

FICHA RESUMEN

Marco de trabajo para análisis y clasificación de incidentes en una entidad financiera utilizando PLN. Ampliación del análisis de modelos de clasificación

1. **ÁREA DE TRABAJO:** PLN (Procesamiento de Lenguaje Natural)
2. **TIPO DE PROYECTO:** Aplicado - Innovación
3. **ESTUDIANTE(S):** Cristian Eduardo Reyes Cardenas
4. **CORREO ELECTRÓNICO:** cristianreyes@javerianacali.edu.co
5. **DIRECCIÓN Y TELEFONO:** Carrera 25 sur # 26 - 69 B/ Centenario
6. **DIRECTOR:** Mario Julián Mora cardona
7. **VINCULACIÓN DEL DIRECTOR:** Profesor de planta
8. **CORREO ELECTRÓNICO DEL DIRECTOR:** mariomora@javerianacali.edu.co
9. **CO-DIRECTOR:** N/A
10. **GRUPO O EMPRESA QUE LO AVALA:**
11. **OTROS GRUPOS O EMPRESAS:** N/A
12. **PALABRAS CLAVE:** Gestión de incidencias, memoria a corto-largo plazo, lenguaje de texto, base de datos, auditoria basada en datos, lenguaje natural, auditoria inteligente, procesamiento de lenguaje natural, machine learning.
13. **FECHA DE INICIO:** 15 MAYO 2022

14. DURACIÓN ESTIMADA: 18 MESES

15. RESUMEN:

Con base en el trabajo de grado: “Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando NPL”, se pretende explorar modelos que no fueron tenidos en cuenta dado que al hacer la evaluación de las métricas de calidad no alcanzaron el umbral esperado por el proyecto. Esto no significa que estos modelos no puedan ser evaluados con mayor profundidad para su implementación.

Dado que este proyecto académico está enfocado en el desarrollo de un prototipo funcional específico, se ha elegido una entidad financiera a la que llamaremos Banco, para proteger su confidencialidad. Esta propuesta de trabajo de grado de maestría tiene como objetivo satisfacer la necesidad de crear un prototipo para el análisis y clasificación de incidentes en dicha entidad financiera empleando técnicas de PLN (Procesamiento de Lenguaje Natural). En particular, el área de auditoría interna carece de comprensión de herramientas tecnológicas para el tratamiento de datos provenientes de reportes en lenguaje natural, aunando la falta de categorización y descripción acertada de incidentes, la dificultad para realizar búsquedas de incidentes en la descripción realizada por el creador de la novedad, ha provocado que se presente falta de certeza en la clasificación de los incidentes para cada uno de los procesos objeto de auditar, lo que ha impedido toma de decisiones basadas en datos, incumplimiento de los acuerdos de nivel de servicios (ANS), y cierre de incidencias sin respuesta acertada.



Marco de trabajo para análisis y clasificación de incidentes en una entidad financiera utilizando PLN. Ampliación del análisis de modelos de clasificación.

Cristian Eduardo Reyes Cárdenas
Código: 8972428

Proyecto Aplicado para optar al título de
Magister en Ciencia de Datos

Director:
Mario Julián Mora Cardona

FACULTAD DE INGENIERÍA Y CIENCIAS
MAESTRÍA EN CIENCIA DE DATOS
SANTIAGO DE CALI, OCTUBRE 1 DE 2025

TABLA DE CONTENIDO

INTRODUCCIÓN	9
1. DEFINICIÓN DEL PROBLEMA.....	12
1.1. PLANTEAMIENTO DEL PROBLEMA.....	12
1.2. FORMULACIÓN DEL PROBLEMA	13
2. OBJETIVOS DEL PROYECTO	15
2.1 OBJETIVO GENERAL	15
2.2 OBJETIVOS ESPECÍFICOS	15
2.3 RESULTADOS ESPERADOS.....	16
3. ALCANCE	18
4. JUSTIFICACIÓN	21
5. MARCO TEORICO DE REFERENCIA Y ANTECEDENTES.....	24
5.1 MARCO TEÓRICO	24
5.1.1 Definición de un marco de trabajo	24
5.1.2 Definición de Procesamiento de lenguaje natural.....	26
5.1.3 Marco de referencia de buenas prácticas COBIT 2019 aplicado a gestión de incidentes	27
5.1.4 Herramientas para el procesamiento del lenguaje natural	28
5.1.5 Técnicas del procesamiento del lenguaje natural.....	33

5.1.6 Exploración de los datos	38
5.1.7 Modelos supervisados	40
5.1.8 Redes neuronales artificiales	42
5.1.9 Arquitectura LSMT (Long-Short Term Memory)	42
5.1.11 Métricas de evaluación	46
Esta métrica es útil cuando se busca un equilibrio entre precisión y recall, especialmente en problemas donde los datos pueden estar desbalanceados. Un F1-score alto indica que el modelo logra una buena clasificación con un balance entre la cantidad de falsos positivos y falsos negativos.	48
5.1.12 Metodología CRISP-DM.....	48
5.2. ANTECEDENTES.....	51
5.2.1 Auditoria basada en datos	53
5.2.2 Transición de auditorias basadas en riesgos a datos	54
5.3. TRABAJOS RELACIONADOS.....	56
5.3.1 Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando PLN	56
5.3.2 Modelo de clasificación de incidentes tecnológicos desde un enfoque de aprendizaje automático en servicios de seguros	57
5.3.3 Diseño y Construcción de Modelo de Clasificación de Incidentes de Seguridad Usando NLP en los Registros de Texto Escrito para Automatizar Etiquetación	58
5.3.4 Método de procesamiento de lenguaje natural y técnicas de minería de datos aplicadas a la clasificación de incidentes informáticos.....	59
5.3.5 Técnicas de procesamiento de lenguaje natural en la inteligencia artificial conversacional textual	60
5.3.6 Método de procesamiento de lenguaje natural y técnicas de minería de datos aplicadas a la clasificación de incidentes informáticos.....	60

5.3.7 Diseño y construcción de modelo de clasificación de incidentes de seguridad usando NLP en los registros de texto escrito para automatizar etiquetación	61
6. METODOLOGÍA	62
6.1 Fases de desarrollo del proyecto	64
7. PRESENTACIÓN DEL PROYECTO	67
7.1 Comprensión de los datos	67
7.1.1 Descripción de las variables	67
7.1.2 Exploración de los datos	69
7.2 Preparación de los datos	70
7.2.1 Remover acentos	71
7.2.2 Colocar en minúsculas	71
7.2.3 Eliminar líneas sobrantes	71
7.2.4 Stemming	71
7.2.5 Insertar espacios entre caracteres especiales para aislarlos	72
7.2.6 Remover caracteres especiales y dígitos	73
7.2.7 Remover espacios en blanco extras	73
7.2.8 Tokenizar	73
7.2.9 Eliminación de Stopwords.....	74
7.3 MODELADO	75
7.3.1 Optimización de hiperparámetros	77
7.4 EVALUACIÓN.....	82

8. PROPUESTA DEL MARCO DE TRABAJO.....	88
9. RESULTADOS OBTENIDOS.....	95
9.1 Descripción general del enfoque y modelos evaluados.....	95
9.2 Resultados de los modelos implementados en este proyecto	96
9.3 Análisis y comparación los modelos del proyecto anterior.....	98
10. REFERENCIAS BIBLIOGRAFICAS	105

LISTA DE TABLAS Y FIGURAS

	Pág.
Tabla 1. Tipos de herramientas y lenguajes de programación empleado para PLN	31
Tabla 2. Resultados de los nuevos modelos implementado	96
Tabla 3. Resultados proyecto previo con técnica bolsa de palabras	97
Tabla 4. Resultados proyecto previo con técnica TF-IDF	98
Figura 1. División del conjunto de datos en entrenamiento y prueba	76
Figura 2. Categorías por división para datos de entrenamiento y prueba	77
Figura 3. Hiperparámetros usados en el modelo RNN	80
Figura 4. Hiperparámetros usados en el modelo CNN	80
Figura 5. Hiperparámetros usados en el modelo Red Neuronal Bidireccional	81
Figura 6. Evaluación del modelo CNN	83
Figura 7. Evaluación del modelo Neuronal Bidirreccional	84
Figura 8. Evaluación del modelo RNN	85
Ilustración 1. Esquema del concepto Stemming.	72
Ilustración 2. Ejemplo de tokenización.	74
Ilustración 3. Diagrama de eliminación de Stopwords.	75

INTRODUCCIÓN

Este proyecto complementará el trabajo realizado por los estudiantes de maestría Guiancarlo Velasco y Juan David Restrepo titulado “Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando NPL”. En dicho trabajo [1] se presentaron evaluaciones de modelos de clasificación para determinar los niveles de clasificación de incidencias en una entidad financiera específica y más puntualmente para la verificación del área de auditoría. Ese trabajo evaluó varios modelos de clasificación, optando luego por profundizar en Regresión Logística y Máquinas de Soporte Vectorial, según criterios de exactitud en el momento de compararlos con otros modelos. Esos otros modelos, según la teoría, también podrían proporcionar unos niveles positivos altos en la clasificación de este tipo de problemas y por lo tanto me motiva continuar dicho trabajo analizando con mayor profundidad el comportamiento de dichos modelos para este caso en particular.

La base de datos utilizada en este proyecto se mantiene consistente con el trabajo previo titulado “Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando NPL”. Esta continuidad en el uso de la misma base de datos facilita una comparación efectiva entre los modelos de clasificación previamente evaluados y los nuevos enfoques propuestos en este proyecto. Al aprovechar el conjunto de datos existente, este trabajo de grado busca no solo expandir y profundizar el análisis de modelos alternativos, sino también permitir una evaluación más precisa de su desempeño en el contexto específico de clasificación de incidentes en la entidad financiera.

Ya es conocido que en el área de auditoría interna de dicha entidad financiera, la falta de comprensión de herramientas tecnológicas hacia el tratamiento de datos mediante el PLN (Procesamiento de Lenguaje Natural), así como la incorrecta categorización y descripción de incidentes, y el desconocimiento de las técnicas de PLN aplicables al registro de incidentes, han generado un vacío en la certeza de la clasificación de los mismos.

Este proyecto se basa en datos que contienen registros de eventos reportados por los colaboradores del banco y clientes. Para el análisis de la información, se aplican técnicas de preprocesamiento y se seleccionan técnicas de clasificación, seguidas de una evaluación del desempeño de las mismas.

Se analizaron diferentes técnicas que buscan establecer una categorización adecuada de los eventos según su nivel de criticidad (Alto, medio o bajo). Esto permite al área de auditoría interna emitir recomendaciones a las áreas de operación correspondientes, para que puedan brindar una respuesta acorde al tipo de evento que se presente dentro de la organización.

De esta manera, se desarrolló un marco de trabajo para el análisis y categorización de incidentes en una entidad financiera, utilizando técnicas de PLN y modelos los de clasificación diferentes en los utilizados en el proyecto de referencia.

Es importante tener en cuenta que la elección de los modelos o técnicas de clasificación

dependen de la naturaleza de los datos, la disponibilidad de recursos computacionales, el tamaño del conjunto de datos y otros factores específicos que se presentaron en el desarrollo del proyecto.

1. DEFINICIÓN DEL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

El planteamiento del problema de este trabajo de grado se centra en los desafíos y limitaciones que enfrenta el área de auditoría interna de una entidad financiera en relación con la correcta clasificación de incidentes reportados por usuarios internos o clientes. Estos incidentes son reportados de manera escrita, en lenguaje natural, a través de una plataforma digital. Estos incidentes son clasificados manualmente por un analista de incidentes, lo que trae consigo errores humanos que dificultan el proceso.

En la actualidad las entidades financieras han visto la necesidad de pensar en estrategias [2] de transformación digital debido a la incorporación de nuevos competidores en el sector bancario, éstas vienen adoptando nueva tecnología para el mejoramiento de sus procesos internos, permitiendo generación de valor frente a sus clientes.

Pedrosa, Laureano, & Costa [3] resaltan que el empleo de la tecnología de información en auditoría cobra relevancia en organismos y entidades que regulan la profesión, el uso de estas herramientas garantiza eficiencia y efectividad en las actividades de auditoría. Las incidencias recibidas por la mesa de servicio, la cual tiene como función principal brindar soporte y responder de manera eficiente las solicitudes de servicio técnico que reportan los colaboradores, se comportan como datos de entrada o patrones de eventos, esta recepción y análisis puede convertirse en un proceso complejo y difícil de manejar

por parte de las entidades financieras, impidiendo el análisis e interpretación de los datos [4]. Para ello, se hace necesario el empleo de algoritmos, técnicas y mecanismos de verificación de minería de datos para su análisis e interpretación.

En el área de auditoría interna, debido al desconocimiento de herramientas tecnológicas para el tratamiento de los datos en el lenguaje natural, la falta de categorización y descripción acertada de incidentes, la ausencia de capacitación en formación especializada en temas relacionados en ciencias de datos, dificultad para realizar búsquedas de incidentes en la descripción realizada por el creador de la novedad y el desconocimiento de técnicas de lenguaje natural aplicable a registro de incidentes, ha provocado que se presente falta de certeza en la clasificación de los incidentes para cada uno de los procesos objeto de auditar en esta entidad, lo que ha impedido toma de decisiones basadas en datos, incumplimiento de acuerdos de nivel de servicios (ANS), cierre de incidencias sin respuesta acertada, poca credibilidad de los datos de consulta del registro de incidentes, además de posibles sanciones de entes regulatorios por no aplicar los controles de manera efectiva y se pueden presentar materialización de los incidentes, convirtiéndose en riesgo operativo para el Banco.

1.2.FORMULACIÓN DEL PROBLEMA

Responder los siguientes interrogantes: ¿Cómo mejorar la clasificación de los incidentes en una entidad financiera reportados por usuarios en lenguaje natural a partir de la línea

base establecida como resultado del trabajo de grado “Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando PLN”? ¿Cuáles son las técnicas de PLN adecuadas para el tratamiento de los datos recopilados de los incidentes reportados por los usuarios del banco? ¿Cómo evaluar el desempeño de las técnicas de clasificación utilizadas en términos de precisión, exhaustividad y eficiencia?

2. OBJETIVOS DEL PROYECTO

2.1 OBJETIVO GENERAL

Proponer un marco de trabajo para el análisis y clasificación de incidentes en una entidad financiera utilizando técnicas de PLN, para mejorar la efectividad de la auditoría interna y proporcionar recomendaciones acordes a la criticidad de los eventos.

2.2 OBJETIVOS ESPECÍFICOS

- Seleccionar las técnicas de Procesamiento del Lenguaje Natural (PLN) adecuadas para la representación de palabras en el contexto del tratamiento de datos de incidentes dentro de la entidad financiera.
- Comparar el rendimiento de diversos modelos de clasificación, incluyendo redes neuronales (CNN y RNN) y una Red Neuronal bidireccional, que no fueron considerados en el proyecto anterior "Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando PLN".
- Evaluar el desempeño de los modelos de clasificación, en términos de precisión, exhaustividad y eficiencia, profundizando sobre diferentes configuraciones de los modelos.

2.3 RESULTADOS ESPERADOS

Los resultados esperados de este trabajo de grado se centran en la mejora del proceso de clasificación de incidentes en entidades financieras ya que permite identificar el tipo de incidente y su nivel de criticidad. Una clasificación precisa de los incidentes es esencial para que la auditoría interna pueda tomar las medidas adecuadas para mitigar los riesgos y mejorar la seguridad de la entidad financiera. En este sentido, se espera encontrar un marco de trabajo que permita categorizar los incidentes de manera eficiente y precisa. Este marco de trabajo debe ser capaz de clasificar los incidentes según su tipo, nivel de criticidad y otros factores relevantes. También debe ser capaz de hacerlo de manera eficiente, para que la auditoría interna pueda dedicar su tiempo a otras tareas importantes. La validación de este marco de trabajo es esencial para garantizar que sea efectivo en la clasificación de incidentes en entidades financieras.

De acuerdo a lo anterior, los resultados de esta tesis son los siguientes:

- Marco de trabajo para categorizar los datos del registro de incidencias en una entidad financiera que involucre varios modelos de clasificación.
- Categorización, lo más precisa posible y adecuada, de los incidentes según su nivel de criticidad (Alto, medio o bajo) mediante el uso de técnicas de clasificación elegidas.
- Comparación frente al trabajo de grado “Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando NPL”.

- Recomendaciones basadas en la categorización de incidentes permitiendo a la auditoría interna emitir criterios más acertados y específicos a las áreas de operación correspondientes, en función de la categorización de los incidentes.
- Validación del marco de trabajo y los modelos implementados.

3. ALCANCE

El proyecto de investigación aplicada “Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando NLP” comprendió el desarrollo de un prototipo funcional para gestionar eficientemente los datos de registro de incidencias en entidad financiera como entrada para la planeación de la auditoría interna, empleó herramientas tecnológicas afín para el desarrollo del mismo, además se contó con el registro de clasificación automática de incidentes y un documento que determinó la clasificación de las herramientas tecnológicas adecuadas para el desarrollo del prototipo.

El proyecto también contó con un registro de clasificación automática de incidentes que posteriormente se usará en las auditorías internas para determinar las señales de alerta que se pueden presentar en la entidad a nivel operativo. El proyecto no involucró la adquisición de herramientas licenciadas para desarrollo del prototipo.

