizar una segunda redistribución de clases, dejando la clase nada controversial como no controversial y las clases poco controversial, controversial y altamente controversial se unieron para formar una sola clase controversial. Los valores obtenidos para esta redistribución se muestran en la tabla 4.4.

Con el análisis de anotadores calculado, se optó por redefinir las etiquetas

Tabla 4.4: Grado de acuerdo entre anotadores para la segunda redistribución de clases

Métrica	Valor
Media	0.145
Desviación estándar	0.180
Acuerdo mínimo	1.0
Acuerdo máximo	-0.223

de clase de la siguiente forma. Nada controversial y poco controversial se unieron para conformar la clase no controversial, y las clases, controversial y altamente controversial, se juntaron para conformar la clase controversial. Esta decisión se tomó, ya que si dejáramos la clase nada controversial como la única que conforma la clase no controversial, solo tendríamos 6 datos para el entrenamiento de los modelos, en cambio, al redistribuir las clases de la manera anteriormente planteada, tendremos 28 datos en la clase no controversial y 32 datos en la clase controversial. Se optó por realizar la asignación final de etiquetas de clase, manteniendo un punto intermedio en el que las clases están balanceadas y el grado de acuerdo entre anotadores incrementa.

4.2.6 Corpus de controversia

Finalmente, el corpus de controversia quedó conformado por 60 noticias. El corpus contiene la siguiente información.

- Los identificadores de los tuits (*tweet id*).
- La etiqueta de clase (no controversial o controversial) para cada uno de los tuits.

Los encabezados de las noticias no se incluyeron, ya que se quiere conocer si una noticia es controversial o no, haciendo uso de la polaridad de las respuestas a dicha noticia, las características para realizar la clasificación de la controversialidad de las noticias fueron extraídas tras el preprocesamiento de los datos de polaridad obtenidos de Twitter, en pocas palabras, estas características son vectores que con-

tienen el conteo de respuestas positivas negativas y neutrales para cada una de las noticias del corpus.

4.3 Rutina de obtención de hilos en Twitter

Para poder realizar el análisis del hilo de conversación de una noticia, se desarrolló una rutina que obtiene una noticia a partir de su id, de esta noticia se obtiene la lista de los id's de los tuits que respondieron a la publicación principal, y una lista de los tuits que citaron a la publicación original. Con la lista de los tuits que respondieron a la publicación principal, se obtuvieron los textos de dichas respuestas y con la lista de tuits que citaron a la publicación original se obtuvieron todos los tuits citados. Cabe mencionar que los tuits citados dan la oportunidad a quien cita el tuit de dar su opinión en una publicación aparte, además estos tuits citados pueden tener una nueva conversación dentro del cuerpo del mismo. Este proceso se realizó de manera recursiva para obtener las citas de las citas y con la información obtenida se generaron árboles de tuits citados para poder observar el nivel de profundidad de los mismos. Para fines de este trabajo, se definirán como bloques de conversación todos los comentarios que aparezcan en el cuerpo de una publicación, es decir, un bloque de conversación serían todos los comentarios de la publicación original, otro bloque de conversación sería el texto contenido en la publicación de un tuit citado y sus comentarios.

4.3.1 Descripción del funcionamiento

La rutina desarrollada obtiene los comentarios en la conversación del tuit principal, y la lista de los tuits que citan al tuit original, para lograr esto la rutina sigue los siguientes pasos.

1. Con el id de tuit que se está analizando se obtiene la siguiente información:

id's de los tuits comentando la publicación y la lista de tuits citando a la publicación en caso de que existan.

2. Con los id's de los tuits comentando la publicación se obtiene el texto de cada comentario
3. Se preprocesa el texto para cada uno de los comentarios obtenidos
4. El texto preprocesado es vectorizado y pasado al clasificador de polaridad que realiza la predicción de polaridad del tuit, los vectores de polaridad resultantes son almacenados temporalmente hasta que se completa la clasificación del bloque de conversación en su totalidad.
5. Con los vectores de polaridad del bloque de conversación se calcula el promedio para cada una de las clases y se almacena este vector de polaridad media.
6. En caso de que la publicación contenga tuits citados, se recorre cada uno de estos tuits repitiendo los pasos 3, 4 y 5 hasta terminar con los tuits citados.
7. En caso de que los tuits citados tengan tuits citados, se repite el paso 6 hasta que se obtiene el hilo de conversación completo para la publicación original, entonces los datos de salida del algoritmo será un arreglo de vectores de polaridad media para cada uno de los bloques de conversación analizados.
8. Se genera un árbol de tuits citados para observar la profundidad del hilo de conversación.

4.4 Preprocesamiento de los datos de polaridad

Los datos de salida de la rutina de obtención de hilos de conversación en Twitter (el arreglo de vectores de polaridad media para cada bloque de conversación) aún tienen que ser preprocesados para poder generar vectores de características para el clasificador de controversialidad.

