UNIVERSIDAD NACIONAL DE MOQUEGUA

ESCUELA PROFESIONAL DE INGENIERÍA DE

SISTEMAS E INFORMÁTICA

TESIS:

MODELOS DE DEEP LEARNING PARA LA DETECCION DE NOTICIAS FALSAS EN ESPAÑOL SOBRE COVID-19

PRESENTADO POR: PEDRO LUIS RAMOS CALLA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS E INFORMATICA

MOQUEGUA – PERU

2024

Folio Nº **015**

ACTA DE SUSTENTACIÓN

I. Modalidad:

Tesis : [X]
Trabajo de Suficiencia Profesional : [ ]
Otro: ..... : [ ]

II. Documento de aprobación para la sustentación:

Resolución de Facultad Nº 233-2024-FACIA-UNAM

Fecha: 21 de Noviembre 2024

III. El jurado ha dictaminado la (el) Tesis, TITULADO:

"Modelos de Deep Learning para la detección de noticias falsas en Español sobre Covid-19"

Bachiller: Pedro Luis Ramos Calla

Asesor Principal: Anibal Fernando Flores Garcia

Co-Asesor: —

IV. Se dictamina:

Como: Aprobado por Unanimidad la sustentación de manera:

Presencial [X] Virtual [ ]

Con nota: Catorce (14) por lo que; Sí [X] No [ ]

Opta el Título Profesional de INGENIERO DE SISTEMAS E INFORMÁTICA.

Siendo las 11:56 horas, del 04 de Diciembre del año 2024 se procede con la firma en conformidad de lo dictaminado por el Jurado.

Presidente
Ender Tito Chura

Primer Miembro

Segundo Miembro
Victor Yana Mamani

Pedro Luis Ramos Calla
Bachiller

Resumen

La presente investigación se enfoca en la aplicación de modelos de aprendizaje profundo para detectar noticias falsas en español, específicamente durante la pandemia de COVID-19, periodo de aumento significativo de la desinformación. Ante la limitada disponibilidad de recursos especializados en procesamiento de lenguaje natural (NLP) para el español, se busca identificar arquitecturas eficaces para esta tarea, enfrentando desafíos derivados de los matices lingüísticos y la necesidad de adaptar los modelos al contexto del idioma. En este estudio se evaluaron ocho modelos: BERT, BETO, ALBETO, DistilBERT, GRU, LSTM, BiLSTM y CNN, configurados para captar patrones específicos en español. Los modelos basados en Transformers, como BERT y BETO, son reconocidos por su capacidad para analizar el contexto bidireccional, mientras que los modelos recurrentes, como GRU y LSTM, destacan en la captura de dependencias temporales en las secuencias de texto. Cada modelo fue implementado con un preprocesamiento detallado y técnicas avanzadas de tokenización, junto con ajustes en hiperparámetros para optimizar su rendimiento. Para el entrenamiento y evaluación, se utilizó un conjunto de datos compuesto por artículos de noticias en español sobre el COVID-19. Los modelos fueron evaluados mediante métricas como precisión, exactitud, recall y F1-Score, con el objetivo de identificar el mejor rendimiento en la detección de noticias falsas. Los resultados mostraron que los modelos basados en Transformers, especialmente BERT y BETO, sobresalieron en precisión y exactitud, aunque los modelos recurrentes también resultaron viables para captar patrones en las secuencias. En conclusión, las arquitecturas basadas en Transformers ajustadas específicamente al español ofrecen una solución prometedora para abordar la desinformación en contextos críticos como el de la pandemia.

Palabras clave: Deep Learning, Detección de noticias falsas, Procesamiento de Lenguaje Natural, Transformers, COVID-19

Abstract

This research focuses on the application of deep learning models to detect fake news in Spanish, specifically during the COVID-19 pandemic, a period marked by a significant increase in misinformation. Given the limited availability of specialized resources for natural language processing (NLP) in Spanish, this study aims to identify effective architectures for this task, addressing challenges related to linguistic nuances and the need to adapt models to the language context. In this study, eight models were evaluated: BERT, BETO, ALBETO, DistilBERT, GRU, LSTM, BiLSTM, and CNN, configured to capture language-specific patterns in Spanish. Transformer-based models, such as BERT and BETO, are recognized for their ability to analyze bidirectional context, while recurrent models, such as GRU and LSTM, excel at capturing temporal dependencies in text sequences. Each model was implemented with detailed preprocessing and advanced tokenization techniques, along with hyperparameter tuning to optimize performance. For training and evaluation, a dataset consisting of Spanish news articles related to COVID-19 was used. The models were evaluated using metrics such as precision, accuracy, recall, and F1-score, in order to identify the best performance in fake news detection. The results showed that Transformer-based models, particularly BERT and BETO, outperformed the others in terms of precision and accuracy, although recurrent models also proved viable for capturing sequential patterns. In conclusion, Transformer-based architectures specifically fine-tuned for Spanish offer a promising solution for addressing misinformation in critical contexts such as the pandemic.