Para llevar a cabo este proyecto, se empleo la misma base de datos utilizada en el trabajo de grado previo, titulado "Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando NPL", desarrollado por los estudiantes de maestría Guiancarlo Velasco y Juan David Restrepo. Esta base de datos comprende un total de 14.287 registros de incidentes reportados por colaboradores internos y clientes de la entidad financiera. Cada registro incluye información detallada, como un identificador único (ID), la fecha de creación, el título, una descripción textual que explica la problemática, la prioridad asignada, el origen de la solicitud, el creador, la oficina, la

regional, la gerencia, la clasificación, el grupo de soporte, el responsable de la atención, el estado, los detalles de la respuesta, la fecha de implementación de la solución, el tiempo de resolución en minutos, la fecha de cierre, el tipo de ticket y si se cumplió o no con el acuerdo de nivel de servicio (SLA). Es importante destacar que el campo "Descripción" se convierte en un componente esencial de nuestro corpus, ya que contiene datos en lenguaje natural. La variable de respuesta que se utilizará en este estudio es la "Prioridad".

La aplicación de los modelos descartados en el trabajo de grado: "Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando NLP", permitió ampliar el análisis y clasificación de los incidentes en dicha entidad financiera. A continuación, se describe brevemente los modelos a implementados:

Se utilizaron Redes Neuronales Convolucionales (CNN) y Redes Neuronales Recurrentes (RNN) para procesar y clasificar los incidentes. Las CNN permitieron extraer características relevantes de los textos de los incidentes, mientras que las RNN capturarán relaciones de dependencia a largo plazo en el texto. Estos modelos se entrenaron utilizando un conjunto de datos etiquetados de incidentes para realizar la clasificación en diferentes categorías.

También se aplicó una red neuronal bidireccional, la cual usa una capa LSTM (Long Short-Term Memory) que procesa el texto en dos direcciones: de inicio a fin y de fin a

inicio. Esto permite que el modelo entienda mejor el contexto de cada palabra, ya que no solo toma en cuenta lo que viene antes, sino también lo que viene después. Este tipo de modelo es útil especialmente cuando el orden de las palabras y su relación dentro de la frase influye mucho en el significado. En el caso de los tickets analizados, se buscó que el modelo captara con mayor claridad la intención del texto y así mejorar la clasificación.

Se emplearon herramientas tecnológicas y software de licenciamiento libre, en la medida posible, se contó con la asesoría del director de trabajo de grado para el desarrollo del proyecto, de igual forma se plantearon propuestas de proyectos de grados futuros relacionados con la aplicación de lenguaje natural en otras áreas del Banco que cuentan con repositorios de bases de datos y pueden emplear la información para toma de decisiones y generar valor a la entidad financiera.

4. JUSTIFICACIÓN

En el área de auditoría interna de la entidad financiera viene presentando falta de certeza en la clasificación de los incidentes para cada uno de los procesos objetos a auditar. En el Banco se presentan un promedio de 15.000 registros de incidentes operativos mensuales, la descripción de los incidentes es realizado por cada uno de los coordinadores de servicio de las sedes bancarias en todo el país y por los colaboradores del Banco que perciben algún tipo de incidente al interior de la entidad.

La relación está en formato texto y se dificulta determinar a qué tipo de incidencia corresponde y su criticidad, dado que los registros lo realizan de acuerdo con su entendimiento. Lo que se buscó con este proyecto de grado fue definir un prototipo para análisis y clasificación de incidentes mediante el empleo de PLN sobre los registros de incidentes que están en formato texto y son almacenados en archivos Excel.

Se buscó de esta manera la categorización de los incidentes que se están presentando al interior del banco y poder clasificarlos, esta información es de vital importancia al momento de la planeación de las auditorías internas, puesto que estas señales de alerta, de no ser tratadas a tiempo pueden presentar materialización de los incidentes, convirtiéndose en riesgos operativos.

El proyecto de investigación aplicado cuenta con el aval de la entidad bancaria por parte

del área de auditoría interna, además suministran la base de datos como insumo para su análisis y tratamiento, también se cuenta con la autorización para el empleo de herramientas tecnológicas necesarias para el desarrollo de este proyecto aplicado. Las áreas auditadas del banco se verán beneficiadas con este proyecto, dado que desde el área de auditoría interna mediante los reportes que genere el prototipo se podrán dar a conocer de manera inmediata las señales de alerta a las áreas encargadas para su pronto tratamiento y evitar de esta manera la materialización del incidente.

En los modelos que fueron descartados inicialmente en el trabajo de grado "Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando NPL" se basa en la búsqueda de obtener los mejores resultados posibles en la clasificación de incidentes en una entidad financiera. Aunque no se proyectaron inicialmente en el trabajo, la literatura y el avance continuo en el campo de procesamiento de lenguaje natural (PLN) sugieren que estos modelos podrían ofrecer mejoras significativas en la precisión y efectividad del análisis y clasificación de los incidentes.

Es posible que los modelos descartados en el trabajo de grado tuvieran ciertas limitaciones o restricciones que llevaron a su exclusión en ese momento. Sin embargo, la literatura científica y los avances tecnológicos constantes en PLN sugieren que estos modelos podrían superar las limitaciones anteriores o tener características que los hacen más adecuados para el análisis y clasificación de los incidentes en una entidad financiera.

Al considerar los modelos previamente descartados, se brinda la oportunidad de aprovechar los conocimientos y técnicas más recientes en el campo de PLN para mejorar los resultados obtenidos en el trabajo de grado. Al ajustar los parámetros (o hiperparámetros) de estos modelos, se busca optimizar su rendimiento y adaptarlos específicamente al contexto de la entidad financiera y los datos disponibles.

El desarrollo de un marco de trabajo para la categorización eficiente de incidentes en una entidad financiera se sustenta en la necesidad de brindar una solución práctica a un desafío común en el ámbito de la auditoría interna. En muchas ocasiones, los usuarios finales de este marco, en su mayoría colaboradores de la entidad financiera, pueden carecer de experiencia en la selección y aplicación de modelos de clasificación específicos. Por lo tanto, es imperativo que el marco de trabajo propuesto no solo sea capaz de categorizar los incidentes de manera precisa, sino que también proporcione una guía paso a paso sobre cómo los usuarios de auditorías deben interpretar y utilizar los resultados de los modelos. Esta justificación se fundamenta en la idea de que la efectividad de la auditoría interna se maximiza cuando los usuarios finales pueden comprender y aplicar adecuadamente los resultados obtenidos, lo que, a su vez, conlleva a la toma de decisiones más informadas y eficaces en la entidad financiera.

5. MARCO TEORICO DE REFERENCIA Y ANTECEDENTES

5.1 MARCO TEÓRICO

A continuación, se relacionan los temas principales relacionados con el proyecto de grado correspondiente a “Marco de trabajo para análisis y clasificación de incidentes en una entidad financiera utilizando PLN. Ampliación del análisis de modelos de clasificación”, estos temas corresponden a la definición de un marco de trabajo y de PLN, metodologías, marcos de referencia y/o buenas prácticas relacionadas con gestión de incidentes, además de herramientas, bibliotecas y técnicas de procesamiento de lenguaje natural.

5.1.1 Definición de un marco de trabajo

Un marco de trabajo, también conocido como “framework”, es una estructura o conjunto de herramientas y pautas que proporciona una base sólida para el desarrollo y la implementación de un proyecto. En el contexto de este proyecto de grado, se puede considerar la implementación de un marco de trabajo para guiar y facilitar el desarrollo del prototipo de análisis y clasificación de incidentes en una entidad financiera utilizando técnicas de Procesamiento del Lenguaje Natural (PLN).

Un marco de trabajo generalmente incluye los siguientes elementos:

1. Bibliotecas y herramientas: Proporciona una selección de bibliotecas, herramientas y recursos necesarios para el desarrollo del proyecto. En el caso de este proyecto, podría

incluir bibliotecas de PLN, como NLTK, spaCy o Gensim, así como herramientas de procesamiento y análisis de datos.

2. Plantillas y estructuras: Ofrece plantillas, estructuras o esqueletos de código que pueden servir como punto de partida para el desarrollo del prototipo. Estas plantillas pueden incluir la carga de datos, la preparación de texto, la implementación de modelos de clasificación y la evaluación del desempeño.

3. Pautas y buenas prácticas: Establece pautas y buenas prácticas para el desarrollo del proyecto. Esto puede incluir recomendaciones sobre la gestión de datos, la selección de modelos, la validación cruzada, la optimización de hiperparámetros y la interpretación de resultados.

4. Documentación y tutoriales: Proporciona documentación detallada y tutoriales que explican los conceptos clave, las técnicas utilizadas y el uso del marco de trabajo. Esto ayuda a los desarrolladores a comprender y utilizar eficazmente las herramientas y funcionalidades disponibles.

Un marco de trabajo proporcionará varios beneficios en el contexto de este proyecto:

1. Eficiencia en el desarrollo: Al proporcionar una estructura y herramientas predefinidas, un marco de trabajo permite un desarrollo más rápido y eficiente del prototipo. Los componentes comunes y las funcionalidades preimplementadas pueden ser reutilizadas, evitando la necesidad de comenzar desde cero.

2. Consistencia y estandarización: Garantizará que el desarrollo del proyecto siga una estructura coherente y cumpla con las mejores prácticas.

3. Facilidad de mantenimiento: Al utilizar un marco de trabajo, el código se vuelve más organizado y estructurado, lo que facilita su mantenimiento a largo plazo. Además, las actualizaciones y mejoras del marco de trabajo pueden beneficiar directamente al proyecto sin requerir cambios significativos en el código existente.

5.1.2 Definición de Procesamiento de lenguaje natural

“El Procesamiento del Lenguaje Natural (PLN) es una subdisciplina de la inteligencia artificial y rama de la lingüística computacional que hace uso de un conjunto de mecanismos que la computadora asimila en un lenguaje de programación definido (formal) y le permite comunicarse con el ser humano en su propio lenguaje...” [5]. Este es un campo técnico que explora la relación entre la inteligencia artificial (IA) y el lenguaje natural que usamos normalmente. De esta manera, puede comprender los mecanismos que utilizan los humanos en relación con la comunicación y comprender qué quieren explicar las máquinas en el texto y el texto que envían [6]. Estos tipos de algoritmos se utilizan comúnmente, entre otras cosas, en la traducción de textos, resúmenes y reconocimiento de voz.

El registro de incidentes viene escrito en lenguaje natural, estos registros no están categorizados o tienen una primera categorización donde aproximadamente el 90% está

mal categorizado. Por lo tanto, se hace necesario utilizar las técnicas de NPL para deducir de los escritos las categorías.

5.1.3 Marco de referencia de buenas prácticas COBIT 2019 aplicado a gestión de incidentes

COBIT 2019 es un marco de referencia para el gobierno y la gestión de la información y la tecnología, COBIT define los componentes para crear y sostener un sistema de gobierno [7]. Los objetivos de gobierno y gestión de COBIT se agrupan en cinco dominios.

Los primeros se agrupan en el dominio evaluar, dirigir y monitorizar, mientras que los objetivos de gestión se agrupan en cuatro dominios APO (alinear, planificar e implementar), BAI (construir, adquirir, implementar), DSS (entregar, dar servicio y soporte), MEA (monitorizar, evaluar y valorar).

El objetivo de interés para el desarrollo del proyecto de grado corresponde al objetivo DSSO2 (Gestionar las peticiones y los incidentes de servicios), encargado de Proporcionar una respuesta oportuna y efectiva a las solicitudes de los usuarios y la resolución de todos los tipos de incidentes. Además de restaurar el servicio normal, registrar y completar las solicitudes de usuario; y registrar, investigar, diagnosticar, escalar y resolver los incidentes, lo que se busca con este proceso es Lograr una mayor productividad y minimizar las interrupciones mediante la resolución rápida de consultas e

incidencias de los usuarios [7].

5.1.4 Herramientas para el procesamiento del lenguaje natural

Actualmente existen muchas aplicaciones, herramientas, bibliotecas, librerías y programas que nos dotan de un entorno de trabajo que facilita el procesamiento del lenguaje natural. Estas herramientas implementan todo tipo de técnicas, métodos y algoritmos; y están disponibles comercialmente y de forma gratuita para diferentes lenguajes de programación bajo diferentes pautas de licencia.

Algunos de los enfoques actuales en el PLN están basados en técnicas de aprendizaje automático. Independientemente de la técnica utilizada para analizar los textos, la principal dificultad para el PLN reside en la ambigüedad del lenguaje natural. La resolución de ambigüedades no es una tarea trivial y requiere un procesamiento exhaustivo a todos los niveles (léxico, referencial, estructural y pragmático) [8]. La polisemia de las palabras, diferentes recursos literarios, acentos extranjeros, regionalismos, errores ortográficos y la intención o carga emocional de las sentencias (ironía, sarcasmo, etc.) [9], son algunos elementos a tener en cuenta a la hora de analizar cualquier texto. Algunos de ellos resultan difíciles de detectar incluso para las personas, lo cual complica aún más su formalización y posterior interpretación por parte de las máquinas.

La tabla número 1 resume el número de estas herramientas y bibliotecas y sus respectivos enlaces al sitio web oficial. La tabla [10] también muestra los lenguajes de programación que pueden usar.

Tabla 1. Tipos de herramientas computacionales para el PLN.

Lenguaje de programación	Nombre de la herramienta	Link de la herramienta	Licencia
JavaScript	Natural	https://github.com/NaturalNode/natural	Gratis
LISP	DEPLH-IN	https://github.com/delph-in/docs/wiki/	Gratis
Node JS	Natural	https://github.com/NaturalNode/natural	Gratis
Perl	Treex	https://ufal.mff.cuni.cz/treex	Paga
Python	MontyLingua	http://alumni.media.mit.edu/~hugo/montylingua/	Gratis
	Natural Language Toolkit (NLTK)	https://www.nltk.org/Home	Gracias
	Silpa Indic Language	https://smc.org.in/silpa	Gratis

	Processing Toolkit		
	Pytorch	https://pytorch.org	Gratis
	SpaCy	https://spacy.io	Gratis
	Gensim	https://radimrehurek.com/gensim/	Gratis
Scala	ScalaNLP	http://www.scalanlp.org	Gratis

Fuente: Elaboración propia.

Con base a la estructuración del proyecto, se realizó un enfoque en un lenguaje de código abierto y libre, donde la manipulación, limpieza, aplicación y análisis de los datos sea sin problemas de licencia, para ello se tomó como base librerías y paquetes del lenguaje de Python y con ello se tienen las siguientes alternativas de programas [10].

5.1.4.1 MontyLingua

MontyLingua [11] es un programa gratuito, enriquecido por el sentido común, para entender el lenguaje natural del inglés de principio a fin. Introduzca un texto en inglés sin procesar en MontyLingua y el resultado será una interpretación semántica de ese texto. Es perfecto para la recuperación y extracción de información, el procesamiento de solicitudes y la respuesta a preguntas. A partir de frases en inglés, extrae tuplas de sujeto/verbo/objeto, extrae adjetivos, frases sustantivas y verbales, y extrae nombres de personas, lugares, eventos, fechas y horas, y otra forma de información semántica.

5.1.4.2 Natural Language Toolkik (NLTK)

NLTK es una plataforma líder para crear programas en Python que trabajen con datos del lenguaje humano [12]. Ofrece interfaces fáciles de usar para más de 50 corpus y recursos léxicos como WordNet, junto con un conjunto de librerías de procesamiento de texto para la clasificación, la tokenización, el stemming, el etiquetado, el análisis sintáctico y el razonamiento semántico, envoltorios para librerías de PLN de uso industrial. NLTK tiene un conjunto de librerías de procesamiento del lenguaje natural, simbólico y estadístico, permite trabajar con corpus, categorizar el texto, analizar estructuras lingüísticas o reconocer las entidades nombradas [13]. Esta herramienta esta orientada a la investigación y la educación, las áreas que abarca NTLK son procesamiento de lenguaje natural, la lingüística empírica, las ciencias cognitivas, la inteligencia artificial, la recuperación de información y el aprendizaje automático [14].

5.1.4.3 Silpa Indic Language Processing Toolkit

Son un conjunto de herramientas de procesamiento del lenguaje natural son colecciones populares de software de procesamiento del lenguaje natural. Son conjuntos de bibliotecas, marcos y aplicaciones para el procesamiento simbólico, estadístico del lenguaje natural y del habla. Las herramientas de PLN suelen realizar la detección de frases, la tokenización, el etiquetado POS, la fragmentación del texto, la lematización, el análisis y la resolución de correferencias y la detección de entidades con nombre, entre otras cosas [15].

5.1.4.4 Pytorch

Es una librería de Python para Deep learning [16]; PyTorch es una biblioteca de aprendizaje automático de código abierto basada en la biblioteca de Torch, utilizado para aplicaciones que implementan cosas como visión artificial y procesamiento de lenguajes naturales, principalmente desarrollado por el Laboratorio de Investigación de Inteligencia Artificial de Facebook (FAIR). Es un software libre y de código abierto liberado bajo la Licencia Modificada de BSD.