4.4.1 Descripción del preprocesamiento

El preprocesamiento consta de dos etapas, la primera es el agrupamiento de los vectores de polaridad media para que sean asignados a la clase con la que compartan más características, la segunda es realizar el conteo de respuestas por bloque de polaridad con estos datos. Supongamos que hubo 10 bloques de conversación en el hilo de la publicación, por lo tanto, tendremos un arreglo que contenga 10 vectores de polaridad media, al agruparlos 6 de ellos fueron negativos, 2 neutrales y 2 positivos, entonces se generará un vector con estas características ordenado como $[N, NEU, P]$ en consecuencia el vector de características para esa noticia será $[6, 2, 2]$. Este proceso se llevará a cabo para cada una de las noticias del corpus de controversia, obteniendo un total de 60 vectores de conteo de polaridad.

El algoritmo seleccionado para realizar el agrupamiento fue k-medias, ya que para los datos de polaridad media no hay etiquetas con las que entrenar un algoritmo de aprendizaje supervisado, es por eso que se optó por utilizar este algoritmo de aprendizaje no supervisado, porque no requiere de ejemplos para su entrenamiento.

4.5 Modelo de controversia

4.5.1 Modelos propuestos

En un primer experimento se propuso la utilización de reglas para realizar la clasificación de controversialidad, las reglas fueron propuestas por nosotros al realizar un análisis visual de los conteos de polaridad, por tal motivo el corpus completo se utilizó para evaluar el desempeño del modelo basado en reglas. Las reglas propuestas son las siguientes.

- Si las clases neutral o positivo contenían más del 90 % del total de los comenta-

rios, entonces la muestra se clasifica como no controversial, ya que se consideró que si una de estas dos clases concentra la mayoría de los comentarios, la mayoría de las personas que respondieron a la publicación estuvieron de acuerdo con la noticia o simplemente no les afectó de forma negativa.

- En caso contrario, la noticia se clasifica como controversial, ya que puede que la mayoría de comentarios fueran negativos, lo que significa que la gran mayoría de usuarios estuvo en desacuerdo con la noticia, o que se generó un acalorado debate en los comentarios.

Este modelo se utilizó para tener una referencia, además de este modelo, se propuso la utilización de los siguientes algoritmos, que fueron implementados con la librería Scikit-learn.

- Árboles de decisión (*DecisionTreeClassifier*)
- Perceptron multicapa (*MLPClassifier*)

Las métricas de evaluación consideradas para evaluar el desempeño de los modelos son las mismas que las utilizadas para evaluar los modelos de polaridad (precisión, exhaustividad y valor-F).

Capítulo 5

Experimentos y resultados

En este capítulo se explican los experimentos con los modelos involucrados en cada una de las etapas de la solución planteada, descrita en el capítulo 4. Ya que desde nuestro conocimiento, la solución propuesta no tiene puntos de referencia contra los cuales comparar, se realizará una comparación contra el estado del arte de aquellas etapas que si cuentan con antecedentes en la literatura, como lo es el modelo para clasificar la polaridad.

5.1 Modelo de polaridad

En esta sección se presenta la configuración utilizada para entrenar y evaluar el desempeño de los modelos de clasificación de polaridad planteados en el capítulo anterior y los resultados obtenidos de la experimentación.

El objetivo de estos experimentos es obtener el modelo de clasificación con el mejor desempeño para que pudiera ser implementado en la rutina de extracción de hilos de conversación de Twitter posteriormente. La selección dependerá de las métricas de evaluación presentadas en el capítulo 4

5.1.1 Configuración de los experimentos

Para la experimentación se separó el corpus general del TASS en dos conjuntos, el de entrenamiento y el de prueba, la división se realizó con la función *train_test_split* de la librería Scikit-learn. Para realizar la partición del corpus, se consideró utilizar el 80 % de los datos del corpus para entrenamiento y 20 % para prueba, además se le dio a la función la semilla 42 para poder replicar los experimentos. Los clasificadores de Scikit-learn fueron inicializados con la semilla 42, y se utilizó el algoritmo *TfidfVectorizer* para realizar la extracción de características y la vectorización de los textos. El modelo pre-entrenado BETO fue utilizado con la semilla que tiene por defecto.

5.1.2 Resultados de la primera fase de experimentación

Las tablas 5.1, 5.2 y 5.3 muestran los resultados obtenidos por los clasificadores de aprendizaje automático. Por otra parte, en la tabla 5.4 se muestran los resultados obtenidos por el modelo BETO. Estos resultados fueron obtenidos al evaluar los modelos con el conjunto de prueba.

Tabla 5.1: Resultados para *LogisticRegression*

	precision	recall	f1-score
negativo	0.79	0.49	0.61
neutral	0.69	0.51	0.58
positivo	0.88	0.63	0.73
promedio	0.79	0.55	0.65

Tabla 5.2: Resultados para *MultinomialNB*

	precision	recall	f1-score
negativo	0.83	0.29	0.43
neutral	0.79	0.22	0.34
positivo	0.84	0.55	0.67
promedio	0.82	0.36	0.48

Con estos resultados pudimos observar el desempeño para cada uno de los modelos propuestos, como se observa, los algoritmos de aprendizaje automático

Tabla 5.3: Resultados para SVC

	precision	recall	f1-score
negativo	0.80	0.50	0.61
neutral	0.72	0.48	0.57
positivo	0.88	0.63	0.73
promedio	0.80	0.54	0.64

Tabla 5.4: Resultados para BETO

	precision	recall	f1-score
negativo	0.77	0.69	0.73
neutral	0.78	0.80	0.74
positivo	0.78	0.83	0.80
promedio	0.78	0.77	0.76

superan a BETO en precisión. Aunque en exhaustividad y valor-F obtuvieron valores más bajos, la comparación con un modelo tan robusto como lo es BETO, nos sirvió para determinar que a pesar de que a los algoritmos de aprendizaje automático no se les dio ningún tipo de ventaja contra BETO, su desempeño a la hora de clasificar la polaridad no es malo.