Keywords: Deep Learning, Fake News Detection, Natural Language Processing, Transformers, COVID-19

Tabla de contenido

I. Datos Generales 9
Titulo 9
Datos del investigador 9
Línea de investigación 9
II. Problema de Investigación 10
Descripción de la Realidad Problemática 10
Formulación del Problema de Investigación 11
Interrogante General 11
Interrogantes Específicas 11
Justificación 12
Objetivos 12
Objetivo General 12
Objetivos Específicos 12
Hipótesis 13
Hipótesis General 13
Hipótesis específicas 13
III. Marco Teórico 14
Antecedentes del problema 14
Bases teóricas 15
Infodemiología 15
Deep Learning 16
Procesamiento de Lenguaje Natural (PLN) 17
Transformers 18
COVID-19 19
CNN 20

BERT 23
Beto 25
ALBETO: 27
DistilBERT: 29
GRU: 30
LSTM: 32
BiLSTM: 34
Variables de la Investigación: 36
Variable Independiente 36
Variable Dependiente 37
IV. Metodología 38
Tipo de investigación 38
Diseño Experimental 38
Operacionalización de variables 38
Población y Muestra 39
Técnicas e instrumentos para la recolección de datos 39
Métodos y técnicas para la presentación y análisis de Datos 39
Procesamiento de los datos 39
Recolección de información: 39
Etiquetado de Datos 40
División de Datos 41
Modelos Utilizados 41
BERT 41
Descripción del Modelo 41
Modelamiento 42
Evaluación 43
BETO 44

Descripción del Modelo 44
Modelamiento 44
Evaluación 45
ALBETO 45
Descripción del modelo 45
Modelamiento 46
Evaluación 47
DistilBERT 47
Descripción del Modelo 47
Modelamiento 48
Evaluación 49
GRU 49
Descripción del Modelo 49
Modelamiento 50
Evaluación 51
LSTM 51
Descripción del Modelo 51
Modelamiento 52
Evaluación 53
BiLSTM 53
Descripción del Modelo 53
Modelamiento 54
Evaluación 55
CNN 55
Descripción del Modelo 55
Modelamiento 56
Evaluación 57

V. Resultados y Discusiones 58
Resultados de los modelos aplicados 58
Discusiones 58
Sobre la hipótesis General 58
Sobre las hipótesis específicas 59
Relevancia y Limitaciones 60
VI. Conclusiones 62
Conclusiones Generales 62
Conclusiones específicas 62
VII. Recomendaciones 64
VIII. Bibliografía 65
IX. Anexos 67

TABLA 1: OPERACIONALIZACIÓN DE VARIABLES 38
TABLA 2: RESULTADOS DE DESEMPEÑO DE MODELOS EN DETECCIÓN DE NOTICIAS FALSAS 58
TABLA 3: MATRIZ DE CONSISTENCIA 67

ILUSTRACIÓN 1: DIAGRAMA DEL ENCODER TRANSFORMER QUE MUESTRA LOS BLOQUES DE AUTOATENCIÓN, FEED-FORWARD Y NORMALIZACIÓN RESIDUAL. 19
ILUSTRACIÓN 2: ARQUITECTURA DE LA CNN PARA CLASIFICACIÓN DE TEXTO 22
ILUSTRACIÓN 3: FUNCIONAMIENTO INTERNO DE UNA UNIDAD GRU MOSTRANDO LAS PUERTAS DE REINICIO (RESET GATE) Y ACTUALIZACIÓN (UPDATE GATE) 32
ILUSTRACIÓN 4: ESTRUCTURA INTERNA DE UNA CELDA LSTM 34
ILUSTRACIÓN 5: ARQUITECTURA DE UNA RED BIDIRECTIONAL LSTM (BILSTM) PARA EL PROCESAMIENTO DE SECUENCIAS DE TEXTO 36
ILUSTRACIÓN 6: DATASET DE NOTICIAS FALSAS EN ESPAÑOL SOBRE COVID-19 69
ILUSTRACIÓN 7: CONFIGURACIÓN DEL MODELO BERT 70
ILUSTRACIÓN 8: MATRIZ DE CONFUSIÓN DEL MODELO BERT 71
ILUSTRACIÓN 9: CONFIGURACIÓN DEL MODELO BETO 72

ILUSTRACIÓN 10: MATRIZ DE CONFUSIÓN DEL MODELO BETO 73
ILUSTRACIÓN 11: CONFIGURACION DEL MODELO ALBETO 74
ILUSTRACIÓN 12: MATRIZ DE CONFUSIÓN DEL MODELO ALBETO 74
ILUSTRACIÓN 13: CONFIGURACIÓN DEL MODELO DISTILBERT 76
ILUSTRACIÓN 14: MATRIZ DE CONFUSIÓN DEL MODELO DISTILBERT 76
ILUSTRACIÓN 15: CONFIGURACIÓN DEL MODELO GRU 77
ILUSTRACIÓN 16: MATRIZ DE CONFUSIÓN DEL MODELO GRU 78
ILUSTRACIÓN 17: CONFIGURACIÓN DEL MODELO LSTM 79
ILUSTRACIÓN 18: MATRIZ DE CONFUSIÓN DEL MODELO LSTM 79
ILUSTRACIÓN 19: CONFIGURACIÓN DEL MODELO BILSTM 80
ILUSTRACIÓN 20: MATRIZ DE CONFUSIÓN DEL MODELO BILSTM 81
ILUSTRACIÓN 21: CONFIGURACIÓN DEL MODELO CNN 82
ILUSTRACIÓN 22: MATRIZ DE CONFUSIÓN DEL MODELO CNN 82