5.1.4.5 SpaCy

SpaCy es una biblioteca de Python que puede usar para crear aplicaciones de procesamiento de lenguaje natural (PLN). SpaCy ofrece modelos pre-entrenados [17] en varios idiomas. Esto, junto con su sintaxis clara, es perfecto para principiantes en el campo de la PLN. Además, puede entrenar el modelo en un campo específico creando un nuevo modelo o volviendo a entrenar el modelo que proporcionó con sus propios datos. spaCy está diseñado para su uso en entornos de producción y, a diferencia de otras bibliotecas como TensorFlow, donde puede experimentar con diferentes arquitecturas de redes neuronales e implementar los últimos modelos de desarrollo, es completo y requiere procesamiento de lenguaje natural. Proporciona un marco para crear aplicaciones. Además, spaCy es eficiente en la CPU a pesar de usar un modelo de red neuronal. La velocidad y precisión del modelo spaCy es una de las mejores del mercado.

5.1.4.6 Gensim

Gensim [18] está diseñado para procesar texto digital no estructurado ("texto sin formato") utilizando algoritmos de aprendizaje automático no supervisados. Los algoritmos de Gensim como Word2Vec, FastText, Latent Semantic Indexes (LSI, LSA, LsiModel) y Latent Dirichlet Allocation (LDA, LdaModel) capturan la estructura semántica de un documento al encontrar patrones estadísticos de co-ocurrencia en el corpus automáticamente. Examine el documento de capacitación. Estos algoritmos no son monitoreados. Es decir, no se requiere intervención humana. Todo lo que necesita es un corpus de documentos de texto sin formato.

5.1.5 Técnicas del procesamiento del lenguaje natural

Bien sea en un área lingüística o computacional el procesamiento del lenguaje abarca modelos y técnicas orientados a la representación del conocimiento y razonamiento. Al igual se usan técnicas de estructuras de datos y algoritmos de búsqueda, además hay que comprender que el lenguaje natural se basa en análisis morfológico, sintáctico, semántico y pragmático del lenguaje [19].

Los sistemas de procesamiento del lenguaje natural se basan en técnicas de la lingüística computacional y la psicología cognitiva [19]. Como resultado, es posible implementar métodos a nivel sintáctico, semántico y pragmático que apoyen y justifiquen el uso de esta tecnología de IA y permitan la ejecución de procesos.

El aprendizaje automático basado en capacidades le da al sistema la capacidad de

"pensar" de manera inteligente, ya que se basa en teorías biológicas y psicológicas sobre el comportamiento del cerebro y cómo aprende [19].

Por lo tanto, el aprendizaje automático se basa en métodos tanto supervisados como no supervisados [20], donde el primero se basa en recibir el conjunto de entrenamiento, donde el sistema logra extraer características de los diferentes objetos o conjuntos de datos y conocer el resultado, luego puede determinar la clasificación a debe obtener un nuevo objeto para el sistema y, por el contrario, el aprendizaje no supervisado solo toma datos de entrada, pero puede relacionarlos, combinar las características de cada entrada para luego clasificarlos y poder tener subconjuntos que ayudan a distinguir nuevos datos.

5.1.5.1 Stemming

Es una técnica que busca reducir las diferentes palabras a su contexto raíz, es bastante útil en diferentes áreas de la computación lingüística, gracias a esta técnica utilizada comúnmente en el procesamiento de lenguaje natural [21] es posible reducir diferentes palabras que provienen de la misma raíz, tal sería el caso por ejemplo de las palabras: Biblioteca, Bibliotecario y Bibliotecóloga, el algoritmo de Stemming buscará encontrar la raíz de estas tres palabras para determinar que las tres pertenecen al conjunto de la "Bibliotec".

5.1.5.2 Tokenización

La tokenización es una técnica esencial en el procesamiento de lenguaje natural que

implica dividir el texto en unidades más pequeñas, conocidas como tokens. Estos tokens pueden ser palabras individuales o subunidades, como frases o sílabas, dependiendo del enfoque del procesamiento de texto [22]. La tokenización descompone el texto en unidades manejables que facilitan su análisis y comprensión. Por ejemplo, en la frase "Analicemos el procesamiento de lenguaje natural," la tokenización dividiría la frase en tokens individuales: "Analicemos," "el," "procesamiento," "de," "lenguaje," y "natural." Esta técnica resulta fundamental en la preparación de datos para su posterior análisis, ya que permite a los modelos de procesamiento de lenguaje natural trabajar con unidades discretas de texto, lo que facilita la extracción de características y la clasificación de incidentes de manera efectiva [22].

5.1.5.3 Stopwords

La eliminación de stopwords es una técnica fundamental en el procesamiento de lenguaje natural que se enfoca en la identificación y remoción de palabras comunes que aportan poco valor semántico a un texto, como "el," "la," "y," "de," "en," entre otros. Estas palabras, conocidas como stopwords, tienden a ser altamente frecuentes en texto, pero ofrecen información limitada para la clasificación de incidentes. La eliminación de stopwords es esencial en la etapa de preprocesamiento de datos, ya que permite reducir la dimensionalidad del texto y enfocarse en las palabras clave y significativas que realmente contribuyen a la categorización precisa de incidentes [23]. Por ejemplo, en la frase "El incidente ocurrió en la sala de reuniones," la eliminación de stopwords resultaría en la

reducción de "El" y "en," lo que permitiría un enfoque más efectivo en palabras clave como "incidente," "ocurrió," "sala," y "reuniones." Esta técnica mejora la calidad de los datos de entrada y contribuye a la eficacia de los modelos de clasificación en el procesamiento de incidentes.

5.1.5.4 Back Propagation Networks

La retropropagación [24] es la esencia del entrenamiento de las redes neuronales. Es el método de ajuste fino de los pesos de una red neuronal basado en la tasa de error obtenida en la época anterior (es decir, la iteración). Un ajuste adecuado de los pesos permite reducir las tasas de error y hacer que el modelo sea fiable aumentando su generalización.

La retropropagación en redes neuronales es una forma abreviada de "propagación de errores hacia atrás". Es un método estándar de entrenamiento de redes neuronales artificiales. Este método ayuda a calcular el gradiente de una función de pérdida con respecto a todos los pesos de la red [24].

5.1.5.5 Modelos de aprendizaje profundo

En esta parte se usó un modelo de aprendizaje profundo basado en redes neuronales, específicamente una red LSTM (Long Short-Term Memory) bidireccional. Este tipo de modelo es muy utilizado en tareas de procesamiento de lenguaje natural porque permite aprender patrones del texto en ambas direcciones, es decir, desde el comienzo hasta el final de la oración y viceversa. Gracias a esto, se puede entender mejor el contexto

general de lo que se está diciendo. En este proyecto, se aplicó este modelo para clasificar los reportes o incidentes que llegan a la entidad financiera, buscando que el sistema identifique mejor el tipo de problema reportado a partir del texto escrito por el usuario [35].

5.1.5.6 Redes Neuronales Convolucionales (CNN)

Las Redes Neuronales Convolucionales (CNN) serán implementadas como un modelo de clasificación de incidentes en este proyecto debido a su capacidad inherente para el procesamiento de datos secuenciales, como el texto natural presente en los incidentes reportados. Las CNN son especialmente adecuadas para extraer características relevantes de datos de entrada, lo que es esencial en la tarea de comprender y categorizar incidentes [24]. En este contexto, las capas convolucionales de la CNN actuarán como filtros que detectarán patrones y características específicas en el texto, como palabras clave y estructuras gramaticales relevantes para la clasificación. Posteriormente, las capas de agrupación (pooling) reducirán la dimensionalidad y resaltarán las características más importantes. Estas características extraídas se alimentarán a capas completamente conectadas que realizarán la clasificación final de los incidentes en categorías de criticidad, permitiendo así la toma de decisiones basadas en datos de manera precisa y eficiente en el contexto de la auditoría interna de la entidad financiera. La elección de las CNN como modelo se basa en su capacidad para capturar la información contextual y semántica en el texto, lo que es crucial para mejorar la precisión en la clasificación de incidentes [25].

5.1.5.7 Redes Neuronales Recurrentes (RNN)

El modelo de redes neuronales recurrentes (RNN, por sus siglas en inglés) es un tipo de arquitectura de redes neuronales artificiales que se utiliza para procesar secuencias de datos, como texto o series de tiempo. A diferencia de las redes neuronales convolucionales (CNN) que se utilizan principalmente para tareas de visión por computadora, las RNN están diseñadas para trabajar con datos secuenciales donde la información anterior es relevante para la información siguiente [26].

La característica distintiva de las RNN es la incorporación de conexiones recurrentes, lo que les permite mantener una especie de memoria interna. Esta memoria es la que les permite recordar y utilizar información previa en la secuencia para tomar decisiones o hacer predicciones sobre los datos actuales [26]. Cada uno de estos modelos de aprendizaje profundo se evaluará en función de su eficacia en la categorización de incidentes en la entidad financiera.

5.1.6 Exploración de los datos

La exploración de datos es una etapa fundamental en cualquier proyecto, ya que permite comprender la estructura, calidad y características del conjunto de datos con el que se trabajará [27]. En el contexto de este proyecto, la exploración de datos es crucial para evaluar la naturaleza de los incidentes reportados en la entidad financiera y determinar qué características pueden influir en su clasificación.

El conjunto de datos utilizado en este proyecto corresponde a los registros históricos de

incidentes reportados por usuarios internos y clientes a través de una plataforma digital.

Cada registro está compuesto por atributos que describen el incidente, tales como:

- Identificador del incidente: Código único asignado a cada reporte.
- Fecha y hora del reporte: Indica el momento en que el incidente fue registrado.
- Descripción del incidente: Texto libre de lenguaje natural que detalla el problema reportado.
- Categoría inicial del incidente: Clasificación asignada por el usuario o el sistema al momento del reporte.
- Estado del incidente: Indica si el incidente está abierto, en proceso o cerrado.
- Tiempo de resolución: Duración en horas o días desde el reporte hasta su cierre.
- Área responsable: Departamento encargado de atender el incidente.
- Nivel de criticidad: Indicador de la urgencia o severidad del incidente.

Para evaluar la calidad y utilidad del conjunto de datos, se llevarán a cabo diversas técnicas de exploración. En primer lugar, se realiza un análisis de valores faltantes y duplicados para identificar registros incompletos o repetidos que puedan afectar la precisión del modelo de clasificación. Posteriormente, se analiza la distribución de los incidentes con el fin de comprender la frecuencia de las diferentes categorías reportadas e identificar posibles sesgos en los datos. Adicionalmente, se examina la longitud y estructura de las descripciones de los incidentes, lo que permite detectar patrones lingüísticos, palabras clave y variaciones en la redacción que pueden influir en el desempeño de los modelos de Procesamiento de Lenguaje Natural (PLN). Asimismo, se

lleva a cabo un análisis de correlación entre variables, como la relación entre el nivel de criticidad del incidente y el tiempo de resolución, lo que proporcionó información clave para la mejora de la gestión de incidentes. Finalmente, se emplean técnicas de visualización, incluyendo gráficos y nubes de palabras, con el objetivo de identificar tendencias y patrones recurrentes en los incidentes reportados, facilitando así una mejor comprensión del comportamiento de los datos y su impacto en el sistema de clasificación [28].

Dado que los datos en este proyecto incluyen texto en lenguaje natural, la exploración inicial es fundamental para el preprocesamiento y la selección de técnicas de Procesamiento de Lenguaje Natural (PLN). Durante esta fase, se pueden identificar problemas como ruido en los datos (errores tipográficos, abreviaciones, caracteres especiales), lo que indica aplicar técnicas de limpieza y normalización antes de entrenar los modelos de clasificación [29].

5.1.7 Modelos supervisados

Los modelos supervisados de aprendizaje automático son un tipo de técnica de modelado en la que un algoritmo aprende a partir de un conjunto de datos etiquetado, es decir, datos de entrada que ya han sido clasificados con la respuesta correcta. En este enfoque, el modelo se entrena utilizando un conjunto de datos de entrenamiento compuesto por pares de entrada y salida esperada, con el objetivo de encontrar una función que relacione ambas variables y pueda generalizar el conocimiento a nuevos datos no vistos. Una vez entrenado, el modelo puede predecir la etiqueta o categoría de datos

desconocidos basándose en los patrones aprendidos [30].

El proceso de aprendizaje en los modelos supervisados consta de dos etapas principales: la fase de entrenamiento y la fase de prueba o validación. En la primera etapa, el modelo ajusta sus parámetros en función de los datos de entrenamiento, optimizando su desempeño a través de algoritmos de optimización como el descenso de gradiente. En la segunda etapa, el modelo se evalúa con un conjunto de datos distinto al utilizado en el entrenamiento para medir su capacidad de generalización y ajustar sus hiperparámetros en caso de ser necesario. El desempeño de estos modelos se mide mediante métricas como la precisión, la exhaustividad (recall), la puntuación F1 y la matriz de confusión, que permiten evaluar la capacidad del modelo para realizar predicciones correctas [30].

En el contexto de este proyecto, los modelos supervisados juegan un papel fundamental en la mejora de la clasificación de incidentes, ya que permiten automatizar el proceso de categorización a partir de datos históricos, reduciendo la carga de trabajo manual y minimizando errores humanos. La selección del modelo adecuado dependerá de factores como la naturaleza de los datos, la cantidad de registros disponibles y la complejidad del problema. En estudios previos, se ha demostrado que algoritmos como las Máquinas de Soporte Vectorial y la Regresión Logística ofrecen buenos resultados en tareas de clasificación de textos [30], mientras que enfoques más recientes basados en aprendizaje profundo, como las Redes Neuronales Recurrentes (RNN) y los Transformers, han

mostrado mejoras significativas en tareas de Procesamiento de Lenguaje Natural [31].

5.1.8 Redes neuronales artificiales

En este proyecto se trabajaron varios modelos basados en redes neuronales artificiales, que han demostrado ser bastante útiles para tareas de clasificación de texto. En primer lugar, se aplicó una CNN (red neuronal convolucional), que es un modelo que suele usarse en imágenes, pero que también sirve para encontrar patrones en texto, ya que puede detectar combinaciones de palabras que aparecen frecuentemente en ciertos tipos de incidentes. Luego, se probó una RNN (red neuronal recurrente), que es un modelo diseñado para trabajar con datos secuenciales, como el texto, porque tiene en cuenta el orden de las palabras. Finalmente, se usó una red neuronal bidireccional con LSTM, que mejora lo que hace la RNN al leer el texto tanto hacia adelante como hacia atrás, entendiendo mejor el contexto completo. Todos estos modelos se entrenaron con los reportes reales de la entidad financiera para clasificar los incidentes y ver cuál ofrecía mejores resultados [32].

5.1.9 Arquitectura LSMT (Long-Short Term Memory)

Dentro del abanico de modelos de Redes Neuronales Recurrentes (RNN), se encuentra una arquitectura más sofisticada conocida como LSTM (Long Short-Term Memory). Esta se distingue por su capacidad para mitigar el problema del desvanecimiento del gradiente, un desafío común en las RNN tradicionales al procesar secuencias largas. La arquitectura LSTM incorpora mecanismos de "puertas" que regulan el flujo de información

a través de la red. Estas puertas controlan qué información se mantiene en la memoria a largo plazo, qué información se olvida y qué nueva información se incorpora. Gracias a esta estructura, la LSTM puede aprender dependencias a largo plazo en los datos secuenciales, lo que la hace especialmente efectiva en tareas complejas como la traducción automática, el modelado del lenguaje y el análisis de series de tiempo extensas [33].

Entre los modelos de representación más utilizados en PLN se encuentran la Bolsa de Palabras (Bag of Words, BoW) y el TF-IDF (Term Frequency - Inverse Document Frequency), los cuales permiten representar documentos de texto en forma de vectores numéricos, facilitando el análisis y clasificación de incidentes [33].

5.1.10.1 Bolsa de palabras de PLN

El modelo de Bolsa de Palabras (Bag of Words, BoW) es una de las representaciones más simples y utilizadas en PLN. Su enfoque consiste en representar un documento de texto como un conjunto de palabras sin considerar su orden ni contexto gramatical. En este modelo, se construye un vocabulario único a partir de todos los documentos del corpus y se representa cada documento como un vector basado en la frecuencia de aparición de cada palabra en el texto [34].

Matemáticamente, si se tiene un conjunto de documentos, se crea una matriz en la que cada fila representa un documento y cada columna corresponde a una palabra del

vocabulario. Los valores dentro de la matriz indican la cantidad de veces que una palabra aparece en un documento específico. Este enfoque es particularmente útil en tareas de clasificación de texto, como la categorización de incidentes en entidades financieras, donde se requiere convertir reportes escritos en lenguaje natural en una estructura numérica que pueda ser interpretada por los modelos de clasificación [34].

Sin embargo, BoW presenta ciertas limitaciones. Dado que ignora el contexto y el significado de las palabras dentro de la oración, puede generar representaciones inexactas cuando se trabaja con textos complejos. Además, las palabras muy frecuentes pueden dominar la representación del documento, lo que puede afectar el rendimiento del modelo. Para mitigar estos problemas, se suelen aplicar técnicas como eliminación de palabras vacías (stopwords), normalización y reducción de dimensionalidad [34].