5.1.3 Resultados de la segunda fase de experimentación, pruebas de robustez

Finalmente, se realizaron pruebas de robustez a los algoritmos de aprendizaje automático, al reducir el tamaño de su vocabulario, esta reducción se realizó al considerar una frecuencia de término mínima de 2, 3 y 5 conteos, los resultados de estas pruebas se muestran en las tablas 5.5 y 5.6.

Se observó que el clasificador con mayor capacidad de generalización, a pesar de que su número de características se vio reducido en un 63 % es la máquina de soporte vectorial (SVC) que fue capaz de realizar la clasificación de forma similar a cuando no se le redujeron las características. Dados los resultados obtenidos de estos experimentos, se decidió utilizar el clasificador SVC con frecuencia de término mínima de 3 en las etapas posteriores de la solución.

Tabla 5.5: Resultados para los modelos de aprendizaje automático tras la prueba de robustez, mín. TF = 1 y 2

	Frecuencia de término mínima					
Modelo	Uno			Dos		
	Precision	Recall	F1	Precision	Recall	F1
LR	0.79	0.55	0.65	0.79	0.56	0.65
MultiNB	0.82	0.36	0.48	0.78	0.44	0.55
SVC	0.80	0.54	0.64	0.80	0.55	0.65
Características	46,791			24,030		

Tabla 5.6: Resultados para los modelos de aprendizaje automático tras la prueba de robustez, mín. TF= 3 y 5

	Frecuencia de término mínima					
Modelo	Tres			Cinco		
	Precision	Recall	F1	Precision	Recall	F1
LR	0.78	0.57	0.66	0.78	0.57	0.66
MultiNB	0.77	0.46	0.57	0.76	0.48	0.59
SVC	0.80	0.56	0.66	0.79	0.57	0.66
Características	17,158			11,547		

5.1.4 Comparación con el estado del arte

Aquí se presenta una comparación de los resultados obtenidos durante la experimentación con los trabajos que realizan clasificación de polaridad mencionados en el estado del arte. Además, en el reporte del TASS 2013 (Villena-Roman et al., 2013), el ganador de dicho taller reportó una precisión del 71 % al trabajar con el corpus general del TASS. La comparación se muestra en la tabla 5.7.

Tabla 5.7: Comparación con el estado del arte

Modelo	precision	recall	F1	Autor
SVC	0.78	0.76	0.76	(Arias et al., 2022)
SVM	0.75	0.75	0.75	(Cámara et al., 2011)
SVC	0.80	0.56	0.66	Nuestra propuesta

La comparación nos sirvió para determinar que nuestro modelo fue el que tuvo una mayor precisión en comparación con el estado del arte, no así para las métricas *recall* y *F1*. Cabe mencionar que estos trabajos se realizaron con otros corpus, aunque la tarea de clasificación fue la misma.

5.2 Rutina de obtención de hilos de Twitter

Una vez seleccionado el modelo de clasificación de polaridad, se implementó en la rutina de extracción de hilos de conversación de Twitter para agrupar los datos obtenidos del análisis y realizar el preprocesamiento descrito en el capítulo 4.

5.2.1 Agrupamiento y conteos de reacciones

Con los vectores de polaridad media por bloque de conversación, se realizó su agrupamiento para asignarles una etiqueta de clase, ya sea negativo, neutral o positivo. Para posteriormente realizar el conteo de respuestas por etiqueta y generar un nuevo vector de características para la etapa de clasificación de controversia. Un ejemplo de esto se observa en la figura 5.1.

Como podemos ver, hubo un total de 90 bloques de conversación. Esto quiere decir que se analizó la conversación principal y 89 tuits citados de los que se extrajo información. De los cuales, 78 fueron en promedio negativos, 10 fueron neutrales y 2 fueron positivos. Entonces, el vector de características generado para la etapa de clasificación de controversia es $[78, 10, 2]$.