I. Datos Generales

Titulo

Modelos de Deep Learning para la Detección de Noticias Falsas en español sobre COVID-19

Datos del investigador

Pedro Luis Ramos Calla

Línea de investigación

Inteligencia Artificial

II. Problema de Investigación

Descripción de la Realidad Problemática

Durante los primeros meses de la pandemia, sus efectos comenzaron a nivel de infectados y decesos, hospitales y centros de salud con alta demanda debido al incremento de pacientes de COVID-19. Los números reportados oficiales fueron publicados por el Ministerio de Salud y permitían conocer la evolución de los casos en Perú y el gobierno del Perú estableció políticas de salud pública para contener la rápida propagación, cuarentena totales y luego parciales. Permitiendo que sectores esenciales puedan continuar sus labores considerando las recomendaciones del Ministerio de Salud.

La infodemiología, término acuñado por Eysenbach tiene como objetivo mejorar la salud pública analizando las fuentes digitales que los usuarios de Internet usan, como foros, blogs, redes sociales, motores de búsqueda. Eysenbach demostró su argumento al encontrar una correlación entre las búsquedas de Google y los casos de gripe. Demostrando que podían ser usadas estas fuentes de datos para vigilancia epidemiológica.

Las noticias falsas en las redes sociales son un problema crítico ya que pueden perjudicar a la población (Zhang & Ghorbani, 2020). Las noticias falsas pueden crearse con la intención de engañar, manipular, difamar o generar miedo en la audiencia. La Organización Mundial de la Salud ha afirmado que las noticias falsas o infodemia son más mortales que el propio virus COVID-19, han afectado a las personas tanto física como mentalmente (I. , D., M. , & S. , 2022).

La pandemia de COVID-19 (Organizacion Mundial de la Salud, s.f.). ha generado un gran volumen de noticias falsas, desinformación y teorías de conspiración en todo el mundo. Algunas de las noticias falsas más comunes relacionadas con la pandemia incluyen teorías de conspiración (Douglas, 2021) sobre el origen del virus, como afirmar que el virus fue creado por el hombre; afirmaciones falsas sobre remedios o curas para enfermedades (Mustapha , Abubakar , Abubakar , Ismaeel , & Abubakar, 2021), información errónea sobre medidas preventivas, como afirmar que el uso de mascarillas es perjudicial para la

salud o que la vacunación no es segura (Louise , y otros, 2022); cifras de casos y muertes, como afirmar que las cifras son exageradas o manipuladas por razones políticas.

Por tanto, la presente tesis tiene como objetivo realizar un estudio infodemiológico usando una metodología de minería de datos y técnicas de procesamiento de lenguaje natural. y fuentes de datos digitales como resultados de motores de búsqueda (Google), redes sociales y datos de movilidad o desplazamiento. A la fecha existen algunos estudios relacionados con énfasis en Perú, más el contexto continúa evolucionando con 4 la aparición de nuevas variantes, vacunas y políticas del gobierno referentes a la salud pública. Y bajo nuestro entendimiento, no existe un estudio actualizado sobre el tema principal del proyecto, y que integre varias fuentes digitales para su estudio y que pueda ser una fuente de información para el gobierno y realizar mejores políticas públicas en salud.

Formulación del Problema de Investigación

Interrogante General

¿Qué modelos son los más recomendados para la clasificación de noticias?

Interrogantes Específicas

¿Qué modelos y enfoques existen para detectar noticias falsas sobre COVID-19, y cómo pueden aplicarse en el contexto específico de Perú y el idioma español?

¿Cómo afecta el preprocesamiento y la diversidad del conjunto de datos en términos de fuentes y estilos de redacción al recall de los modelos de Deep Learning en la detección de noticias falsas sobre COVID-19 en español?

¿De qué manera la integración de técnicas de aumento de datos y validación cruzada durante el entrenamiento de modelos de Deep Learning afecta el F1-score en la detección de noticias falsas sobre COVID-19?

¿Cuál es el rendimiento comparativo de diferentes modelos de Deep Learning entrenados para la detección de noticias falsas, y en qué medida los modelos basados en Transformers superan a otros tipos de modelos en esta tarea?

Justificación

En lo social, El presente proyecto puede proporcionar información valiosa sobre las preocupaciones y percepciones de la población en relación con cuestiones del COVID-19.

En lo económico, el tesista cuenta con el apoyo de la Universidad Nacional de Moquegua, ya que el tema propuesto es parte de un proyecto de investigación financiado con fondos de canon y regalías mineras.

Desde el punto de vista tecnológico se cuenta con las herramientas de software y conocimientos necesarios para llevar a cabo la investigación.

Por lo mencionado en los párrafos precedentes se justifica la ejecución del proyecto de investigación descrito.