5.1.10.2 TF-IDF (Term Frequency – Inverse Document Frequency)

El modelo TF-IDF (Term Frequency - Inverse Document Frequency) es una mejora sobre BoW que permite ponderar las palabras dentro de un documento en función de su importancia relativa dentro de un corpus. En lugar de contar simplemente la frecuencia de aparición de una palabra, TF-IDF busca resaltar términos relevantes y reducir la influencia de palabras demasiado comunes en todos los documentos [35].

La métrica TF-IDF se calcula en dos partes:

1. Frecuencia de término (TF – Term Frequency): Indica la frecuencia con la que una palabra aparece en un documento en relación con el número total de palabras en ese

documento. Se define como:

$$TF(t, d) = \frac{f_{t,d}}{\sum_k f_{t,d}}$$

Donde, $f_{t,d}$ es el número de veces que el término t aparece en el documento d , y el denominador representa el número total de palabras en d .

2. Frecuencia inversa de documento (IDF – Inverse Document Frequency): Penaliza las palabras comunes en todo el corpus, asignando un peso menor a términos que aparecen con alta frecuencia en todos los documentos. Se define como:

$$IDF(t) = \log\left(\frac{N}{df_t} + 1\right)$$

Donde N es el número total de documentos y df_t es el número de documentos que contienen el término t .

El valor final de TF-IDF se obtiene multiplicando ambas métricas:

$$TF - IDF(t, d) = TF(t, d) \times IDF(t)$$

Este método es ampliamente utilizado en tareas de recuperación de información y clasificación de textos [36], ya que asigna mayor peso a palabras que son significativas dentro de un documento pero que no son excesivamente comunes en todo el corpus. En el contexto de esta tesis, TF-IDF es una técnica clave para representar los incidentes reportados y mejorar la precisión de los modelos de clasificación, permitiendo identificar términos relevantes que ayuden a diferenciar incidentes de alta y baja criticidad [37].

5.1.11 Métricas de evaluación

La evaluación de modelos de clasificación es un paso fundamental en el desarrollo de soluciones basadas en aprendizaje automático. En el contexto de este proyecto, donde se busca mejorar la clasificación de incidentes, es esencial contar con métricas adecuadas para medir el desempeño de los modelos supervisados. Entre las métricas más utilizadas en clasificación se encuentran la exactitud (accuracy), precisión (precision), exhaustividad (recall) y la métrica F1-score, cada una de las cuales proporciona información clave sobre la capacidad del modelo para clasificar correctamente los incidentes según su criticidad [38].

- Exactitud (Accuracy): La exactitud mide la proporción de predicciones correctas realizadas por el modelo sobre el total de predicciones. Se define matemáticamente como:

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

Donde,

TP (True Positives) son los incidentes correctamente clasificados como positivos.

TN (True Negatives) son los incidentes correctamente clasificados como negativos.

FP (False Positives) son los incidentes incorrectamente clasificados como positivos.

FN (False Negatives) son los incidentes incorrectamente clasificados como negativos.

Aunque el accuracy es una métrica intuitiva, puede ser engañosa en conjuntos de datos desbalanceados, donde una clase puede estar sobrerrepresentada. En estos casos, otras métricas como precision, recall y F1-score proporcionan una evaluación más robusta.

- Precisión (Precision): La precisión mide la proporción de predicciones positivas correctas en relación con todas las predicciones positivas realizadas por el modelo. Se calcula como:

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

Una alta precisión indica que el modelo tiene pocos falsos positivos, lo cual es crucial en aplicaciones donde es importante evitar clasificaciones incorrectas, como en la detección de incidentes críticos en auditoría interna.

- Exhaustividad (Recall o sensibilidad): La exhaustividad o recall mide la proporción de instancias positivas correctamente identificadas por el modelo, es decir, qué tan bien el modelo detecta todos los casos positivos. Se define como:

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

Un modelo con un recall alto tiene pocos falsos negativos, lo que significa que es capaz de identificar correctamente la mayoría de los incidentes críticos. Sin

embargo, un recall alto puede venir a costa de una menor precisión, lo que significa que algunos incidentes pueden ser clasificados erróneamente como críticos cuando no lo son [38].

- F1- Score: El F1-score es una métrica que combina precisión y recall en un único valor, calculado como la media armónica de ambas métricas:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Esta métrica es útil cuando se busca un equilibrio entre precisión y recall, especialmente en problemas donde los datos pueden estar desbalanceados. Un F1-score alto indica que el modelo logra una buena clasificación con un balance entre la cantidad de falsos positivos y falsos negativos.

5.1.12 Metodología CRISP-DM

La metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) es un estándar ampliamente utilizado en la minería de datos que proporciona un marco estructurado para el desarrollo de proyectos de análisis de datos. Su enfoque es flexible y adaptable, lo que permite su aplicación en diversos contextos, incluyendo la clasificación de incidentes en una entidad financiera mediante técnicas de Procesamiento de Lenguaje Natural (PLN) [39]. CRISP-DM se compone de seis fases principales, que permiten un flujo de trabajo iterativo y adaptable según las necesidades del proyecto. Estas fases son:

1. Comprensión del Negocio (Business Understanding): En esta fase se define el problema a resolver y se establecen los objetivos del proyecto en función de los requerimientos del negocio. En el contexto de este proyecto, el objetivo es mejorar la clasificación de incidentes en auditoría interna utilizando PLN para optimizar la detección de eventos críticos y reducir errores en la categorización manual. Se analizan los procesos actuales de la entidad financiera y se identifican los beneficios esperados de la implementación del modelo de clasificación [39].

2. Comprensión de los Datos (Data Understanding): En esta etapa se exploran los datos disponibles para evaluar su calidad, estructura y características. Se realiza un análisis exploratorio que incluye la identificación de valores faltantes, la distribución de categorías de incidentes y la identificación de patrones textuales en las descripciones de incidentes. Esta fase permite entender la viabilidad del uso de PLN en la clasificación de incidentes y detectar posibles desafíos en la limpieza y preprocesamiento de los datos [39].

3. Preparación de los Datos (Data Preparation): La preparación de los datos es crucial para garantizar la efectividad de los modelos de clasificación. En esta fase se aplican técnicas de limpieza y transformación de datos, incluyendo la eliminación de datos irrelevantes, la tokenización, la eliminación de stopwords, la lematización y la conversión del texto en vectores mediante técnicas como Bolsa de Palabras (BoW) y TF-IDF. Esta

fase también implica la división del conjunto de datos en subconjuntos de entrenamiento y prueba para la evaluación de los modelos [39].

4. Modelado (Modeling): En esta etapa se seleccionan y entrenan los modelos de clasificación supervisados y no supervisados. También se configuran hiperparámetros y se evalúan diversas representaciones del texto para determinar qué técnica ofrece mejores resultados en la clasificación de incidentes según su criticidad [39].

5. Evaluación (Evaluation): Una vez entrenados los modelos, se evalúa su desempeño utilizando métricas como accuracy, precision, recall y F1-score. Esta evaluación permite identificar qué modelo logra la mejor clasificación de incidentes y si cumple con los objetivos del negocio definidos en la primera fase. Se analizan los errores de clasificación y se determina si es necesario realizar ajustes en la selección de características o en la configuración de los modelos [39].

6. Despliegue (Deployment): La fase de despliegue consiste en la implementación del modelo en un entorno de producción para su uso real en la clasificación de incidentes. La aplicación del modelo se puede traducir en recomendaciones para su integración en los sistemas de auditoría interna de la entidad financiera. En un entorno real, esta fase incluiría la automatización del proceso de clasificación y la integración del modelo con herramientas de monitoreo y actualización continua [39].

5.2. ANTECEDENTES

La reciente crisis financiera no sólo puso de manifiesto las deficiencias en los procesos de gestión de riesgos, control y gobernanza de los bancos, sino que también puso de manifiesto la necesidad de mejorar la calidad de las auditorías de los bancos. Los auditores de los bancos pueden desempeñar un papel importante para contribuir a la estabilidad financiera cuando realizan auditorías bancarias de calidad que fomentan la confianza del mercado en los estados financieros de los bancos. Las auditorías bancarias de calidad son también una valiosa aportación al proceso de supervisión. El Comité de Supervisión Bancaria de Basilea (el Comité, o BCBS) [40] publica información sobre las auditorías de los bancos para mejorar la calidad de las auditorías de los bancos y aumentar la eficacia de la supervisión prudencial, que contribuyen a la estabilidad financiera.

El comité de auditoría de un banco desempeña un papel fundamental en el fomento de una auditoría bancaria de calidad mediante el ejercicio efectivo de sus responsabilidades con respecto al auditor y a la auditoría legal. Las directrices de este documento [41] promueven una comunicación bidireccional eficaz entre el comité de auditoría y el auditor para que el comité de auditoría pueda desempeñar sus responsabilidades de supervisión y contribuir a la eficacia del proceso de auditoría. Las directrices también constituyen la base para la evaluación por parte del supervisor de la vigilancia del comité de auditoría sobre la auditoría del banco.

Un auditor lleva a cabo la auditoría de los estados financieros de un banco para obtener una garantía razonable de que los estados financieros en su conjunto están libres de incorrecciones significativas, ya sea por fraude o por error, lo que le permite expresar una opinión sobre si los estados financieros están preparados, en todos sus aspectos significativos, de acuerdo con el marco de información financiera aplicable [42], e informar sobre los estados financieros, y comunicar lo que exigen las normas de auditoría internacionalmente aceptadas, de acuerdo con las conclusiones del auditor.

Es probable que la auditoría tenga mucha menos interacción entre humanos en relación con tareas muy repetitivas y basadas en reglas. Las herramientas de interfaz podrían utilizarse para compartir automáticamente la información en tiempo real con las herramientas de IA del auditor, que a su vez podría analizar, probar y señalar las anomalías o problemas que requieren la atención del auditor. De este modo, la interacción humana se centraría en las transacciones de alto riesgo y no en las consultas rutinarias [43]. En este caso, las herramientas de IA podrían identificar las transacciones inusuales y, al mismo tiempo, proporcionar información sobre las consideraciones pertinentes que el auditor podría tener en cuenta, incluidas las normas aplicables (normas de contabilidad, de divulgación, de auditoría o reglamentarias), situaciones históricas similares o resultados de fuentes disponibles públicamente (incluidas situaciones similares de grupos de pares del sector) [43].

La herramienta de IA también podría analizar las actas de las reuniones del consejo de administración o comunicaciones clave para ayudar al auditor a identificar riesgos adicionales y solicitar información de apoyo relevante, así como programar reuniones con las personas adecuadas para discutir asuntos de auditoría. Todo ello, además de ser capaz de procesar grandes cantidades de datos (como la lectura de extractos bancarios y contratos legales) y conciliar las cuentas muchas veces más rápido que un auditor humano y con menos errores [44].

5.2.1 Auditoria basada en datos

Las tecnologías avanzadas proporcionan una gran cantidad de información a un auditor que le permite emitir un juicio. Pero el auditor seguirá siendo quien emita ese juicio. La tecnología es un facilitador y no tiene rival cuando se trata de identificar correlaciones entre conjuntos de datos o variables. Sin embargo, se necesita la visión y la experiencia humanas para comprender en última instancia el contexto subyacente al resultado, así como la causalidad del resultado en relación con las entradas proporcionadas. Los resultados de la IA son, en el mejor de los casos, predicciones probabilísticas basadas en inferencias en la correlación de datos y no deben tomarse como verdades (es decir, las predicciones no son necesariamente la respuesta "correcta ") [45].

El auditor debe utilizar su criterio profesional para evaluar los resultados de la IA en combinación con otras pruebas. Las herramientas de IA pueden ofrecer otro nivel de conocimiento, pero no son la única respuesta.

Un auditor confirma la información y determina si se trata de una anomalía y, lo que es más importante, determina lo que implica o cómo concluir sobre lo apropiado del tratamiento de la información. En consecuencia, será aún más importante que los contadores públicos tengan conocimientos que vayan más allá de la experiencia en los estatutos de la contabilidad y la auditoría, para llegar a las bases fundamentales de la contabilidad y la auditoría, y de los procesos empresariales [46].

5.2.2 Transición de auditorías basadas en riesgos a datos

El uso de herramientas de IA también puede plantear cuestiones en torno al uso del muestreo. Por ejemplo, con respecto a los procedimientos de auditoría sustantiva, puede ser más eficiente utilizar el muestreo de auditoría si el auditor no puede diseñar un procedimiento con parámetros suficientemente precisos para un conjunto de datos para evitar la necesidad de gestionar el número resultante de valores atípicos, ya que el número de elementos notables o atípicos puede aumentar (a veces hasta los miles) cuando se analiza el 100% de la población [47].

En la actualidad, los informes de auditoría suelen publicarse tras el cierre del periodo de auditoría (normalmente entre 25 y 120 días después del cierre del periodo) [48]; sin embargo, los usuarios exigen cada vez más información puntual. Muchas partes de la auditoría ya pueden automatizarse o ejecutarse en paralelo y, por lo tanto, dar lugar a una finalización más rápida de los pasos individuales de la auditoría (a pesar de las

limitaciones de la IA identificadas anteriormente). La auditoría continua (por ejemplo, mensual, trimestral u otro plazo pertinente) o la información en tiempo real (por ejemplo, a medida que se producen las transacciones) pueden llegar a ser habituales.

La velocidad a la que el auditor puede emitir su opinión tras el cierre de un periodo está intrínsecamente limitada por la velocidad de información del cliente. El uso de herramientas de IA por parte del auditor puede dar lugar a que la herramienta de IA del auditor identifique continuamente las transacciones significativas a medida que se contabilizan (suponiendo que los controles estén diseñados y funcionen eficazmente) y realice automáticamente procedimientos para validar dichas transacciones (por ejemplo, relacionándolas con los detalles de las transacciones bancarias, evaluándolas con respecto a las condiciones contractuales), de modo que el auditor sólo tenga que evaluar si se requieren procedimientos adicionales en las afirmaciones que no pudieron comprobarse en tiempo real [49].

Este tipo de información continua y en tiempo real también puede ser relevante para los clientes que quieran implementarla como parte de su propia función de control o auditoría interna.

5.3. TRABAJOS RELACIONADOS

A continuación, se presentan trabajos relacionados al prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando PLN, estos trabajos abordan temas similares al mencionado anteriormente, de igual manera se relacionan trabajos que, aunque abordan problemas diferentes al interés del presente trabajo, mencionan técnicas, herramientas, metodologías para el tratamiento del problema actual. También se reportan trabajos que, aunque aborden problemas diferentes al de interés lo hacen usando técnicas o estrategias que podrían aplicar en el problema actual. Se presenta el trabajo, un resumen de como abordan la situación, la referencia o link donde se encuentra el detalle del trabajo, y un párrafo que indique de qué manera este trabajo aporta al desarrollo del proyecto actual y en qué es diferente de la propuesta que se está planteando.

5.3.1 Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando PLN

El trabajo se centra en el desarrollo de un sistema prototipo que utiliza procesamiento de lenguaje natural (PLN) para analizar y clasificar los incidentes en una entidad financiera. El objetivo principal del trabajo [50] es mejorar la eficiencia y la precisión en la gestión de incidentes en el contexto financiero, aprovechando las capacidades del procesamiento de lenguaje natural. El prototipo propuesto utiliza técnicas avanzadas de PLN, como modelos de aprendizaje automático y algoritmos de clasificación, para analizar el

contenido de los incidentes y clasificarlos en diferentes categorías.

Se realiza un análisis exhaustivo de los incidentes financieros y se identifican las características clave que permiten su clasificación. Se diseñan y desarrollan modelos de aprendizaje automático basados en PLN, como redes neuronales convolucionales (CNN), redes neuronales recurrentes (RNN) y modelos de atención, para procesar y clasificar los incidentes de manera automatizada.

El prototipo se desarrolla utilizando un conjunto de datos etiquetados de incidentes financieros, y se evalúa su desempeño mediante métricas de precisión, exhaustividad y F1-score. Además, se realizan ajustes y optimizaciones de parámetros para obtener los mejores resultados posibles.

El trabajo de grado concluye con una evaluación exhaustiva de los resultados obtenidos, destacando las fortalezas y limitaciones del prototipo propuesto. También se presentan recomendaciones para futuras investigaciones y mejoras en el análisis y clasificación de incidentes en entidades financieras utilizando PNL.

5.3.2 Modelo de clasificación de incidentes tecnológicos desde un enfoque de aprendizaje automático en servicios de seguros

El trabajo [51] aborda un problema de clasificación de incidentes tecnológicos por medio del análisis de los modelos de regresión logística, árboles de clasificación, bosque aleatorio, análisis lineal discriminante y máquinas de vectores de apoyo, los cuales son de aprendizaje automático.

Para esto, toma datos de una empresa de seguros colombiana con la intención de comparar entre sí, este grupo de modelos supervisados en distintos escenarios y así, identificar el de mayor exactitud en la clasificación de los incidentes.

Estos modelos le permiten a la compañía ahorrar recursos en la medida de que ya no es necesario que el personal use parte de su tiempo para realizar esta labor de forma manual, por otro lado, si se realiza esta clasificación con mayor exactitud el equipo encargado de atender el incidente puede recibir con prontitud la solicitud y puede ser atendida evitando reprocesos y con una mejor atención hacia el usuario final [51].

Empresas como las del sector asegurador y el bancario comparten ciertas similitudes con relación a las estrictas regulaciones que tienen para prestar un servicio más oportuno, los modelos estudiados en este trabajo pueden servir de referencia para construir la solución más apropiada para el problema planteado sobre la clasificación de incidentes para una entidad bancaria [51].