5.2.2 Árboles de tuits citados

Además de la clasificación de polaridad y el preprocesamiento de los datos, la rutina de extracción de hilos de conversación de Twitter genera árboles de tuits citados, por el momento, estos árboles solo sirven para visualizar la profundidad de propagación de los tuits, pero en un futuro se contempla utilizarlos para identificar los nodos controversiales de la red. Por el tamaño, se muestra un ejemplo de los árboles de tuits citados generados para una noticia de la que se tomaron 4 bloques de conversación en la figura 5.2. Una explicación más detallada de esta figura es la siguiente. El nodo principal contiene el promedio del bloque de conversación del tuit original, y

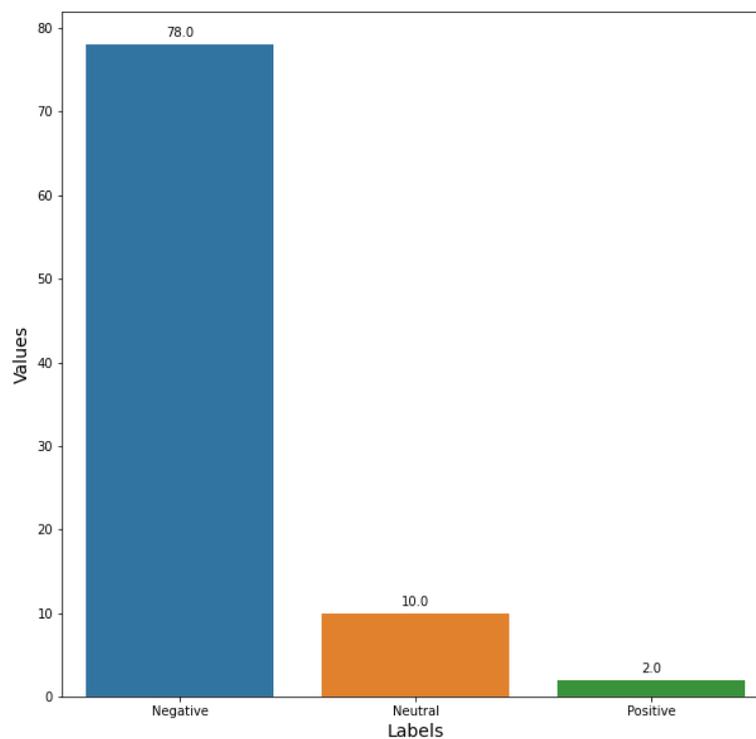


Figura 5.1: Conteos de respuestas por etiqueta de clase de los bloques de conversación analizados

el texto que contiene dicho tuit. Los nodos hijos muestran el id del tuit citado, y la polaridad media de ese bloque de conversación.

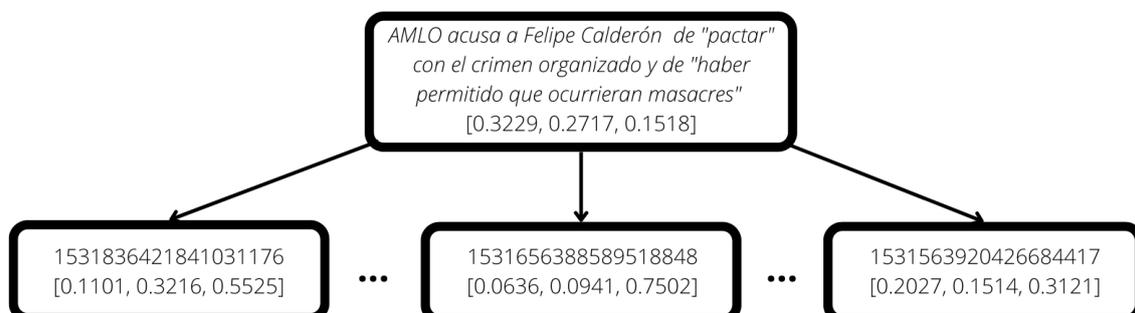


Figura 5.2: Ejemplo de un árbol de tuits citados generado con la rutina de extracción de hilos de Twitter

5.3 Modelo de controversia

Una vez obtenidos los vectores de características, se complementó el corpus de controversia, como se muestra en la tabla 5.8. En una primera fase de experimentación, se utilizó un modelo basado en reglas, en la fase posterior se implementaron árboles de decisión y un perceptrón multicapa (MLP) para realizar la tarea de clasificación.

Tabla 5.8: Ejemplo del corpus de controversia al ser complementado con los datos de los conteos de polaridad de los bloques de conversación

id del tuit	negativo	neutro	positivo	clase
1525877417675395072	10	1	5	0.0
1525870025676046338	2	3	3	0.0
1525706541013389313	54	6	6	1.0
1525687416488484864	0	0	0	0.0
1525645891465465856	8	32	23	1.0

5.3.1 Primera fase de experimentación

Configuración de los experimentos

En una primera fase de experimentación, se utilizó un modelo basado en reglas en el que si más del 90 % del total de bloques de conversación analizados se concentraba en la clase neutral o positiva, los tuits se clasificaban como no controversiales, en caso contrario los tuits se clasificaban como controversiales.

Resultados

En esta primera fase de experimentación, se utilizó el corpus de controversia completo para la evaluación del desempeño, ya que las reglas fueron deducidas manualmente al inspeccionar los datos. Los resultados obtenidos se muestran en la tabla 5.9 y su matriz de confusión en la figura 5.3.