Objetivos

Objetivo General

Proponer Modelos de Deep learning para la detección de noticias falsas en español sobre COVID-19

Objetivos Específicos

Investigar y seleccionar algoritmos de Deep learning adecuados para la clasificación de texto

Recopilar y preprocesar un conjunto de datos de noticias en español sobre COVID-19

Entrenar y evaluar los modelos seleccionados utilizando el conjunto de datos preprocesado

Comparar el rendimiento de los modelos y seleccionar el más efectivo

Hipótesis

Hipótesis General

La implementación de modelos avanzados de Deep Learning, basados en técnicas de Transformers mejora la precisión, el recall y el F1-score en la detección de noticias falsas en español sobre COVID-19, en comparación con modelos de Deep Learning CNN y RNN.

Hipótesis específicas

Los modelos de Deep Learning basados en Transformers, exhibe una mayor precisión en la clasificación de noticias falsas en comparación con los modelos basados en CNN y RNN.

La integración de técnicas de aumento de datos y validación cruzada en el entrenamiento de modelos de Deep Learning para la detección de noticias falsas incrementa el F1-score de los modelos.

El entrenamiento de modelos de Deep Learning con un conjunto de datos preprocesado y diverso en términos de fuentes y estilos de redacción mejora el recall en la detección de noticias falsas sobre COVID-19 en español.

La comparación del rendimiento entre modelos de Deep Learning entrenados revela que los modelos basados en Transformers superan a otros tipos en la tarea de detección de noticias falsas

III. Marco Teórico

Antecedentes del problema

Se han llevado a cabo estudios de vigilancia epidemiológica utilizando un enfoque de infodemiología para analizar diversas enfermedades, siguiendo la propuesta de Eysenbach. Esto ha implicado un aumento en las fuentes digitales utilizadas para la recolección de datos y en el alcance del análisis realizado.

(Steffen , Michael , & Artur , 2021), Se observó un aumento en la cantidad de investigaciones realizadas sobre infodemiología entre marzo de 2020 y enero de 2021. Durante este período, se utilizó una variedad de fuentes, incluido Google Trends, para analizar el interés en temas relacionados con el coronavirus, el confinamiento y el distanciamiento social. Estas palabras clave se utilizaron para rastrear las tendencias de búsqueda y evaluar su impacto en la información disponible en línea. Los estudios, de naturaleza global, confirmaron la hipótesis de que el aumento en las búsquedas en Google se correlacionaba con un incremento en las publicaciones relacionadas con la vigilancia de la información, la infodemiología y el uso de Google Trends como herramienta de análisis.

(Martin , Stuckler, & Arora, 2019), reafirma uso de motores de búsqueda en los últimos años para acceder a información sobre una variedad de temas relacionados con servicios para consumidores y productos de consumo ha sido robusta. Simultáneamente, ha surgido un interés en aumento por obtener respuestas a preguntas relacionadas con la salud. Como resultado, tanto profesionales de la salud como investigadores de salud 11 pública han reconocido el potencial de fuentes de datos disponibles, como Google Trends, que muestra los patrones de búsqueda de los usuarios. Por último, se analizan las fortalezas, limitaciones y recomendaciones para futuras investigaciones en este ámbito.

(Reem & Amal, 2021) Llevaron a cabo una investigación infodemiológica con el fin de explorar la difusión de temas de discusión en árabe y examinaron el comportamiento de los usuarios en Arabia Saudita. Utilizaron Google Trends y Twitter como fuentes de datos. Asimismo, emplearon los datos proporcionados por Google Mobility para analizar laconexión entre las actividades de los usuarios y las

diversas fases de la pandemia, con el objetivo de prever nuevos casos de COVID-19 basándose en los cambios en la movilidad de los usuarios.

(Chire Saire & Pineda-Briseño , 2021), Se llevó a cabo una recopilación de los trabajos publicados entre marzo y julio de 2020, centrándose en la región de América Latina y en técnicas de Minería de Datos aplicadas al Procesamiento de Lenguaje Natural. La cantidad de trabajos realizados es limitada, y hay pocos estudios Infodemiológico en Perú. Las fuentes de datos utilizadas fueron Twitter y Google Trends. El objetivo principal del capítulo de libro es promover la investigación en esta área a nivel de los países latinoamericanos.

(Chire-Saire, Khalid , Oblitas-Cruz, & Ahmed , 2021) Se llevó a cabo un estudio de los países hispanohablantes de Sudamérica utilizando un enfoque infodemiológico y utilizando como fuentes de datos Twitter, Google Trends y los casos diarios de COVID19 reportados por los países. El objetivo fue investigar evidencias sobre el impacto del COVID-19 en la salud mental de la población. Se confirmó que el interés en las 12 búsquedas y publicaciones estaba directamente relacionado con la evolución de la pandemia, y que su disminución sugería una adaptación de las personas al contexto de la pandemia.

Bases teóricas

Infodemiología

Además de analizar la circulación de información, la infodemiología se centra en comprender cómo las personas interpretan, comparten y reaccionan ante los mensajes relacionados con la salud. Esta perspectiva sociotécnica permite identificar no solo la difusión de datos verificados, sino también de rumores, mitos y desinformación, los cuales pueden influir negativamente en las decisiones preventivas de la población. A diferencia de los estudios epidemiológicos tradicionales, que examinan la propagación de enfermedades biológicas, la infodemiología analiza la “propagación informacional”, un fenómeno que puede extenderse incluso más rápido que un virus físico debido al alcance global e inmediato de los medios digitales.