5.3.3 Diseño y Construcción de Modelo de Clasificación de Incidentes de Seguridad

Usando NLP en los Registros de Texto Escrito para Automatizar Etiquetación

En el sector minero chileno, el trabajo [52] aborda una problemática en materia de seguridad debido a las actividades del día a día. Esta actividad económica lleva consigo ciertos riesgos que, en caso de verse materializados, pueden llevar a la compañía a tener problemas en el aumento de los costos financieros en cuanto a que se deben asumir

indemnizaciones, días no laborados y reemplazo en el cargo a causa del ausentismo del trabajador calificado. Esto a su vez, puede llevar a costos de no calidad en el trabajo realizado si el reemplazo no es el indicado. Además, se tienen los costos relacionados a la reputación y al ambiente laboral de la empresa.

Las empresas del sector llevan a cabo el registro de unos OPS que son Observaciones Preventivas de Seguridad, en estos se especifica los comportamientos inusuales de ciertos trabajadores en temas de seguridad para tomar medidas preventivas. El gran inconveniente de estos registros es que la clasificación que se hace una vez se identifica la categoría del riesgo es manual, lo que hace del proceso una tarea muy compleja y poco eficiente [52].

5.3.4 Método de procesamiento de lenguaje natural y técnicas de minería de datos aplicadas a la clasificación de incidentes informáticos

En [13] se muestra un caso de estudio de una institución educativa en la que se cuenta con datos históricos recopilados por la mesa de servicios, relacionados con incidentes informáticos. Esta área se conecta con el resto de procesos de tecnología de la institución por lo que de ella depende la oportuna atención de los requerimientos de los empleados para realizar sus labores diarias.

En este caso, se hace uso de una metodología que aplica técnicas de procesamiento de lenguaje natural (PLN) para el entendimiento de los datos históricos, lo que le permite a

la mesa de apoyo identificar las variables que impactan con mayor frecuencia en los incidentes informáticos para su pertinente intervención [53].

El sector educativo y el bancario tienen bastantes diferencias, pero genera mucho valor tomar como referencia, técnicas probadas para solucionar la problemática enfocada al entendimiento de los datos en formato de texto con el objetivo de una toma de decisiones mucho más acertada [53].

5.3.5 Técnicas de procesamiento de lenguaje natural en la inteligencia artificial conversacional textual

Este trabajo resalta las técnicas de procesamiento de lenguaje natural en la inteligencia artificial conversacional textual, indican como la inteligencia artificial también impacta en la manera como se comunica el ser humano con la máquina, mediante los Chatbots, donde no solamente interpreta texto si no que son una especie de procesamiento lenguaje natural, el trabajo se relaciona con las técnicas que debemos emplear para la lectura de los textos consignados en los registros de incidentes del Banco [54].

5.3.6 Método de procesamiento de lenguaje natural y técnicas de minería de datos aplicadas a la clasificación de incidentes informáticos

El desarrollo de los diferentes dispositivos electrónicos que permiten el almacenamiento de gran cantidad de datos hace que el análisis de estos sea complicado. Para ello, se desarrolló la minería de datos, permitiendo la transformación de estos datos que ingresan

a la compañía en información útil a fin de poder analizar e identificar fallas. El análisis y la predicción de la asociación representan dos objetivos: la exploración de datos para la comprensión y la construcción de modelos para la clasificación [55]. Los modelos mencionados anteriormente tienen similitud con los temas que se van a tratar en el proyecto de grado aplicado en relación a la clasificación de incidentes.

5.3.7 Diseño y construcción de modelo de clasificación de incidentes de seguridad usando NLP en los registros de texto escrito para automatizar etiquetación

El tema tratado en este trabajo corresponde a registro de incidente laborales, sin embargo, mencionan como lograron entrenar tres algoritmos (Logit, Support Vector Machine y una red neuronal) para clasificar automáticamente los incidentes, los resultados muestran un buen desempeño, los cuales demuestran que es posible clasificar automáticamente los textos OPS. Los algoritmos mencionados anteriormente serán objeto de análisis para determinar si se ajustan a resolver el problema planteado en este proyecto [56].

6. METODOLOGÍA

La metodología de este proyecto se ha estructurado de manera coherente con los objetivos específicos planteados, con el propósito de alcanzar una solución efectiva para la clasificación de incidentes en una entidad financiera mediante técnicas de Procesamiento de Lenguaje Natural (PLN). A continuación, se describe la metodología detallada que se siguió para lograr cada uno de los objetivos específicos:

1. Selección de técnicas de PLN: La primera etapa consiste en una revisión exhaustiva de diversas técnicas de Procesamiento del Lenguaje Natural (PLN) para la representación de palabras.

a. Se llevó a cabo una revisión exhaustiva de las técnicas de representación de texto más adecuadas para el contexto financiero, considerando modelos preentrenados de PLN, como Word Embeddings (Word2Vec, GloVe) y modelos de lenguaje basados en transformers (BERT, GPT).

b. Se evaluaron las técnicas de representación mediante experimentos comparativos en términos de su capacidad para capturar la información semántica y contextual en los incidentes financieros.

2. Análisis de los modelos de clasificación: En esta segunda etapa se realizó un análisis detallado de varios modelos de clasificación enfocados en redes neuronales artificiales, los cuales no fueron explorados en profundidad en el proyecto anterior titulado "Prototipo para el análisis y clasificación de incidencias en una entidad financiera mediante PLN".

Específicamente, se trabajó con tres modelos principales: redes neuronales convolucionales (CNN), redes neuronales recurrentes (RNN) y una red neuronal bidireccional basada en LSTM. Estos modelos fueron seleccionados por su capacidad para aprender patrones complejos en secuencias de texto, permitiendo mejorar la clasificación automática de los reportes o incidentes en la entidad financiera.

a. Se llevó a cabo un análisis específico del rendimiento de estos modelos de redes neuronales, observando cómo cada uno procesa el texto y qué tan precisos son al momento de identificar la categoría del incidente reportado.

b. Se implementarán y entrenarán estos modelos utilizando conjuntos de datos de incidentes previamente etiquetados.

c. Se compararon los modelos en función de su precisión y eficiencia en la clasificación de incidentes, considerando diferentes configuraciones y hiperparámetros.

3. Evaluación de los modelos de clasificación: La etapa final consiste en evaluar el rendimiento de los modelos de clasificación aplicadas en términos de precisión, integridad y eficacia. Para ello, se llevarán a cabo estudios exhaustivos sobre distintas configuraciones de modelos.

a. Se evaluó el rendimiento de los modelos de clasificación, en particular de las CNN, utilizando métricas como precisión, exhaustividad y eficiencia.

b. Se llevó a cabo una evaluación exhaustiva de diferentes configuraciones de modelos, incluyendo variaciones en capas, funciones de activación y tamaños de filtros en las CNN.

c. Se identificarán las configuraciones óptimas que maximicen la precisión en la clasificación de incidentes.

La metodología propuesta permitió abordar de manera sistemática y coherente cada uno de los objetivos específicos de la investigación, asegurando así un enfoque riguroso y eficiente para el análisis y clasificación de incidentes en la entidad financiera mediante técnicas avanzadas de PLN.

6.1 FASES DEL DESARROLLO DEL PROYECTO

En el desarrollo del proyecto de grado se empleó metodología CRISP-DM, que fue considerada debido a que tiene un enfoque sistemático y estructurado para realizar proyectos de ciencias de datos.

Cabe resaltar que para el desarrollo de este trabajo no fue considerada la etapa de despliegue, dado que el alcance está definido solo hasta el desarrollo de prototipo y nuevos modelos de clasificación.

Ahora bien, de acuerdo a la metodología implementada se han realizado ciertas actividades de acuerdo a la fase, las actividades según la fase son:

1. Comprensión del Negocio (Business Understanding)

- Definición del problema
- Objetivos del proyecto

- Alcance
- Justificación
- Marco teórico de referencia y antecedentes
- Antecedentes
- Metodología
- Cronograma

2. Comprensión de los Datos (Data Understanding)

- Obtención de los datos (Registro de incidentes reportados en mesa de servicio de la entidad financiera)
- Descripción de la base de datos (Variables)
- Análisis exploratorio de los datos

3. Preparación de los Datos (Data Preparation)

- Remover acentos
- Colocar minúsculas
- Eliminar líneas sobrantes
- Stemming
- Insertar espacios entre caracteres especiales para aislarlos
- Remover caracteres especiales y dígitos
- Remover espacios en blanco extras
- Tokenización

- Eliminación de Stopwords

4. Modelado (Modeling)

- Dividir datos en datos de prueba y datos de entrenamiento.
- Configurar modelos de representación: bolsa de palabras y modelo de TF-IDF
- Aplicar modelos de clasificación con cada uno de los modelos de representación
- Cálculo de métricas de modelos
- Selección de mejores modelos
- Optimización de hiperparametros

5. Evaluación (Evaluation)

- Evaluación de las predicciones de los nuevos modelos aplicados
- Comparación de los modelos anteriores vs los nuevos aplicados

7. PRESENTACIÓN DEL PROYECTO

7.1 COMPRENSIÓN DE LOS DATOS

La entidad financiera facilitó una muestra en formato Excel de los tickets que se escalaron durante enero de 2022. Estos tuvieron a cabo un proceso previo de “anonimización” con el objetivo de mantener la confidencialidad de ciertos puntos claves. Para ese mes se escalaron 14.286 tickets, esta muestra resultó bastante significativa ya que permitió ejecutar sin problemas las fases posteriores.

Con relación a los tickets, estos son escalados a la mesa de servicios por parte de las distintas gerencias de la entidad a través de los distintos medios habilitados para tal fin, los más importantes para ese mes fueron el portal de reporte de tickets con 10.273 y los flujos de trabajo con 4005.

7.1.1 Descripción de las variables

Cada una de las 14.286 solicitudes o tickets del conjunto de datos trabajado cuenta con una serie de características diferentes, a saber:

- **ID:** un identificador único para la solicitud o ticket.
- **Fecha de Creación:** la fecha de creación de la solicitud o ticket.
- **Título:** título de la solicitud o ticket.
- **Descripción:** texto en el cual se explica la problemática que se le está presentando al colaborador de la entidad.

- **Prioridad:** clasificación de prioridad que se le da al ticket, puede ser: Bajo, Baja - Media, Medio, Alto y Urgente.
- **Origen:** aplicativo desde el cual llega la solicitud.
- **Creado Por:** identificador único del colaborador que crea la solicitud.
- **Oficina:** punto de venta, establecimiento o unidad empresarial de la entidad financiera.
- **Regional:** departamento donde se encuentra la oficina registrada en la solicitud.
- **Gerencia:** área de la entidad financiera donde trabaja el colaborador que realiza la solicitud.
- **Clasificación:** clasificación que los colaboradores de la mesa de servicio le asignan a las solicitudes con características similares.
- **Grupo de Soporte:** la solicitud se les escala a los distintos niveles de la mesa de servicios para encontrar la solución a la problemática.
- **Asignado A:** identificador único del colaborador que atiende la solicitud.
- **Estado:** al tratarse de solicitudes del mes de enero de 2022 ya todas se encuentran cerradas.
- **Detalle Respuesta:** texto en el cual se explica la solución a la problemática de la solicitud.
- **Fecha Solución:** la fecha en que se implementa la solución al problema.

- **Tiempo Solución Minutos:** tiempo transcurrido en minutos para implementar la solución al problema.
- **Fecha Cierre:** la fecha en que se cierra la solicitud o ticket.
- **Tipo Ticket:** todas están clasificadas como solicitudes.
- **Cumplió SLA:** se menciona si la solicitud esta vencida o no con respecto al tiempo de respuesta acordada por la mesa de servicio para las solicitudes.

Con estas variables se realizó un análisis exploratorio de datos que tenía como objetivo analizar su comportamiento y dar un valor añadido al trabajo al presentar datos valiosos que le permitieran a la entidad bancaria tomar futuras decisiones relacionadas a los hallazgos de la exploración.

7.1.2 Exploración de los datos

Se realizó un análisis exploratorio de datos para las distintas variables del conjunto de datos de tickets como valor agregado para la entidad financiera. Sin embargo, dando alcance a los objetivos definidos en el trabajo se continuó el proceso con dos variables: las descripciones de las distintas solicitudes y la prioridad asignada a los tickets.

De la variable “Prioridad” se identificó que había algunas categorías que contaban con muy pocos registros como, la de Baja – Media, que solo contaba con 8 registros y por lo tanto representaba el 0.1% del total de tickets. Otra prioridad que fue apareció con baja frecuencia fue la de Urgente que solo contaba con 75 apariciones lo que representaba el

0.5% del total de tickets. Al final para simplificar los pasos posteriores se tomó la decisión de unificar los tickets clasificados como Urgente con los de Alto y los tickets clasificados como Baja – Media con los de clasificación bajo.

Las razones principales para soportar estas modificaciones fueron las siguientes:

1. Las categorías con pocos registros tienen un bajo significado, esto hubiese llevado a que los modelos implementados en los pasos posteriores tuvieran demasiados problemas para clasificar correctamente estas categorías.
2. Recordemos que la muestra utilizada en el proyecto es de enero de 2022, en los siguientes meses la entidad financiera simplifico su forma de clasificar las prioridades de las solicitudes limitando el alcance a bajo y alto.

Con relación a la variable que almacenaba las descripciones de los distintos tickets de la muestra se analizó el top 100 de palabras que más se repetían. En este punto identificaron las palabras que, en nuestro idioma sirven de conectores entre palabras, como las que más se repiten. Esto se debía solucionar en la limpieza de los datos ya que contar con estas palabras no aportaban valor al proceso de clasificación de solicitudes y además podía opacar palabras que se repiten menos pero que son claves para entender porque un ticket se clasifica como Alto y no como Bajo.

7.2 PREPARACIÓN DE LOS DATOS

Una vez finalizada toda la fase de entendimiento de los datos se continuó con la preparación de datos, en esta fase se construyó una función que tomaba el corpus que

para el caso específico es la variable donde se almacenan las descripciones de las 14.286 solicitudes y se aplicaron distintos pasos que buscaban volverlas descripciones más comprensibles para los modelos. A continuación, se hace una breve descripción del paso a paso de la función de preprocesamiento y normalización del corpus.

7.2.1 Remover acentos

Este paso buscaba identificar en el corpus los caracteres o letras acentuadas. Posteriormente se usaba la librería “unicodedata” para transformarlos en caracteres ASCII o caracteres ingleses lo que ayudaba a normalizar las palabras del corpus.

7.2.2 Colocar en minúsculas

Para facilitar el manejo del texto lo más conveniente era convertir todos los caracteres en minúsculas.

7.2.3 Eliminar líneas sobrantes

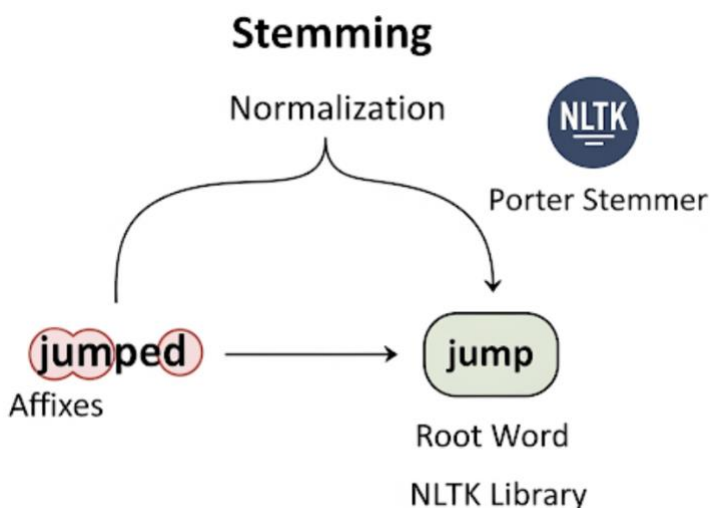
Utilizando expresiones regulares se realizó un proceso de eliminación de las líneas sobrantes entre cada una de las descripciones para de esta manera facilitar el posterior manejo de los datos sin que se diera algún error que se fuera difícil de identificar.

7.2.4 Stemming

Para entender el proceso de separación de palabras primero se debe entender que en español, los afijos son unidades como los prefijos y sufijos. Estos se unen a la raíz de una palabra para cambiar su significado o crear nuevas palabras. Este proceso es

conocido como inflexión. El paso de stemming nos ayuda a normalizar y estandarizar las palabras según su raíz independientemente de sus inflexiones.

Ilustración 1. Esquema del concepto: Stemming.



Fuente: Elaboración propia.

Para este proceso se utilizó uno de los stemmers más empleados, el Porter, este algoritmo fue desarrollado por Martin Porter y está disponible en la librería NLTK. Es de amplio uso porque soporta los idiomas de inglés y español.

7.2.5 Insertar espacios entre caracteres especiales para aislarlos

En este paso se llevó a cabo un proceso de aislamiento de los caracteres especiales y dígitos almacenados en las descripciones por medio de expresiones regulares. Este paso permitiría su posterior eliminación.