De esta etapa de experimentación, se puede observar que los resultados son

Tabla 5.9: Resultados obtenidos en la primera etapa de experimentación, clasificación de controversia

	accuracy	precision	recall	F1
no controversial		0.59	0.57	0.58
controversial		0.64	0.66	0.65
promedio	0.62	0.61	0.61	0.61

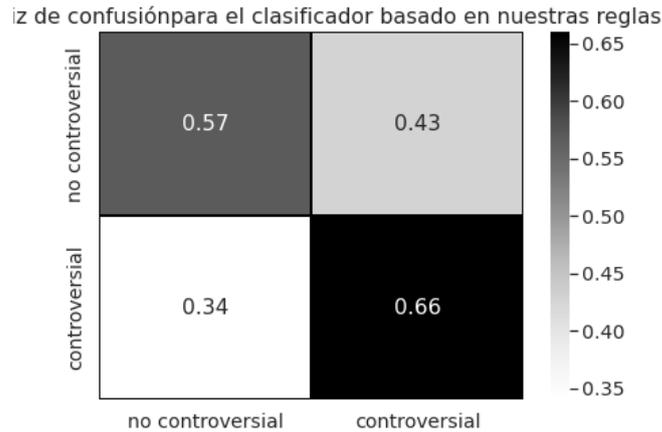


Figura 5.3: Matriz de confusión para los resultados de la primera fase de experimentación, clasificación de controversia

prometedores, aunque no son muy buenos, y a pesar de que las reglas fueron inferidas manualmente por nosotros y el tamaño del corpus de controversia es pequeño, este modelo basado en reglas sobrepasa el 50% en todas las métricas. Por lo que se puede decir que las muestras no están siendo clasificadas aleatoriamente. Los resultados de esta fase servirán como base para comparar con los modelos propuestos para la segunda fase de experimentación.

5.3.2 Segunda fase de experimentación

Modelos

Para la segunda fase de experimentación, se implementaron dos modelos de la librería Scikit-learn para Python. El primero de ellos fue el perceptrón multicapa *MLPClassifier*. El otro fue un árbol de decisión *DecisionTreeClassifier*. La configu-

ración utilizada para realizar estos experimentos se describe a continuación.

Configuración de los experimentos

En primer lugar, los conteos de polaridad se normalizaron al dividir cada vector entre el total de conteos del mismo, ya que no todas las noticias tenían 100 bloques de conversación. Posteriormente, el corpus se dividió en los conjuntos de entrenamiento y prueba utilizando la función *train_test_split*, para el conjunto de entrenamiento se seleccionó el 70% del total de los datos y para el corpus de prueba el 30% con la semilla 7. Los datos quedaron distribuidos de la siguiente forma. Para el conjunto de entrenamiento se tuvieron 20 datos pertenecientes a la clase no controversial y 22 pertenecientes a la clase controversial (42 en total). Para el corpus de prueba, 8 muestras de la clase no controversial y 10 de la clase controversial (18 en total). Los modelos se entrenaron con el conjunto de entrenamiento y su desempeño se evaluó con datos nuevos (el conjunto de prueba). Para entrenar el perceptrón multicapa se utilizó la estrategia *leave one out* para superar el tamaño del conjunto de datos y tuviera varias épocas de entrenamiento.

Resultados

Una vez finalizados los experimentos, se procedió a obtener sus resultados de evaluación. El perceptrón multicapa y el árbol de decisión obtuvieron los resultados mostrados en las tablas 5.10 y 5.11 respectivamente.

Tabla 5.10: Resultados para el perceptrón multicapa, clasificación de controversia

	accuracy	precision	recall	F1
no controversial		0.50	0.12	0.20
controversial		0.56	0.90	0.69
promedio	0.56	0.53	0.51	0.45

Las matrices de confusión de estos modelos se observan en la figura 5.4 para el perceptrón multicapa y en la figura 5.5 para el árbol de decisión.

Tabla 5.11: Resultados para el árbol de decisión, clasificación de controversia

	accuracy	precision	recall	F1
no controversial		0.70	0.88	0.78
controversial		0.88	0.70	0.78
promedio	0.78	0.79	0.79	0.78

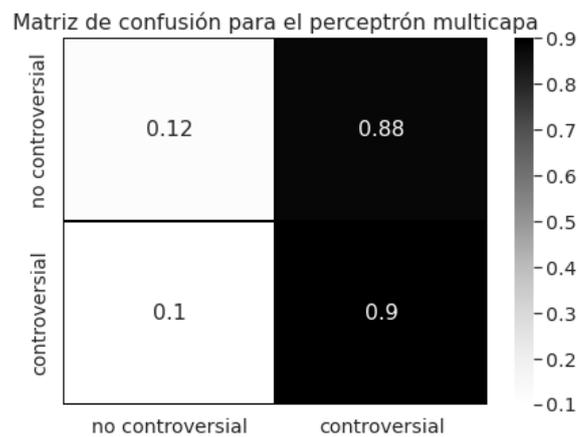


Figura 5.4: Matriz de confusión para perceptrón multicapa, clasificación de controversia