Un concepto clave dentro de la infodemiología es la **infodemia**, que se refiere al exceso de información tanto verídica como falsa que circula durante una emergencia sanitaria, dificultando que las personas puedan identificar fuentes fiables. Este fenómeno se intensificó durante la pandemia por COVID-19, donde la combinación de incertidumbre, miedo y amplio acceso a redes sociales generó un entorno donde la desinformación se propagó de manera masiva. La infodemiología aporta herramientas para analizar estos flujos informacionales y medir su impacto, permitiendo a las autoridades diseñar estrategias de comunicación más efectivas.

Asimismo, la infodemiología se apoya en técnicas de análisis computacional, como minería de datos, análisis semántico y modelos de inteligencia artificial, que permiten monitorear grandes volúmenes de contenido en tiempo real. Estas herramientas facilitan la detección de tendencias emergentes, la identificación de patrones de desinformación y la evaluación del sentimiento público frente a temas relacionados con la salud. Gracias a ello, esta disciplina se ha vuelto fundamental para apoyar políticas públicas, campañas de prevención y sistemas de vigilancia epidemiológica, contribuyendo a reducir los efectos negativos de la desinformación en la sociedad.

Deep Learning

El Deep Learning se basa en la idea de que una red neuronal con múltiples capas puede aprender representaciones progresivamente más complejas a medida que avanza a través de diferentes niveles de abstracción. En las primeras capas, la red aprende características básicas, mientras que en las capas más profundas identifica patrones más sofisticados, lo que le permite resolver tareas complejas sin necesidad de diseñar manualmente reglas o atributos específicos. Este enfoque contrasta con los métodos tradicionales de aprendizaje automático, donde era imprescindible definir características manualmente para que el modelo pudiera utilizarlas durante el entrenamiento.

Con el aumento exponencial de los datos disponibles y la mejora del hardware especialmente las unidades de procesamiento gráfico (GPU) y los aceleradores especializados el Deep Learning ha permitido abordar problemas que antes resultaban inalcanzables. Su capacidad para aprender directamente de datos

no estructurados, como texto, imágenes, audio o video, lo ha convertido en una herramienta clave en áreas como la visión por computadora, el reconocimiento de voz y el análisis del lenguaje natural. En particular, los avances recientes en arquitecturas como Transformers han transformado por completo el procesamiento del lenguaje, logrando resultados sin precedentes en tareas de clasificación, traducción y generación de texto.

Asimismo, el Deep Learning ha demostrado ser altamente flexible, ya que sus modelos pueden adaptarse mediante técnicas de entrenamiento como el aprendizaje supervisado, no supervisado, semi-supervisado y el fine-tuning. Esta última técnica es especialmente útil en aplicaciones donde se dispone de datos limitados, permitiendo ajustar modelos preentrenados en grandes corpus para resolver problemas específicos. Gracias a esta adaptabilidad, el Deep Learning se ha convertido en el estándar actual para el desarrollo de sistemas inteligentes capaces de comprender y procesar información de manera similar al razonamiento humano.

Procesamiento de Lenguaje Natural (PLN)

El Procesamiento de Lenguaje Natural (PLN) es una rama de la inteligencia artificial cuyo objetivo principal es permitir que las computadoras comprendan, analicen y generen lenguaje humano de manera eficiente. Para lograrlo, el PLN combina técnicas lingüísticas tradicionales con modelos matemáticos y estadísticos capaces de interpretar la estructura y el significado del texto. Esta disciplina ha avanzado significativamente gracias a la disponibilidad de grandes volúmenes de datos y al uso de técnicas de aprendizaje profundo, que han permitido mejorar la comprensión semántica y contextual del lenguaje.

Una de las tareas fundamentales del PLN es la **representación del texto**, proceso mediante el cual las palabras y oraciones se transforman en estructuras numéricas que los modelos pueden interpretar. Tradicionalmente, esto se realizaba mediante métodos simples como el *Bag of Words* o los *TF-IDF*, los cuales ignoraban el orden y el contexto de las palabras. Sin embargo, el desarrollo de los **embeddings** representaciones densas que capturan relaciones semánticas permitió a los modelos trabajar con características más ricas, donde palabras con significados

similares se ubican cerca dentro de un espacio vectorial. Este enfoque representó un cambio crucial en la capacidad de los sistemas para procesar lenguaje natural.

Además, el PLN abarca una amplia gama de tareas como el análisis de sentimientos, la extracción de información, el resumen automático, la traducción automática y la clasificación de textos. En los últimos años, la introducción de arquitecturas como los **Transformers** ha impulsado un salto cualitativo, ya que estas permiten capturar dependencias a larga distancia dentro de un texto y comprender relaciones complejas entre palabras. Como resultado, tareas que previamente eran consideradas extremadamente difíciles ahora pueden resolverse con altos niveles de precisión, impulsando avances en áreas como la salud digital, la educación, el comercio electrónico y la comunicación automatizada.