7.2.6 Remover caracteres especiales y dígitos

Por medio de las expresiones regulares se eliminaron los caracteres especiales y los dígitos que se encontraban en las descripciones de los distintos tickets. Este paso era de vital importancia ya que en las descripciones existían algunos datos personales de los colaboradores como la cédula o el número telefónico, así que se logró eliminar toda esta información privada junto con el resto de los caracteres no alfanuméricos que no aportaban a la clasificación y que por el contrario le sumaban ruido al texto.

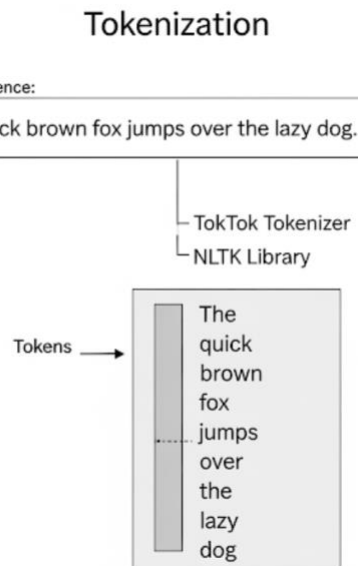
7.2.7 Remover espacios en blanco extras

Este paso se realizó por medio de expresiones regulares y buscaba eliminar espacios en blanco que no eran necesarios y que podían generar algún error de compilación del código de los modelos de clasificación.

7.2.8 Tokenizar

Tokenizar es un paso de vital importancia en la normalización del corpus, esta tiene como objetivo dividir o segmentar las frases en las palabras que la componen. La librería de NLTK ofrece varias opciones para este proceso, por lo que se decidió implementar el tokenizador TokTok que es uno de los últimos introducidos por NLTK y que asume que la entrada (corpus) tiene una frase por línea.

Ilustración 2. Ejemplo de la Tokenización.

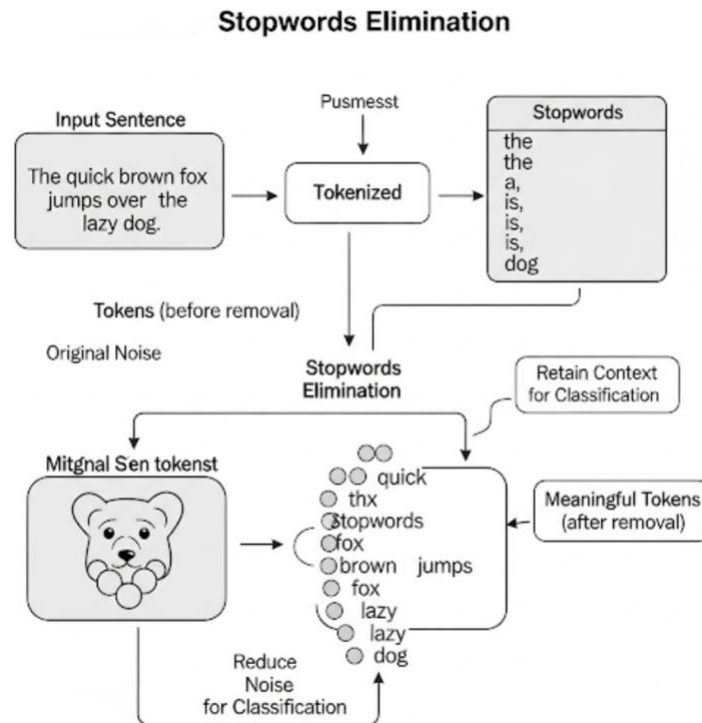


Fuente: Elaboración propia.

7.2.9 Eliminación de Stopwords

Las stopwords o palabras vacías son aquellas que tienen poco o ningún significado y suelen eliminarse del texto al procesarlo para que no generen ruido y se puedan conservar aquellas palabras que tienen el máximo significado y contexto para aportar a la clasificación.

Ilustración 3. Diagrama de eliminación de Stopwords



Fuente: Elaboración propia.

La eliminación de stopwords se realizó en simultaneo con el proceso de Tokenización ya que este permitía ir haciendo una identificación de su frecuencia de aparición basado en tokens singulares para proceder a filtrarlas del corpus.

7.3 MODELADO

Para esta etapa del proyecto ya se contaba con un corpus textual previamente normalizado, compuesto por las descripciones de los incidentes reportados en la entidad financiera. Estas descripciones habían pasado por un proceso de limpieza que incluyó la eliminación de tildes, signos de puntuación, palabras vacías (stopwords) y otros elementos que suelen generar ruido en el análisis de texto. Gracias a esto, el corpus

estaba preparado para alimentar los modelos sin que factores del lenguaje afectaran significativamente su rendimiento.

Con el corpus limpio y las etiquetas asociadas a cada descripción, se realizó la división del conjunto de datos en dos grupos: uno de entrenamiento y otro de prueba. Esto con el fin de entrenar los modelos con una parte de la información y luego evaluar qué tan bien podían predecir las categorías en textos que no habían visto antes. La división se hizo utilizando la función “train_test_split” de la librería “sklearn”, con una proporción del 67% para entrenamiento y 33% para prueba, y un “random_state=42” para asegurar la reproducibilidad del proceso. El código utilizado se presenta en la figura 1:

Figura 1. División del conjunto de datos en entrenamiento y prueba.

```
#Dividimos datos en subconjuntos de entrenamiento y prueba.
from sklearn.model_selection import train_test_split

train_corpus, test_corpus, train_label_names, test_label_names = \
    train_test_split(np.array(data_clean['Descripcion_limpia']),
                    np.array(data_clean['Category']),
                    test_size=0.33, random_state=42)

train_corpus.shape, test_corpus.shape

((9571,), (4715,))
```

Fuente: Elaboración propia.

El resultado fue un conjunto de 9.571 descripciones para entrenamiento y 4.715 descripciones para prueba, lo que permitió trabajar con una cantidad suficiente de datos para entrenar y evaluar los modelos de clasificación.

La distribución de las categorías en ambos conjuntos también fue tomada en cuenta, ya

que es importante que los modelos tengan ejemplos de todas las clases para poder aprender a diferenciarlas. Las tres categorías definidas fueron: Bajo, Medio y Alto, y la división por clase quedó como se observa en la figura 2:

Figura 2. Categorías por división para datos de entrenamiento y prueba.

```
#Mostramos la categoría por división
from collections import Counter

trd = dict(Counter(train_label_names))
tsd = dict(Counter(test_label_names))

(pd.DataFrame([[key, trd[key], tsd[key]] for key in trd],
              columns=['Category', 'Train Count', 'Test Count'])
 .sort_values(by=['Train Count', 'Test Count'],
              ascending=False))
```

	Category	Train Count	Test Count
1	medio	4317	2086
0	bajo	3919	1985
2	alto	1335	644

Fuente: Elaboración propia.

Como se puede ver, aunque la mayoría de los incidentes fueron clasificados como "Medio" o "Bajo", también se contó con una representación de incidentes de categoría "Alta", lo que permitió entrenar modelos que reconocieran las tres clases. Esta distribución no es completamente balanceada, pero fue suficiente para probar el rendimiento de los modelos de redes neuronales artificiales que se aplicaron más adelante: CNN, RNN y red neuronal bidireccional.

7.3.1 Optimización de hiperparámetros

Durante la construcción de los modelos de redes neuronales artificiales (RNN, CNN y red

neuronal bidireccional), se definieron una serie de hiperparámetros que fueron configurados con base en buenas prácticas documentadas en la literatura y en pruebas exploratorias. Estos hiperparámetros son valores que no se aprenden automáticamente durante el entrenamiento del modelo, sino que deben ser definidos previamente por quien diseña la red. Su correcta selección puede influir significativamente en el desempeño del modelo, ya que afectan tanto la capacidad de aprendizaje como la velocidad de convergencia y la generalización.

7.3.1.1 Hiperparámetros comunes en los tres modelos

A pesar de que se trata de arquitecturas distintas, los tres modelos compartieron una serie de hiperparámetros que ayudaron a mantener cierta uniformidad en el entrenamiento y facilitaron la comparación entre ellos:

- ✓ **max_words = 5000**: se estableció un tamaño de vocabulario limitado a las 5.000 palabras más frecuentes del corpus, con el fin de reducir la dimensionalidad y enfocarse en términos relevantes.
- ✓ **max_len = 100**: cada secuencia de texto fue limitada a un máximo de 100 tokens. Los textos más cortos se rellenaron con ceros y los más largos se truncaron, asegurando un tamaño uniforme en las entradas al modelo.
- ✓ **Embedding output_dim = 128**: se usó una dimensión de 128 para representar cada palabra en el espacio vectorial, permitiendo capturar relaciones semánticas de forma eficiente.

- ✓ **Dropout = 0.5:** se aplicó una tasa de abandono del 50% para reducir el riesgo de sobreajuste, desconectando aleatoriamente la mitad de las neuronas durante cada paso de entrenamiento.
- ✓ **Dense(64, activation='relu'):** se añadió una capa densa intermedia de 64 neuronas con activación ReLU, que actúa como una capa de transformación no lineal antes de la salida.
- ✓ **Capa de salida con softmax:** todos los modelos terminaron en una capa de salida con activación softmax, ya que el problema es de clasificación multiclase.
- ✓ **Función de pérdida = categorical_crossentropy:** se utilizó esta función por ser la más adecuada para problemas donde las etiquetas están codificadas como one-hot (una clase por muestra).
- ✓ **Optimizador = adam:** este optimizador se seleccionó por su eficiencia y buena capacidad de adaptación, combinando lo mejor de los métodos RMSprop y AdaGrad.
- ✓ **Métrica de evaluación = accuracy:** se usó la precisión global como métrica principal para monitorear el entrenamiento y la validación de los modelos.

7.3.1.2 Hiperparámetros específicos por modelo

Además de los parámetros comunes, cada arquitectura neuronal tuvo configuraciones particulares de acuerdo con sus características y el tipo de procesamiento que realiza sobre las secuencias de texto:

Figura 3. Hiperparámetros usados en el modelo RNN

#Paso 3: Definir la red RNN con LSTM

```
model = Sequential()
model.add(Embedding(input_dim=max_words, output_dim=128, input_length=max_len))
model.add(LSTM(64, return_sequences=False))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dense(y_train.shape[1], activation='softmax')) # salida multiclase

model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

Fuente: Elaboración propia.

En este modelo se utilizó una capa LSTM (Long Short-Term Memory) con 64 unidades. Esta capa está diseñada para procesar secuencias de datos donde el orden importa, manteniendo una "memoria" interna que ayuda a capturar dependencias a lo largo del texto. El parámetro `return_sequences=False` indica que solo se conserva el último estado oculto de la secuencia, lo cual es útil cuando se quiere hacer una predicción final sobre todo el texto y no sobre cada palabra. Esta estructura es adecuada para tareas donde se espera que el contexto anterior tenga peso en la clasificación.

Figura 4. Hiperparámetros usados en el modelo CNN

```
#Definimos el modelo CNN y su métrica
model = Sequential()
model.add(Embedding(input_dim=max_words, output_dim=128, input_length=max_len))
model.add(Conv1D(filters=128, kernel_size=5, activation='relu'))
model.add(GlobalMaxPooling1D())
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dense(y_train.shape[1], activation='softmax')) # salida multiclase

model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

Fuente: Elaboración propia.

En el caso del modelo CNN se incorporó una capa Conv1D con 128 filtros y un kernel de

tamaño 5. Esta capa permite capturar patrones locales en las secuencias de palabras, es decir, combinaciones frecuentes de términos que suelen aparecer juntas en ciertos tipos de incidentes. El tamaño del kernel define cuántas palabras se consideran en cada ventana de análisis. Esta arquitectura es menos sensible al orden global del texto, pero muy eficiente en la detección de estructuras repetitivas. Luego, se usó una capa de GlobalMaxPooling1D que selecciona la activación más fuerte de cada filtro, lo que simplifica la información y la hace más manejable para las capas densas posteriores.

Figura 5. Hiperparámetros usados en el modelo Red Neuronal Bidireccional

```
#Definimos el modelo BI-LSTM
from tensorflow.keras.layers import Bidirectional

model = Sequential()
model.add(Embedding(input_dim=max_words, output_dim=128, input_length=max_len))
model.add(Bidirectional(LSTM(64)))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dense(y_train.shape[1], activation='softmax')) # salida multiclase

model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

Fuente: Elaboración propia.

Este modelo está basado en una capa LSTM envuelta en una capa Bidirectional, también con 64 unidades. A diferencia del modelo RNN tradicional, esta red procesa cada texto en ambas direcciones: de izquierda a derecha y de derecha a izquierda. Esto permite que el modelo tenga una visión más completa del contexto de cada palabra dentro de la oración, lo que es especialmente útil cuando la información relevante no siempre está al inicio o al final del texto. La arquitectura bidireccional mejora la capacidad de entender relaciones más complejas entre palabras, aunque también implica un mayor costo

computacional y tiempo de entrenamiento.

7.4 EVALUACIÓN

Luego del proceso de entrenamiento de los modelos de redes neuronales, se procedió a evaluar su desempeño con el conjunto de datos de prueba, es decir, con descripciones de incidentes que los modelos no habían visto antes. La métrica principal utilizada para esta evaluación fue el accuracy, que indica el porcentaje de predicciones correctas realizadas por el modelo sobre el total de casos evaluados. Esta métrica es útil para tener una idea general de qué tan bien se comporta el modelo en términos de clasificación global.

Los resultados del accuracy para cada uno de los modelos implementados fueron los siguientes:

Figura 6. Evaluación del modelo CNN

```
#Entrenamos el modelo CNN
model.fit(X_train, y_train,
         epochs=10,
         batch_size=32,
         validation_data=(X_test, y_test))
```

Epoch 1/10
300/300 [=====] - 15s 44ms/step - loss: 0.7263 - accuracy: 0.6857 - val_loss: 0.5112 - val_accuracy: 0.7951
Epoch 2/10
300/300 [=====] - 12s 40ms/step - loss: 0.4102 - accuracy: 0.8408 - val_loss: 0.4852 - val_accuracy: 0.8085
Epoch 3/10
300/300 [=====] - 13s 44ms/step - loss: 0.2801 - accuracy: 0.8987 - val_loss: 0.5367 - val_accuracy: 0.8019
Epoch 4/10
300/300 [=====] - 16s 54ms/step - loss: 0.1951 - accuracy: 0.9342 - val_loss: 0.6185 - val_accuracy: 0.7911
Epoch 5/10
300/300 [=====] - 12s 41ms/step - loss: 0.1452 - accuracy: 0.9505 - val_loss: 0.6996 - val_accuracy: 0.7949
Epoch 6/10
300/300 [=====] - 17s 57ms/step - loss: 0.1071 - accuracy: 0.9660 - val_loss: 0.7998 - val_accuracy: 0.7934
Epoch 7/10
300/300 [=====] - 13s 42ms/step - loss: 0.1052 - accuracy: 0.9673 - val_loss: 0.8415 - val_accuracy: 0.7945
Epoch 8/10
300/300 [=====] - 13s 43ms/step - loss: 0.0843 - accuracy: 0.9737 - val_loss: 0.9455 - val_accuracy: 0.7915
Epoch 9/10
300/300 [=====] - 14s 47ms/step - loss: 0.0772 - accuracy: 0.9771 - val_loss: 1.0166 - val_accuracy: 0.7958
Epoch 10/10
300/300 [=====] - 13s 45ms/step - loss: 0.0716 - accuracy: 0.9777 - val_loss: 0.9563 - val_accuracy: 0.7934

Fuente: Elaboración propia.

Como se evidencia en los resultados de entrenamiento, el modelo CNN logró una evolución progresiva y consistente en la métrica de accuracy, alcanzando una precisión del 97.77% en los datos de entrenamiento al final de la décima época. Aunque el accuracy sobre el conjunto de validación fue inferior, se mantuvo cercano al 80%, lo que indica una buena capacidad del modelo para generalizar a datos no vistos.

Este desempeño puede atribuirse a la capacidad de las redes convolucionales (CNN) para identificar patrones locales en las secuencias de texto, lo que resulta especialmente

útil en tareas de clasificación de frases cortas o poco estructuradas, como es el caso de los reportes de incidentes en lenguaje natural. A lo largo del entrenamiento también se observó una reducción sostenida del valor de pérdida (loss), lo que indica una adecuada optimización del modelo sin señales evidentes de sobreajuste.

En resumen, el comportamiento del modelo CNN demuestra que este tipo de arquitectura es particularmente eficaz para extraer representaciones discriminativas del texto, incluso en contextos donde las descripciones son breves, informales o presentan variaciones lingüísticas.

Figura 7. Evaluación del modelo Neuronal Bidirreccional

```
#Entrenamos el modelo BI-LSTM
model.fit(X_train, y_train,
         epochs=10,
         batch_size=32,
         validation_data=(X_test, y_test))
```