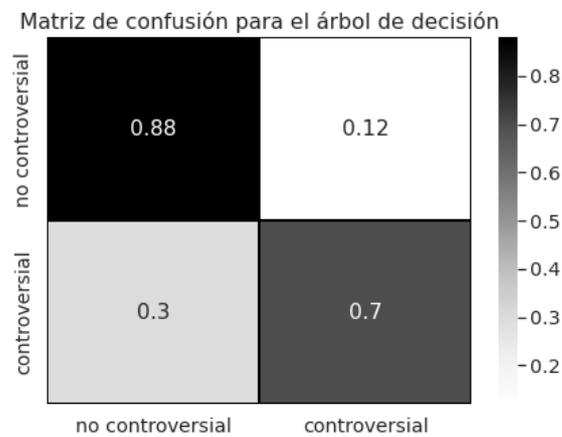


Figura 5.5: Matriz de confusión para el árbol de decisión, clasificación de controversia

Comparación de los modelos

Con los resultados de todos los experimentos, se realizó una comparación entre los modelos, tomando como base el modelo de clasificación propuesto por nosotros. Esta comparación se brinda en la tabla 5.12.

Tabla 5.12: Comparación de los modelos para clasificar controversia en noticias

Modelo	precision	recall	F1	accuracy
clasificador propuesto (base)	0.61	0.61	0.61	0.62
perceptrón multicapa	0.53	0.51	0.45	0.56
árbol de decisión	0.79	0.79	0.78	0.78

Al analizar esta tabla, nos podemos dar cuenta que el árbol de decisión supera en desempeño a la base propuesta por nosotros, en cambio, el perceptrón multicapa, tiene un desempeño menor en comparación con la base.

5.3.3 Discusión

De los resultados obtenidos en las etapas de la solución propuesta, podemos observar que en general nuestra propuesta base obtuvo resultados prometedores. El árbol de decisión superó a la base propuesta, lo que indica que nuestra solución propuesta tiene un gran campo de oportunidad para explorar y proponer nuevas ideas.

Del clasificador de polaridad, observamos que a pesar de que en el preprocesamiento no se utilizaron técnicas como la lematización o el *stemming*, o incluso técnicas avanzadas como el etiquetado gramatical, el modelo obtuvo buenos resultados. Este buen desempeño influyó en las etapas posteriores, ya que si hubiese obtenido mal desempeño, la clasificación de controversia se hubiera visto afectada en gran medida por la mala clasificación de polaridad que se realizó. El etiquetado de los datos en sí mismo fue un proceso controversial, ya que el grado de acuerdo entre anotadores fue muy bajo. A pesar de esto, El clasificador de controversia (árbol de decisión) generó buenas reglas para realizar la clasificación de las muestras correctamente. Mejorar estos resultados será problemático hasta que se desarrolle un conjunto de datos que contenga una gran cantidad de datos, y el etiquetado de estos se realice profesionalmente.

Capítulo 6

Conclusiones y trabajo futuro

6.1 Conclusiones

En este trabajo se propuso una nueva forma de realizar la clasificación de la controversialidad de las noticias, al utilizar la polaridad como características para entrenar los modelos que realicen la clasificación de controversialidad. Además, se desarrolló un corpus de controversia, que a pesar de no tener un buen grado de acuerdo entre anotadores, fue de gran utilidad para desarrollar una base que nos permitió fijar un punto de comparación para evaluar el desempeño de los modelos. En general, la propuesta aquí descrita y cada una de sus etapas tienen un buen desempeño para lograr el entrenamiento de un modelo de clasificación de controversialidad de las noticias y que además es capaz de predecir de buena forma la clase de pertenencia de datos nuevos. Los resultados obtenidos son prometedores, ya que dan pie a continuar con la investigación sobre esta propuesta. Una de las desventajas del corpus desarrollado es su tamaño, porque como se pudo apreciar el perceptrón multicapa no pudo aprender correctamente a identificar los patrones para efectuar la clasificación, entonces algoritmos de clasificación más robustos como los del enfoque del aprendizaje profundo tendrán malos desempeños.

Finalmente, la unión de las tres etapas de clasificación descritas en este trabajo y la forma en la que se unieron (polaridad, agrupamiento, controversia) resultaron ser buenas para lograr la tarea final de clasificación de la controversialidad de las noticias desde la perspectiva de sus lectores.

6.2 Trabajo futuro

Aún existe una gran oportunidad para mejorar la clasificación de controversia en cada una de las etapas de la solución. Para empezar, el clasificador de polaridad puede mejorarse, desde utilizar más métodos de preprocesamiento para asegurarnos de que las características que están siendo obtenidas sean las ideales. Emplear incrustaciones de palabras para vectorizar los textos, modificar los parámetros de los clasificadores para incrementar su desempeño. Incluso usar algoritmos de aprendizaje profundo como lo son los transformadores, ya que su desempeño fue muy bueno para realizar la clasificación, aunque no se tomó en cuenta en etapas posteriores porque hacer uso de transformadores no es el enfoque principal del trabajo. Probar con ensambles de modelos en el que cada uno de ellos se especialice en realizar la clasificación binaria de cada una de las clases.

La rutina de extracción de árboles de Twitter puede ser mejorada, quitando los promedios de polaridad por bloques de conversación, y almacenando cada uno de los vectores de polaridad del hilo de conversación de Twitter, para que a la hora de realizar los conteos la información sea lo más fiel posible a la realidad.