Transformers

Los modelos Transformers representan un cambio fundamental en la forma en que se procesan secuencias, ya que eliminan la dependencia del procesamiento paso a paso característico de las redes recurrentes. Su arquitectura se basa exclusivamente en mecanismos de atención, lo que permite evaluar simultáneamente la relevancia de cada palabra respecto a las demás dentro de una oración. Esto facilita la captura de relaciones semánticas tanto cercanas como distantes, independientemente de la longitud de la secuencia, algo que resultaba limitado en modelos como LSTM o GRU.

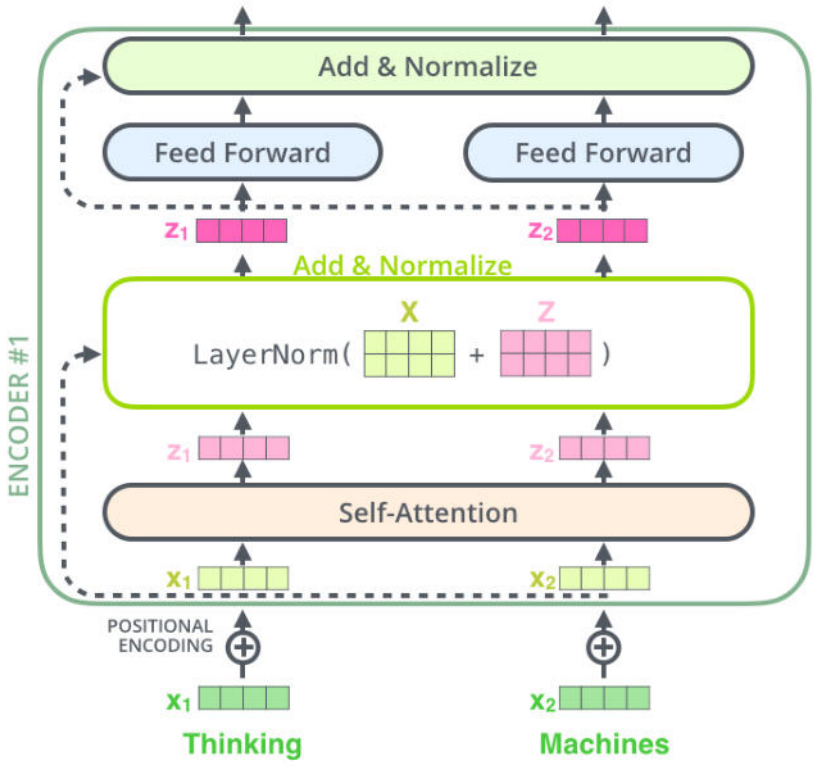
Una característica central de los Transformers es el mecanismo de auto-atención (*self-attention*), mediante el cual cada palabra obtiene una representación contextualizada al ponderar la influencia de todas las demás palabras en la secuencia. Este mecanismo utiliza tres proyecciones fundamentales: *queries*, *keys* y *values*, a partir de las cuales se calcula qué elementos del texto deben recibir mayor énfasis. Como resultado, el modelo logra comprender de manera más precisa estructuras complejas, como relaciones sintácticas y dependencias semánticas.

Otra innovación relevante es el uso de *multi-head attention*, donde múltiples cabezales de atención operan en paralelo para capturar distintos tipos de relaciones lingüísticas. Asimismo, los Transformers incorporan *positional embeddings* para

introducir información sobre el orden de las palabras en una arquitectura que no procesa el texto de manera secuencial. La combinación de estos elementos hace que los Transformers sean altamente eficaces para tareas como clasificación de texto, reconocimiento de entidades, traducción automática y generación de lenguaje natural.

Finalmente, los Transformers han demostrado una gran capacidad de escalabilidad, permitiendo el entrenamiento de modelos de gran tamaño que capturan patrones complejos del lenguaje. Esta característica se refleja en modelos como BERT y sus variantes, los cuales han establecido nuevos estándares de rendimiento en múltiples tareas del procesamiento del lenguaje natural.

Ilustración 1: Diagrama del encoder Transformer que muestra los bloques de autoatención, feed-forward y normalización residual.



COVID-19

La pandemia de COVID-19 representó uno de los mayores desafíos sanitarios y sociales del siglo XXI, debido no solo a su rápida propagación, sino también al impacto que generó en los sistemas de comunicación global. La

necesidad de información inmediata provocó que millones de personas recurrieran a internet y redes sociales para obtener datos sobre síntomas, contagios, tratamientos y medidas preventivas. Este escenario creó un entorno donde la información verificada convivió con rumores, interpretaciones erróneas y noticias falsas, incrementando la dificultad para que la población diferenciara entre fuentes confiables y contenidos engañosos.

En este contexto emergió un fenómeno crítico conocido como **infodemia**, caracterizado por la sobreabundancia de información correcta o incorrecta que circuló a gran velocidad durante la crisis sanitaria. La Organización Mundial de la Salud (OMS) alertó que la infodemia podía afectar negativamente el comportamiento de las personas, ya que la exposición a información errónea podía llevar a decisiones inadecuadas, disminuir la confianza en las autoridades sanitarias e incluso generar rechazo hacia las medidas de protección. Por ello, comprender el flujo de información durante la pandemia se volvió tan importante como estudiar la propagación del virus mismo.