```
Epoch 1/10
300/300 [=====] - 47s 156ms/step - loss: 0.4019 - accuracy: 0.8496 - val_loss: 0.5261 - va
l_accuracy: 0.8000
Epoch 2/10
300/300 [=====] - 40s 133ms/step - loss: 0.3119 - accuracy: 0.8859 - val_loss: 0.5421 - va
l_accuracy: 0.7877
Epoch 3/10
300/300 [=====] - 38s 128ms/step - loss: 0.2540 - accuracy: 0.9094 - val_loss: 0.6336 - va
l_accuracy: 0.7905
Epoch 4/10
300/300 [=====] - 33s 110ms/step - loss: 0.2049 - accuracy: 0.9285 - val_loss: 0.7013 - va
l_accuracy: 0.7862
Epoch 5/10
300/300 [=====] - 30s 101ms/step - loss: 0.1655 - accuracy: 0.9441 - val_loss: 0.7622 - va
l_accuracy: 0.7637
Epoch 6/10
300/300 [=====] - 45s 149ms/step - loss: 0.1381 - accuracy: 0.9524 - val_loss: 0.8588 - va
l_accuracy: 0.7695
Epoch 7/10
300/300 [=====] - 40s 133ms/step - loss: 0.1158 - accuracy: 0.9606 - val_loss: 0.9273 - va
l_accuracy: 0.7782
Epoch 8/10
300/300 [=====] - 47s 156ms/step - loss: 0.1052 - accuracy: 0.9646 - val_loss: 1.0329 - va
l_accuracy: 0.7745
Epoch 9/10
300/300 [=====] - 41s 138ms/step - loss: 0.0899 - accuracy: 0.9693 - val_loss: 1.1192 - va
l_accuracy: 0.7697
Epoch 10/10
300/300 [=====] - 31s 103ms/step - loss: 0.0852 - accuracy: 0.9730 - val_loss: 1.1183 - va
l_accuracy: 0.7792
```

Fuente: Elaboración propia.¹

Durante el entrenamiento del modelo Bidireccional LSTM (Bi-LSTM) se observó un comportamiento consistente en términos de aprendizaje, logrando alcanzar una precisión final del 97.30% en los datos de entrenamiento, según se muestra en la décima época. A pesar de este alto desempeño en el conjunto de entrenamiento, el accuracy en el conjunto de validación fue de aproximadamente 77.92%, lo cual indica que el modelo fue capaz de generalizar con un nivel de efectividad competitivo, aunque levemente inferior al modelo CNN.

El modelo demostró una disminución progresiva en la función de pérdida (loss) a lo largo de las épocas, lo cual refleja una adecuada optimización de los parámetros internos. No obstante, se evidenció un incremento gradual en la pérdida de validación (val_loss) a partir de la sexta época, lo que podría ser una señal incipiente de sobreajuste (overfitting), especialmente porque mientras el accuracy seguía mejorando en entrenamiento, el desempeño en validación comenzaba a degradarse ligeramente.

Este comportamiento es común en modelos complejos como los Bi-LSTM, ya que su arquitectura, al procesar las secuencias en ambas direcciones (de izquierda a derecha y

¹ Entrenamiento del modelo Bidireccional LSTM (Bi-LSTM) durante 10 épocas. Se observa un aumento progresivo en el accuracy del conjunto de entrenamiento, alcanzando un 97.30% al finalizar.

viceversa), tiende a captar patrones contextuales más profundos, pero también a memorizar detalles específicos del conjunto de entrenamiento si no se regula adecuadamente.

Figura 8. Evaluación del modelo RNN

```
#Entrenamiento
model.fit(X_train, y_train,
         epochs=10,
         batch_size=32,
         validation_data=(X_test, y_test))
```