El corpus de controversia resultó ser de gran utilidad para establecer una base para comparar el desempeño de los modelos, pero tiene algunos puntos en contra, el tamaño del corpus, el índice de acuerdo entre anotadores es muy pequeño. Es por eso que se sugiere desarrollar un corpus más grande en el que el etiquetado se lleve a cabo de forma profesional, y sobre todo, que a la hora de desarrollar el corpus, las desventajas aquí planteadas sirvan de referencia para evitarlas.

En cuanto al clasificador de controversialidad, para poder probar con algoritmos más avanzados como las redes neuronales artificiales, es necesario que tengamos más muestras para el entrenamiento. Pero hay espacio para probar con otros algoritmos de inducción de reglas, en fin las oportunidades de investigación y experimentación una vez sentadas las bases son muy amplias.

Referencias

- Abreu Salas, J. I., Mirabal, P., & Ballester-Espinosa, A. (2021-09). Cascade of Biased Two-class Classifiers for Multi-class Sentiment Analysis.
- Ahuja, S., & Dubey, G. (2017). Clustering and sentiment analysis on Twitter data. *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*, 1-5. <https://doi.org/10.1109/TEL-NET.2017.8343568>
- Amigo, E., Carrillo de Albornoz, J., Chugur, I., Corujo, A., Gonzalo, J., Martin, T., Meij, E., de Rijke, M., & Spina, D. (2013). Overview of RepLab 2013: Evaluating Online Reputation Monitoring Systems. *Proceedings of the Fourth International Conference of the CLEF initiative*, 333-352.
- Arias, F., Guerra-Adames, A., Zambrano, M., Quintero-Guerra, E., & Tejedor-Flores, N. (2022). Analyzing Spanish-Language Public Sentiment in the Context of a Pandemic and Social Unrest: The Panama Case. *International Journal of Environmental Research and Public Health*, 19(16). <https://doi.org/10.3390/ijerph191610328>
- Bastick, Z. (2021). Would you notice if fake news changed your behavior? An experiment on the unconscious effects of disinformation. *Computers in Human Behavior*, 116, 106633. <https://doi.org/https://doi.org/10.1016/j.chb.2020.106633>
- Berger, A. L., Pietra, V. J. D., & Pietra, S. A. D. (1996). A Maximum Entropy Approach to Natural Language Processing. *Comput. Linguist.*, 22(1), 39-71.
- Bhowmick, A. K., Sai Bharath Chandra, G., Singh, Y., & Mitra, B. (2018). Constructing Influence Trees from Temporal Sequence of Retweets: An Analytical Approach. *2018 IEEE International Conference on Big Data (Big Data)*, 624-633. <https://doi.org/10.1109/BigData.2018.8622315>
- Callejas-Hernández, C., Rivadeneira-Pérez, E., Sánchez-Vega, F., López-Monroy, A. P., & Villatoro-Tello, E. (2022). The Winning Approach for the Recommendation Systems Shared Task@ REST_MEX 2022.
- Camara, E. M., Cumbreiras, M. A. G., Roman, J. V., & Morera, J. G. (2016). TASS 2015 – The Evolution of the Spanish Opinion Mining Systems. *Procesamiento del Lenguaje Natural*, 56(0), 33-40. <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/5284>
- Cámara, E. M., Valdivia, M. T. M., Ortega, J. M. P., & López, L. A. U. (2011). Técnicas de clasificación de opiniones aplicadas a un corpus en español. *Procesamiento del Lenguaje Natural*, 47(0), 163-170. <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/958>

- Cañete, J., Chaperon, G., Fuentes, R., Ho, J.-H., Kang, H., & Pérez, J. (2020). Spanish Pre-Trained BERT Model and Evaluation Data. *PML4DC at ICLR 2020*.
- Chowdhary, K. R. (2020). Natural Language Processing. En *Fundamentals of Artificial Intelligence* (pp. 603-649). Springer India. https://doi.org/10.1007/978-81-322-3972-7_19
- Coletta, L. F. S., da Silva, N. F. F., Hruschka, E. R., & Hruschka, E. R. (2014). Combining Classification and Clustering for Tweet Sentiment Analysis. *2014 Brazilian Conference on Intelligent Systems*, 210-215. <https://doi.org/10.1109/BRACIS.2014.46>
- Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3), 273-297. <https://doi.org/10.1023/A:1022627411411>
- Crank, J. (1963). Charles Babbage and his Calculating Engines. *Physics Bulletin*, 14(2), 42. <https://doi.org/10.1088/0031-9112/14/2/013>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
- Gambino, J. O. J. (2019). *Sentiment polarity prediction of Twitter users' opinions to national newspapers news* (Tesis doctoral). Centro de Investigación en Computación.
- Henriquez, C., Pla, F., Hurtado, L. F., & Guzman, J. (2017). Analisis de sentimientos a nivel de aspecto usando ontologias y aprendizaje automatico. *Procesamiento del Lenguaje Natural*, 59(0), 49-56. <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/5492>
- Hessel, J., & Lee, L. (2019). Something's Brewing! Early Prediction of Controversy-causing Posts from Discussion Features. <https://doi.org/10.48550/ARXIV.1904.07372>
- Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is Twitter, a Social Network or a News Media? *Proceedings of the 19th International Conference on World Wide Web*, 591-600. <https://doi.org/10.1145/1772690.1772751>
- Li, X., Peng, Q., Sun, Z., Chai, L., & Wang, Y. (2019). Predicting Social Emotions from Readers' Perspective. *IEEE Transactions on Affective Computing*, 10(2), 255-264. <https://doi.org/10.1109/TAFFC.2017.2695607>
- Martinez-Camara, E., Martin-Valdivia, M. T., Ureña-Lopez, L. A., & Mitkov, R. (2015). Polarity Classification for Spanish Tweets Using the COST Corpus. *J. Inf. Sci.*, 41(3), 263-272. <https://doi.org/10.1177/0165551514566564>
- Martin-Valdivia, M.-T., Martinez-Camara, E., Perea-Ortega, J.-M., & Ureña-Lopez, L. A. (2013). Sentiment polarity detection in Spanish reviews combining supervised and unsupervised approaches. *Expert Systems with Applications*, 40(10), 3934-3942. <https://doi.org/https://doi.org/10.1016/j.eswa.2012.12.084>
- Marvin, M., & Seymour, A. P. (1969). Perceptrons. *Cambridge, MA: MIT Press*, 6, 318-362.
- McCorduck, P., Minsky, M., Selfridge, O. G., & Simon, H. A. (1977). History of artificial intelligence. *IJCAI*, 951-954.

- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115-133. <https://doi.org/10.1007/BF02478259>
- Mejova, Y. (2009). Sentiment analysis: An overview. *University of Iowa, Computer Science Department*.
- Mejova, Y., Zhang, A. X., Diakopoulos, N., & Castillo, C. (2014). Controversy and Sentiment in Online News. <https://doi.org/10.48550/ARXIV.1409.8152>
- Navas-Loro, M., & Rodriguez-Doncel, V. (2020). Spanish corpora for sentiment analysis: a survey. *Language Resources and Evaluation*, 54(2), 303-340. <https://doi.org/10.1007/s10579-019-09470-8>
- Ochoa Luna, J., & Ari, D. (2018). Deep Neural Network Approaches for Spanish Sentiment Analysis of Short Texts. En G. R. Simari, E. Ferme, F. Gutierrez Segura & J. A. Rodriguez Melquiades (Eds.), *Advances in Artificial Intelligence – IBERAMIA 2018* (pp. 430-441). Springer International Publishing.
- Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. <https://doi.org/10.48550/ARXIV.CS/0205070>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
- Pennacchiotti, M., & Popescu, A.-M. (2010). Detecting controversies in Twitter: a first study. *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, 31-32. <https://aclanthology.org/W10-0516>
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6, 386-408.
- Roth, C., St-Onge, J., & Herms, K. (2022). Quoting is not Citing: Disentangling Affiliation and Interaction on Twitter. En B. R.M., C. C., C. H., M. E., R. L.M. & S.-P. M. (Eds.), *Complex Networks & Their Applications X. COMPLEX NETWORKS 2021*. Springer, Cham.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536. <https://doi.org/10.1038/323533a0>
- Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), 210-229. <https://doi.org/10.1147/rd.33.0210>
- Saralegi, X., & San Vicente, I. (2012). Tass: Detecting sentiments in spanish tweets. *Proceedings of the TASS Workshop at SEPLN*.
- Shamrat, F. M. J. M., Chakraborty, S., Imran, M., Muna, J. N., Billah, M. M., Das, P., & Rahman, M. O. (2021). Sentiment analysis on twitter tweets about COVID-19 vaccines using NLP and supervised KNN classification algorithm. *Indonesian Journal Of Electrical Engineering and Computer Science*, 23(1), 463-470.

- Turney, P. D. (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. <https://doi.org/10.48550/ARXIV.CS/0212032>
- Vásquez, J., Gómez-Adorno, H., & Bel-Enguix, G. (2021). Bert-based Approach for Sentiment Analysis of Spanish Reviews from TripAdvisor. *IberLEF@ SEPLN*, 165-170.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. <https://doi.org/10.48550/ARXIV.1706.03762>
- Villena-Roman, J., Lana-Serrano, S., Martinez-Camara, E., & Gonzalez-Cristobal, J. C. (2013). TASS - Workshop on Sentiment Analysis at SEPLN. *Procesamiento del Lenguaje Natural*, 50.
- Vizcarra, G., Mauricio, A., & Mauricio, L. (2018). A Deep Learning Approach for Sentiment Analysis in Spanish Tweets. En V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis & I. Maglogiannis (Eds.), *Artificial Neural Networks and Machine Learning – ICANN 2018* (pp. 622-629). Springer International Publishing.
- Wang, B., Sun, Y., Zhang, T., Sugi, T., & Wang, X. (2020). Bayesian classifier with multivariate distribution based on D-vine copula model for awake/drowsiness interpretation during power nap. *Biomedical Signal Processing and Control*, 56, 101686. <https://doi.org/https://doi.org/10.1016/j.bspc.2019.101686>