Además, el COVID-19 impulsó la producción masiva de contenido en plataformas digitales, donde los algoritmos de recomendación, la viralización espontánea y las comunidades virtuales contribuyeron a amplificar mensajes imprecisos. Este entorno potenció la diseminación de teorías conspirativas, remedios falsos, desinformación sobre vacunas y contenidos manipulados que afectaron la percepción pública del riesgo. Debido a esto, surgió la necesidad de desarrollar herramientas automatizadas incluyendo modelos de procesamiento del lenguaje natural capaces de detectar y clasificar noticias falsas relacionadas con el COVID-19, con el objetivo de mitigar su impacto en la salud pública.

CNN

Además de su uso tradicional en visión por computadora, las redes convolucionales han demostrado ser altamente eficaces para el análisis de texto gracias a su capacidad para capturar patrones locales en secuencias lingüísticas. A diferencia de los modelos basados en recurrencia, las CNN no procesan la información de manera secuencial, lo que permite reducir significativamente el costo computacional y acelerar el entrenamiento del modelo. Esta propiedad se

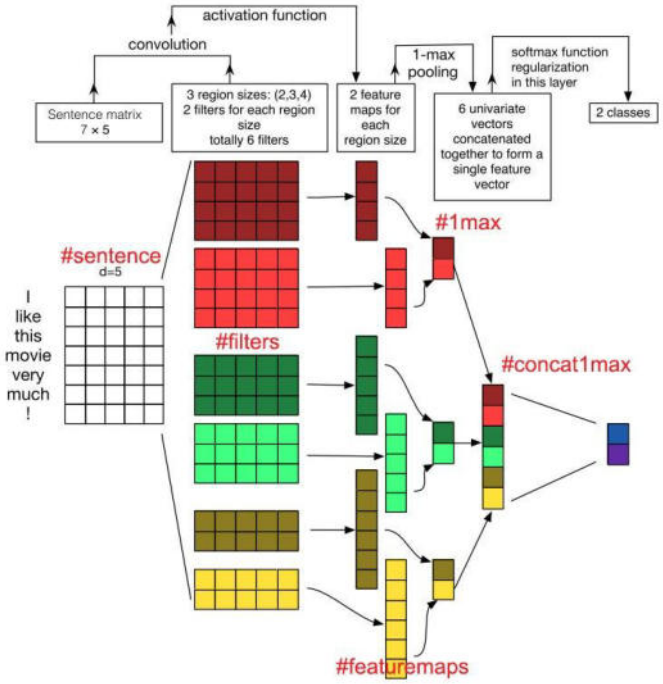
vuelve especialmente útil en tareas de clasificación donde se requiere identificar fragmentos característicos del texto, tales como expresiones específicas, construcciones sintácticas frecuentes o combinaciones de términos que suelen aparecer en contenido desinformativo.

En el enfoque propuesto por (Kim, 2014), considerado una referencia fundamental en el uso de CNN para clasificación de oraciones, los filtros convolucionales actúan como detectores de patrones lingüísticos. Cada filtro puede captar un tipo particular de n-grama o estructura semántica, lo que permite que el modelo identifique relaciones locales independientemente de su ubicación en la secuencia. Este principio resulta sumamente efectivo en textos cortos o medianos como titulares o fragmentos informativos donde la presencia de ciertas expresiones puede indicar la veracidad o falsedad del contenido.

Posteriormente, las operaciones de **max-pooling** permiten extraer las características más relevantes detectadas por cada filtro. Esta etapa introduce una forma de invarianza posicional, lo que significa que el modelo reconocerá un patrón importante sin importar en qué parte del texto aparezca. Dicha característica es especialmente útil en el análisis de noticias falsas, donde el lenguaje manipulador o engañoso puede presentarse en distintas posiciones dentro del mensaje. Tras la etapa de extracción de características, las activaciones resultantes se combinan y se envían a una capa totalmente conectada que realiza la predicción final de clasificación.

Gracias a su simplicidad, rapidez y efectividad, las CNN continúan siendo una alternativa competitiva frente a modelos más complejos, especialmente cuando se trabaja con grandes volúmenes de datos o cuando se requiere un sistema con bajo costo computacional. Su capacidad para detectar estructuras locales y su arquitectura altamente paralelizable hacen que este tipo de redes siga siendo ampliamente utilizado en tareas de procesamiento de lenguaje natural y minería de textos.

Ilustración 2: arquitectura de la CNN para clasificación de texto



Modelo Matemático:

1. Representación del Texto

Cada palabra se convierte en un embedding de tamaño 128:

$$x_i \in R^{128}$$

y la secuencia completa es:

$$X = [x_1, x_2, \dots, x_t]$$

2. Convolución

Usas 128 filtros de tamaño 5. Matemáticamente, un filtro $W^{(j)}$produce:

$$c_i^j = ReLU(\sum_{t=0}^{4} w_i^j * x_{i+t} + b^j)$$

Cada c_i^jrepresenta la activación del filtro al detectar un patrón en la ventana

$$[x_i, x_{i+2}, \dots, x_{i+4}]$$

3. Feature Map

Aplicamos el filtro a toda la secuencia:

$$\boldsymbol{C(j)} = [\boldsymbol{c_1^j}, \boldsymbol{c_2^j}, \dots, \boldsymbol{c_{t-4}^j},)]$$

Cada feature map captura cómo responde un filtro a lo largo del texto.