```
Epoch 1/10
299/300 [=====>.] - ETA: 0s - loss: 1.0078 - accuracy: 0.4405
300/300 [=====] - 21s 60ms/step - loss: 1.0077 - accuracy: 0.4405 - val_loss: 1.0025 - val_accuracy: 0.4424
Epoch 2/10
300/300 [=====] - 16s 55ms/step - loss: 1.0032 - accuracy: 0.4475 - val_loss: 1.0060 - val_accuracy: 0.4424
Epoch 3/10
300/300 [=====] - 18s 59ms/step - loss: 1.0021 - accuracy: 0.4424 - val_loss: 0.9974 - val_accuracy: 0.4424
Epoch 4/10
300/300 [=====] - 18s 60ms/step - loss: 1.0018 - accuracy: 0.4475 - val_loss: 0.9988 - val_accuracy: 0.4424
Epoch 5/10
300/300 [=====] - 17s 56ms/step - loss: 1.0012 - accuracy: 0.4491 - val_loss: 0.9989 - val_accuracy: 0.4424
Epoch 6/10
300/300 [=====] - 15s 50ms/step - loss: 1.0012 - accuracy: 0.4507 - val_loss: 0.9970 - val_accuracy: 0.4424
Epoch 7/10
300/300 [=====] - 17s 58ms/step - loss: 1.0006 - accuracy: 0.4489 - val_loss: 0.9974 - val_accuracy: 0.4424
Epoch 8/10
300/300 [=====] - 17s 57ms/step - loss: 1.0001 - accuracy: 0.4447 - val_loss: 1.0002 - val_accuracy: 0.4424
Epoch 9/10
300/300 [=====] - 18s 60ms/step - loss: 1.0010 - accuracy: 0.4431 - val_loss: 0.9981 - val_accuracy: 0.4424
Epoch 10/10
300/300 [=====] - 18s 59ms/step - loss: 1.0011 - accuracy: 0.4511 - val_loss: 0.9974 - val_accuracy: 0.4424
```

Fuente: Elaboración propia.²

² Resultados del entrenamiento del modelo RNN simple durante 10 épocas. Se observa un comportamiento plano en la métrica de accuracy (aproximadamente 0.4424), sin mejoras significativas a lo largo del proceso.

En el caso del modelo RNN simple, los resultados muestran un rendimiento significativamente inferior en comparación con las otras arquitecturas implementadas. Como se aprecia en el registro de entrenamiento, el modelo se estancó con un accuracy constante de aproximadamente 0.4424 durante todas las épocas, tanto en los datos de entrenamiento como en validación. Este comportamiento indica que el modelo no logró aprender patrones discriminativos útiles a partir del corpus procesado.

Además, los valores de pérdida (loss) se mantuvieron alrededor de 1.00, sin una tendencia clara de mejora, lo cual refuerza la idea de que el modelo no logró optimizarse de manera efectiva. Esto puede deberse a varias razones. Por un lado, los reportes de incidentes utilizados como entrada suelen ser textos breves y poco estructurados, lo que limita la capacidad de las RNN de capturar dependencias a largo plazo una de sus principales fortalezas. Por otro lado, la arquitectura RNN simple, al no estar combinada con mecanismos de memoria más robustos como LSTM o GRU, puede ser insuficiente para problemas donde el contenido semántico está distribuido de forma irregular o es escaso.

A nivel computacional, el modelo RNN no mostró ventajas significativas en tiempos de entrenamiento respecto a las demás redes, pero sí una clara desventaja en cuanto a su desempeño. Esto resalta la importancia de considerar tanto la naturaleza del texto como la complejidad del modelo a implementar.

8. PROPUESTA DEL MARCO DE TRABAJO

El principal propósito de este proyecto es la construcción de un marco de trabajo sólido, práctico y replicable para la clasificación de incidentes en una entidad financiera, utilizando técnicas de Procesamiento de Lenguaje Natural (PLN) combinadas con modelos de aprendizaje automático y aprendizaje profundo. Este marco busca consolidar los avances del proyecto anterior centrado en modelos clásicos supervisados y complementarlo con nuevas herramientas, modelos más avanzados y una estructura que facilite su implementación en otros contextos similares.

La creación de este marco de trabajo parte del análisis detallado de los modelos implementados, los resultados obtenidos en la clasificación de textos, y la necesidad de contar con una estructura clara que sirva como guía tanto para proyectos futuros como para su posible aplicación real dentro de entornos organizacionales. A continuación, se presentan los componentes clave que conforman esta propuesta.

1. Bibliotecas y herramientas: Uno de los primeros pasos para establecer un marco de trabajo funcional es definir qué herramientas y bibliotecas son necesarias para su desarrollo e implementación. Este proyecto utilizó una combinación de herramientas ampliamente adoptadas en el análisis de texto y aprendizaje automático:

- ✓ Bibliotecas de PLN

- *NLTK*: Útil para tareas de limpieza básica, tokenización y eliminación de stopwords.
 - *spaCy*: para tareas más complejas de procesamiento como lematización o reconocimiento de entidades.
 - *re*: para limpieza con expresiones regulares.
- ✓ Preprocesamiento y análisis de datos
- *Pandas, numpy, matplotlib, seaborn*: para manipulación y visualización de datos.
- ✓ Modelado y evaluación
- *scikit-learn*: para la división del conjunto de datos, evaluación de modelos, métricas y comparación con modelos clásicos.
 - *tensorflow.keras*: para la construcción, entrenamiento y evaluación de modelos profundos como RNN, CNN y redes bidireccionales.
- ✓ Codificación de etiquetas
- *LabelEncoder* y *to_categorical*: para transformar las etiquetas de texto en formato numérico adecuado para modelos de clasificación multiclase.

Este conjunto de herramientas proporciona una base estable, gratuita y fácil de implementar en diferentes entornos, permitiendo adaptar el marco según los recursos disponibles o la complejidad del problema a resolver.

2. Plantillas y estructuras: Como parte de este marco, se propone una estructura general que puede servir como plantilla para el desarrollo de prototipos de clasificación

de texto:

✓ Carga y exploración de datos

- Lectura de bases de datos.
- Identificación de variables clave.
- Análisis exploratorio inicial de frecuencia de clases y longitud de textos.

✓ Preprocesamiento y normalización

- Eliminación de tildes, signos y caracteres especiales.
- Conversión a minúsculas.
- Eliminación de stopwords y palabras poco frecuentes.
- Tokenización, lematización o stemming, según el caso.

✓ Codificación y representación del texto

- Conversión de texto a secuencias numéricas mediante técnicas como Bag of Words, TF-IDF o embeddings.
- Ajuste de secuencias a una longitud estándar (padding).

✓ División del conjunto de datos

- Separación en conjuntos de entrenamiento y prueba (por defecto, 67%/33%).
- Mantenimiento de la proporción de clases para evitar sesgos.

✓ Modelamiento

- Selección del modelo a aplicar (CNN, RNN, Bidirectional LSTM, o modelos clásicos).
- Configuración de hiperparámetros clave: número de neuronas, tamaño del vocabulario, longitud de secuencias, tasa de dropout, etc.

✓ Evaluación

- Cálculo de métricas estándar: Accuracy, Precision, Recall y F1-Score.
- Matriz de confusión y visualización de errores comunes.

✓ Comparación y análisis

- Evaluación cruzada entre modelos para seleccionar el de mejor desempeño.
- Reflexión sobre la interpretabilidad, rendimiento y aplicabilidad en entornos reales.

3. Pautas y buenas prácticas: Este marco también incorpora una serie de recomendaciones prácticas y lineamientos para asegurar que los desarrollos sean sostenibles, replicables y escalables:

- Cuidar la calidad del texto: la limpieza del corpus es crucial. Si el texto está mal escrito, desordenado o lleno de ruido, incluso los mejores modelos tendrán un bajo rendimiento.

- Mantener balance entre clases: si las clases están desbalanceadas (por ejemplo, muchos incidentes “Bajo” y pocos “Alto”), se recomienda aplicar técnicas de muestreo o asignar pesos por clase.
- Elegir la representación adecuada del texto: si el corpus es grande y diverso, los embeddings (como en el caso del embedding layer de Keras) tienden a dar mejores resultados que las representaciones clásicas como Bag of Words o TF-IDF.
- Hiperparámetros coherentes: no sobreajustar los modelos. A veces modelos simples con buenos parámetros generales (como *dropout = 0.5*, *batch_size = 32*, *epochs = 10*) funcionan mejor que arquitecturas muy complejas mal ajustadas.
- Evaluar con varias métricas: no solo usar accuracy. La F1-score y la matriz de confusión ayudan a ver si el modelo está aprendiendo realmente o solo acierta por azar en clases mayoritarias.
- Documentar el proceso: guardar versiones del código, conjuntos de datos y modelos entrenados, lo cual facilita la replicación, mejora continua o implementación en otros entornos.

4. Documentación y recursos de apoyo: El marco de trabajo debe ir acompañado de una documentación clara que facilite su comprensión y reutilización. Por ello, se recomienda incluir:

- Comentarios en el código explicando cada paso del pipeline.
- Guías paso a paso para ejecutar desde la carga de datos hasta la generación del informe final de métricas.
- Recomendaciones de uso dependiendo del tipo de problema (por ejemplo, textos largos vs textos cortos, clasificación binaria vs multiclase).
- Ejemplos de ejecución: notebooks o scripts con ejemplos reales de entrenamiento y predicción.

Este proyecto deja como insumo no solo los modelos entrenados y evaluados, sino también una estructura lista para ser implementada en problemas similares, con la posibilidad de ser ajustada o mejorada a futuro por otros equipos, ya sea dentro de la misma entidad financiera o en otros sectores donde se manejen textos de reportes, solicitudes o reclamos.

Con base en la experiencia adquirida tanto en este como en el proyecto anterior, se propone este marco de trabajo como una guía completa y flexible para la clasificación automática de incidentes utilizando PLN. A diferencia del enfoque anterior —centrado en modelos clásicos y representaciones básicas del texto—, esta propuesta incorpora modelos más robustos y técnicas modernas de análisis de texto que, aunque requieren más procesamiento y entrenamiento, ofrecen una mejor comprensión del lenguaje utilizado por los usuarios en sus reportes.

La estructura modular del marco permite adaptarlo fácilmente a otros problemas

relacionados con clasificación de texto, análisis de sentimiento, categorización de reclamos, entre otros. Así, este trabajo no solo cumple su objetivo investigativo, sino que también deja una herramienta práctica que puede ser aplicada en contextos reales para mejorar la eficiencia de procesos internos y la toma de decisiones basadas en datos textuales.

9. RESULTADOS OBTENIDOS

9.1 DESCRIPCIÓN GENERAL DEL ENFOQUE Y MODELOS EVALUADOS

Para esta nueva etapa del proyecto se planteó la comparación entre diferentes modelos de clasificación de texto aplicados a reportes o incidentes registrados en una entidad financiera. A diferencia del proyecto anterior, donde se trabajaron modelos clásicos de aprendizaje automático supervisado, en este nuevo enfoque se buscó explorar técnicas más avanzadas, específicamente modelos basados en redes neuronales artificiales.

En el proyecto previo titulado "Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando PLN", se evaluaron un total de ocho modelos supervisados: Regresión Logística, Naive Bayes, K-vecinos más cercanos, Árboles de Decisión, Máquinas de Soporte Vectorial (SVM), Random Forest, modelos de ensamble como Boosting y Bagging, y finalmente un modelo de Red Neuronal simple. Todos estos modelos fueron evaluados con las mismas métricas que se utilizaron en esta etapa, lo que permitió mantener coherencia al momento de comparar resultados. Las métricas utilizadas fueron: Accuracy, Precisión, Recall y F1-Score, que en conjunto permiten analizar no solo la efectividad general del modelo, sino también su capacidad para identificar correctamente cada clase.

En este nuevo proyecto se centró la atención en modelos más complejos y especializados

en el procesamiento de lenguaje natural, por lo que se implementaron tres enfoques distintos basados en redes neuronales: una red neuronal convolucional (CNN), una red neuronal recurrente (RNN), y una red neuronal bidireccional con LSTM (Long Short-Term Memory). La intención fue observar si estos modelos, más enfocados en entender la estructura y el contexto del texto, lograban superar los resultados obtenidos con los modelos tradicionales usados previamente.

Esta comparación se realizó utilizando los mismos datos preprocesados y las mismas categorías de clasificación de incidentes, para asegurar que la evaluación fuera justa y reflejara realmente las capacidades de cada tipo de modelo frente al problema de clasificación textual en un entorno financiero.

9.2 RESULTADOS DE LOS MODELOS IMPLEMENTADOS EN ESTE PROYECTO

En la implementación de los modelos basados en redes neuronales artificiales cabe resaltar que estos modelos fueron entrenados con los datos preprocesados de los tickets reportados por los usuarios y evaluados con las mismas métricas utilizadas en el trabajo anterior: Accuracy, Precisión, Recall y F1-score.

A continuación, se presentan los resultados obtenidos por cada modelo:

Tabla 2. Resultados de los nuevos modelos implementado

Red artificial utilizada	Métricas de clasificación			
	Accuracy	Precisión	Recall	F1-score
RNN	0,4424	0,1957	0,4424	0,2714
CNN	0,7934	0,7936	0,7934	0,7928
Red Neuronal Bidireccional	0,7792	0,7786	0,7792	0,7785

Fuente: Elaboración propia.

Como se observa en la tabla, el modelo CNN fue el que presentó los mejores resultados en todas las métricas, seguido muy de cerca por la red bidireccional. El modelo RNN obtuvo un rendimiento considerablemente más bajo, especialmente en precisión y F1-score, lo cual sugiere que tuvo más dificultad para identificar correctamente las clases.

Una observación importante es que los tres modelos tomaron más tiempo para entrenar y generar resultados en comparación con los modelos del proyecto anterior. Esto es normal en redes neuronales, ya que implican un procesamiento más complejo y pesado. A pesar del esfuerzo computacional invertido, los resultados no mostraron diferencias abrumadoras entre CNN y la red bidireccional. Además, no se identificaron patrones o tendencias fuertes en las descripciones de los incidentes que ayudaran de forma significativa al proceso de clasificación automática. Esto también puede explicar por qué

las redes no lograron un rendimiento aún más alto, ya que el contenido de los tickets tiende a ser variable y poco estructurado.

9.3 ANÁLISIS Y COMPARACIÓN LOS MODELOS DEL PROYECTO ANTERIOR

Al comparar los resultados del nuevo proyecto con los obtenidos en el trabajo anterior, se puede concluir que las mejoras en la clasificación no fueron contundentes en todos los casos, aunque hubo modelos que sí mostraron un comportamiento competitivo frente a los métodos clásicos de aprendizaje automático.

A continuación se presentan los resultados de los modelos aplicados en el proyecto “Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando PLN”:

Tabla 3. Resultados proyecto previo con técnica bolsa de palabras

Técnicas de clasificación de Aprendizaje Automático	Métricas				
	Métricas de clasificación				Métricas de Regresión
	Accuracy	Precisión	Recall	F1-score	R2
Regresión logística					0,8075
Naive bayes	0,7890	0,7926	0,7890	0,7858	
K-vecinos	0,6526	0,6579	0,6526	0,6548	
Árboles de decisión	0,7559	0,7545	0,7559	0,7549	

Máquinas de soporte vectorial	0,7913	0,7905	0,7913	0,7907	
Random forest	0,7758	0,7756	0,7758	0,7736	
Boosting o Bagging	0,6477	0,6800	0,6477	0,6278	
Redes neuronales	0,7631	0,7629	0,7631	0,7628	

Fuente: Toma de “Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando PLN” página 43.

Tabla 4. Resultados proyecto previo con técnica TF-IDF.

Técnicas de clasificación de Aprendizaje Automático	Métricas				
	Métricas de clasificación				Métricas de Regresión
	Accuracy	Precisión	Recall	F1-score	R2
Regresión logística					0,8059
Naive bayes	0,7589	0,7889	0,7589	0,7358	
K-vecinos	0,5832	0,7285	0,7285	0,5404	
Árboles de decisión	0,7419	0,7414	0,7419	0,7414	
Máquinas de soporte vectorial	0,8127	0,8124	0,8127	0,8115	
Random forest	0,7644	0,7634	0,7644	0,7615	
Boosting o Bagging	0,6704	0,6704	0,6704	0,6482	
Redes neuronales	0,7665	0,7665	0,7665	0,7665	

Fuente: Toma de “Prototipo para análisis y clasificación de incidentes en una entidad financiera

utilizando PLN” página 43.

El modelo RNN, a pesar de su capacidad para procesar secuencias y tener en cuenta el orden de las palabras, fue el que tuvo el rendimiento más bajo entre todos los modelos evaluados. Obtuvo un Accuracy de 0,4424 y una F1-score de apenas 0,2714. Esto puede explicarse por el hecho de que los reportes o tickets analizados no tienen una estructura de texto clara, y en muchos casos son frases muy cortas o poco informativas. Al ser un modelo sensible al contexto secuencial, la falta de patrones fuertes afectó directamente su capacidad de aprendizaje.

En contraste, la CNN fue el modelo que presentó mejores resultados en este nuevo enfoque, con un Accuracy de 0,7934 y una F1-score de 0,7928. Si bien no superó al mejor modelo del proyecto anterior (Máquinas de Soporte Vectorial con TF-IDF, que logró un Accuracy de 0,8127), se acercó bastante. Una de las ventajas de la CNN es que no necesita entender el orden completo de las palabras, sino que trabaja mejor capturando patrones locales o combinaciones frecuentes de palabras. Esto resulta útil cuando los textos no siguen una redacción uniforme, como ocurre con los tickets de incidentes.

Por su parte, la Red Neuronal Bidireccional con LSTM mostró un rendimiento sólido (Accuracy de 0,7792 y F1-score de 0,7785), por encima de la mayoría de modelos del proyecto anterior, pero también sin superar a los mejores como SVM o Naive Bayes con Bag of Words. Lo interesante de este modelo es que intenta capturar el contexto completo

leyendo el texto hacia adelante y hacia atrás, lo cual puede ser una ventaja en textos más largos o mejor estructurados. Sin embargo, en este caso, su desempeño quedó apenas por debajo del CNN.

En general, no se puede afirmar que los modelos del nuevo proyecto superan de forma definitiva a los anteriores, pero sí se observó que los modelos basados en redes neuronales, especialmente CNN y la red bidireccional, tienen potencial para ofrecer resultados competitivos. Esto es importante considerando que, aunque requieren más tiempo de entrenamiento, también pueden adaptarse mejor a bases de datos más grandes o más limpias. Además, es posible que en un contexto con descripciones de incidentes más completas y consistentes, estos modelos de aprendizaje profundo logren destacarse aún más.

9.4 CONCLUSIONES DEL PROYECTO

El desarrollo de este trabajo de grado tuvo como principal propósito proponer un marco de trabajo efectivo para el análisis y clasificación de incidentes en una entidad financiera, haciendo uso del Procesamiento de Lenguaje Natural (PLN) y modelos de aprendizaje profundo. A partir del análisis comparativo con el proyecto previo titulado "Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando PLN", se lograron avances importantes que fortalecen el enfoque técnico y práctico en la clasificación automatizada de incidentes.

Una de las principales conclusiones que se puede extraer de este estudio es que los modelos basados en redes neuronales (CNN, RNN y red bidireccional con LSTM) presentaron un rendimiento competitivo frente a los modelos supervisados clásicos usados en el proyecto anterior. En particular, la red convolucional (CNN) se destacó por obtener los mejores resultados en términos de precisión y F1-score, mostrando su eficacia al capturar patrones relevantes en descripciones textuales breves y poco estructuradas.

La red bidireccional con LSTM, por su parte, demostró ser una alternativa sólida, con métricas apenas por debajo de la CNN, lo cual sugiere que su capacidad para analizar el contexto hacia adelante y hacia atrás en el texto fue útil para entender la semántica de los incidentes. No obstante, la red RNN obtuvo un desempeño inferior al de las otras dos, lo cual puede estar relacionado con su limitación para retener información contextual a largo plazo, lo que afectó su efectividad.

Desde el punto de vista metodológico, el uso de la metodología CRISP-DM permitió estructurar de forma ordenada cada fase del proyecto, facilitando la comprensión del problema, el análisis de los datos, la preparación del corpus, el entrenamiento de los modelos y la evaluación de resultados. Esta metodología también sirvió de base para la construcción del marco de trabajo propuesto, el cual recopila herramientas, técnicas,

buenas prácticas y plantillas replicables para que futuros proyectos puedan adaptarlo a diferentes contextos organizacionales.

El proyecto evidenció que la calidad de los datos y su preparación previa son fundamentales para obtener buenos resultados en tareas de PLN. Se aplicaron técnicas como la limpieza de caracteres especiales, la normalización del texto, la tokenización, el stemming y la eliminación de stopwords, lo cual permitió entregar al modelo un corpus más limpio y representativo. Esto ayudó a mitigar los ruidos propios del lenguaje natural y contribuyó al desempeño observado.

Aunque no se encontraron patrones lingüísticos dominantes que permitieran diferenciar claramente entre las categorías de incidentes, los modelos neuronales lograron capturar relaciones semánticas sutiles que permitieron mejorar en algunos casos la clasificación. Esto reafirma la utilidad del PLN como herramienta complementaria en auditoría interna, especialmente cuando se busca fortalecer los sistemas de alerta temprana y la toma de decisiones basada en datos.

Finalmente, el marco de trabajo propuesto aporta una solución integral que no solo considera el desarrollo técnico de modelos, sino también la posibilidad de escalar y replicar el análisis en otros procesos similares dentro de la entidad o incluso en otros sectores. La estructura modular del marco facilita su implementación, lo cual puede

contribuir a la transformación digital de procesos internos donde el análisis textual es clave.

10. REFERENCIAS BIBLIOGRAFICAS

- [1] [50] G. Velasco, J. Restrepo, "Prototipo para análisis y clasificación de incidentes en una entidad financiera utilizando NPL".
- [2] The Institute of Internal Auditors, "Artificial Intelligence – Considerations for the Profession of Internal Auditing," Glob. Perspect. INSIGHTS (Special Ed., p. 9, 2017.
- [3] I. Pedrosa, R. M. S. Laureano, and C. J. Costa, "Motivações dos auditores para o uso das Tecnologias de Informação na sua profissão: Aplicação aos Revisores Oficiais de Contas," RISTI - Rev. Iber. Sist. e Tecnol. Inf., no. 15, pp. 101–118, 2015, doi: 10.17013/risti.15.101-118.
- [4] D. M. Garces-Eslava, "Metodo de procesamiento de lenguaje natural y tecnicas de mineria de datos aplicadas a la clasificacion de incidentes informaticos," Interfases, no. 12, pp. 11–29, 2019, doi: 10.26439/interfases2019.n012.4635.
- [5] Augusto Cortez Vásquez, M., Hugo Vega Huerta, M., Jaime, L., & Quispe, P. (n.d.). Procesamiento de lenguaje natural.
- [6] Aho, A. v. (2008). Compiladores : principios, técnicas y herramientas. Pearson Educación.
- [7] ISACA, COBIT 2019 MARCO DE REFERENCIA. Schaumburg, 2019. [Online]. Available: www.isaca.org
- [8] [54] R. J. . . Celi-Parraga, E. A. Varela-Tapia, I. L. Acosta-Guzmán, y N. R. Montañó-

- Pulzara, «Técnicas de procesamiento de lenguaje natural en la inteligencia artificial conversacional textual », AP, vol. 3, n.º 4.1, pp. 40–52, nov. 2021.
- [9] Juan, A., Díaz, C., Directora, P., María, D., & Garcés Gómez, P. (n.d.). Pragmalingüística del disfemismo y la descortesía Los actos de habla hostiles en los medios de comunicación virtual.
- [10] “List of natural language processing toolkits - Wikipedia.” https://en.wikipedia.org/wiki/List_of_natural_language_processing_toolkits#Natural_language_processing_toolkits (accessed Jun. 29, 2022).
- [11] <http://alumni.media.mit.edu/~hugo/montylingua/>
- [12] <https://code.tutsplus.com/es/tutorials/introducing-the-natural-language-toolkit-nltk--cms-28620>
- [13] “Universidad Politécnica de Madrid Trabajo Fin de Grado Aplicación de Herramientas de Reconocimiento de Entidades Nombradas a SmarTerp,” 2021.
- [14] “NLTK :: Natural Language Toolkit”. NLTK :: Natural Language Toolkit. <https://www.nltk.org/> (accedido el 4 de julio de 2022).
- [15] [39] S. S. Martínez & D. F. López Análisis de la deserción universitaria en instituciones de la RED RIESC mediante metodología de minería de datos CRISP-DM. [online]. Disponible en: <http://hdl.handle.net/11349/39683>.
- [16] Fin, T., Grado, D. E., Martínez, J., Tutor, M., & Silla Jiménez, F. (2017). Iniciación al Entorno de Deep Learning Torch.

- [17] Ángel, M., & Ramírez, M. (n.d.). Inteligencia artificial aplicada a la ley de protección de datos.
- [18] Copyright 2020 by Tutorials Point. (2020). TutorialsPoint Gensim.
- [19] Shiliang Sun, Chen Luo, Junyu Chen, A review of natural language processing techniques for opinion mining systems, *Information Fusion*, Volume 36, 2017, Pages 10-25, ISSN 1566-2535, <https://doi.org/10.1016/j.inffus.2016.10.004>.
- [20] [22] Fan Zhang, Hasan Fleyeh, Xinru Wang, Minghui Lu, Construction site accident analysis using text mining and natural language processing techniques, *Automation in Construction*, Volume 99, 2019, Pages 238-248, ISSN 0926-5805, (<https://www.sciencedirect.com/science/article/pii/S0926580518306137>).
- [21] [23] Haroon, M. (2018). Comparative Analysis of Stemming Algorithms for Web Text Mining. *International Journal of Modern Education and Computer Science*, 10(9), 20–25. <https://doi.org/10.5815/ijmeecs.2018.09.03>
- [24] [26] Thierry Denoeux, Régis Lengellé, Initializing back propagation networks with prototypes, *Neural Networks*, Volume 6, Issue 3, 1993, Pages 351-363, ISSN 0893-6080, [https://doi.org/10.1016/0893-6080\(93\)90003-F](https://doi.org/10.1016/0893-6080(93)90003-F).
- [25] [43] ISA 265, “Communicating Deficiencies in Internal Control to Those Charged with Governance and Management,” paragraph 9.
- [27] PÉREZ RIVERA, Shadith, CALDERÓN MOLINA, Valentina. Decisiones empresariales mediante la analítica de datos en redes sociales: una aproximación al contexto colombiano. Universidad de los Andes. 2024.

- [28] [30] Zhou, J., Han, X., Liu, Z., & Sun, M. (2020). "A review of classification algorithms for text processing." *Journal of Artificial Intelligence Research*, 67(1), 1003-1028.
- [29] [31] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). "Attention is all you need." *Advances in Neural Information Processing Systems (NeurIPS)*.
- [32] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). "Latent Dirichlet Allocation." *Journal of Machine Learning Research*, 3, 993-1022.
- [33] [36] Ramos, J. (2003). "Using TF-IDF to determine word relevance in document queries." *Proceedings of the First Instructional Conference on Machine Learning*.
- [34] PIRABAN BARBOSA, Juanita, MORENO MARÍN, Julián, TRUJILLO PEREA, Jaime, y otros. *Procesamiento de lenguaje natural para el mercado accionario colombiano*. Universidad de los Andes. 2021.
- [35] [37] Carvajal Lombo, D. (2024). *Modelos de Deep learning para la clasificación de textos en los objetivos de desarrollo sostenible*. Universidad de los Andes. Disponible en: <https://hdl.handle.net/1992/73857>.
- [38] Parra Anzola, F *Categorización de letras de canciones de un portal web usando agrupación*. [Internet]. 2013 [citado: 2025, marzo] Universidad Nacional de Colombia Sede Bogotá Facultad de Ingeniería Departamento de Ingeniería de Sistemas e Industrial.
- [40] Basel Committee on Banking Supervision *The relationship between banking supervisors and banks' external auditors*. (2002).

- [41] Ceballos Gallego, Santiago. Diseño de una estrategia de limpieza y estandarización de direcciones postales a través de redes neurales recurrentes tipo LSTM. 2020
- [42] International Standard on Auditing (ISA) 200, Overall Objectives of the Independent Auditor and the Conduct of an Audit in Accordance with International Standards on Auditing, paragraph 11.
- [44] Antonio, F., Rubiano, G., de Trabajo, E., Mendez Jimenez, A., Cubillos, M., Dorley, B., & León López, E. (n.d.). Guía de Auditoría Interna basada en riesgos para Entidades Públicas Versión 4 Dirección de Gestión y Desempeño Institucional.
- [45] Yzquierdo Herrera, R. (2013). Ciencias de la Información Minería de proceso como herramienta para la auditoria (Vol. 44, Issue 2).
- [46] IBM. (2010). A New Way of Working: Insights from Global Leaders. United States of America: IBM Institute for Business Value.
- [47] Educación, S., Barral Rivada Rafael Bautista Mesa Horacio Molina Sánchez, A., Suárez Cortés Horacio Molina Sánchez, G., Franco Franco, W., Jiménez, N., Sarmiento Pavas, D., Serrano Amaya, G., Suárez Cortés, G., & Suárez Tirado, J. (n.d.). Convergencia con las Normas Internacionales de Auditoría (NIA) en Colombia Orientación Pedagógica sobre la aplicación de las Normas Internacionales de Auditoría (NIA) Autores.
- [48] <https://www.auditool.org/blog/auditoria-externa/4224-que-aspectos-se-deben-tener-en-cuenta-al-cierre-de-una-auditoria-de-informacion-financiera>
- [49] Castello Ricardo J. (n.d.). Auditoría en entornos informáticos.

- [51] Andrea Gómez-Jaramillo, P., González-Echavarría, F., & Iván Pérez-Rave, J. (n.d.). Technological incident classification model from a machine learning approach in insurance services •. Revista DYNA, 89(221), 161–167. <https://doi.org/10.15446/dyna.v89n221.100070>
- [52] Muñoz Cáceres, P. A. (n.d.). Referencia22. Retrieved June 29, 2022, from <https://repositorio.uchile.cl/bitstream/handle/2250/176979/Diseño-y-construcción-de-modelo-de-clasificación-de-incidentes-de-seguridad-usando-NLP-en-los-registros-de-texto.pdf?sequence=1&isAllowed=y>
- [53] Garces-Eslava, D. M. (2019). Método de procesamiento de lenguaje natural y técnicas de minería de datos aplicadas a la clasificación de incidentes informáticos. *Interfases*, 12, 11–29. <https://doi.org/10.26439/interfases2019.n012.4635>
- [55] D. M. Garces-Eslava, “Metodo de procesamiento de lenguaje natural y tecnicas de mineria de datos aplicadas a la clasificacion de incidentes informaticos,” *Interfases*, no. 12, pp. 11–29, 2019, doi: 10.26439/interfases2019.n012.4635.
- [56] P. A. Muñoz Cáceres, “Diseño y construcción de modelo de clasificación de incidentes de seguridad usando NLP en los registros de texto escrito para automatizar etiquetación.” pp. 1–85, 2020.