4. Global Max Pooling

Selecciona la activación más fuerte:

$$\boldsymbol{h_j} = \left(\frac{max}{i}\right) \boldsymbol{c_1^j}$$

Este valor representa "el patrón más importante detectado por ese filtro".

Todos los filtros producen:

$$h = \ [\boldsymbol{h_1}, \boldsymbol{h_2}, \dots, \boldsymbol{h_{128}}]$$

5. Capas Densas

Primero:

$$\boldsymbol{z} = \boldsymbol{ReLU}(\boldsymbol{w_1 h} + \boldsymbol{b_1})$$

Luego la salida final

$$\boldsymbol{\hat{y}} = \boldsymbol{\sigma}(\boldsymbol{w_2 z} + \boldsymbol{b_2})$$

donde σes la función sigmoide para clasificación binaria.

BERT

El modelo BERT (Bidirectional Encoder Representations from Transformers) es una arquitectura basada en Transformers que introdujo un enfoque innovador para el procesamiento de lenguaje natural. A diferencia de modelos tradicionales que procesan el texto de izquierda a derecha o de derecha a izquierda, BERT utiliza una estrategia de aprendizaje completamente bidireccional, lo que le

permite analizar simultáneamente el contexto anterior y posterior de cada palabra. Este diseño le proporciona una comprensión mucho más profunda del significado de las oraciones y de las relaciones entre los términos, convirtiéndolo en uno de los modelos más influyentes en el campo del PLN (Devlin, Chang, Lee, & Toutanova, 2019).

BERT se entrena mediante dos tareas fundamentales: Masked Language Modeling (MLM) y Next Sentence Prediction (NSP). En MLM, ciertas palabras de la oración se enmascaran aleatoriamente y el modelo debe predecirlas, obligándolo a comprender el contexto de manera global. En NSP, el modelo aprende relaciones entre pares de oraciones, lo que resulta útil en tareas como clasificación de textos, respuesta automática y detección de relaciones semánticas. Gracias a este preentrenamiento masivo, BERT puede adaptarse fácilmente a nuevas tareas con un proceso llamado fine-tuning, donde se ajusta a un conjunto de datos específico, para esta investigación.

En la detección de desinformación, BERT ofrece ventajas significativas. Su capacidad para comprender el significado contextual de cada palabra le permite identificar matices lingüísticos que suelen aparecer en contenidos engañosos, como lenguaje sensacionalista, afirmaciones imprecisas o estructuras discursivas manipuladoras. Además, al trabajar con el token especial [CLS], BERT resume toda la información del texto en un solo vector de representación, el cual se utiliza para la clasificación final. Esta propiedad lo convierte en un modelo potente, preciso y altamente adecuado para distinguir entre noticias verdaderas y falsas en español, especialmente cuando el contenido contiene afirmaciones ambiguas o redactadas de manera engañosa.

Modelo Matemático

1. **Entrada y tokenización**

   Sea una secuencia de texto tokenizada:

   $$\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_T}]$$

2. **Embeddings de entrada**

   Cada token se representa como la suma de tres embeddings:

   $$\boldsymbol{E_i} = \boldsymbol{E_i^{token}} + \boldsymbol{E_i^{position}} + \boldsymbol{E_i^{segment}}$$

3. **Proyecciones Query, Key y Value**

   Para cada capa Transformer:

$$\boldsymbol{Q = EW_Q}, \quad \boldsymbol{K = EW_K}, \quad \boldsymbol{V = EW_V}$$

4. **Self-Attention escalada**

$$\backslash\boldsymbol{text}\{\boldsymbol{Attention}\}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) \;=\; \text{softmax}!\left(\frac{\boldsymbol{QK}^{\top}}{\sqrt{\boldsymbol{d_k}}}\right)\boldsymbol{V}$$

5. **Multi-Head Attention**

$$\text{MHA}(\boldsymbol{E}) = \text{Concat}(\boldsymbol{head_1}, \ldots, \boldsymbol{head_h})\boldsymbol{W_O}$$

6. **Red Feed-Forward**

$$\text{FFN}(\boldsymbol{x}) = \boldsymbol{W_2}\,\text{ReLU}(\boldsymbol{W_1 x} + \boldsymbol{b_1}) + \boldsymbol{b_2}$$

7. **Representación del token [CLS]**

   Tras pasar por todas las capas del encoder, se obtiene el vector:

$$\boldsymbol{h}_{[\boldsymbol{CLS}]}$$

   Este vector resume la información semántica global del texto.

8. **Capa de clasificación**

   BERT usa una capa lineal + softmax interno

$$\hat{\boldsymbol{y}} =\backslash\textbf{textsoftmax}\big(\mathbf{W_{cls}h_{[CLS]}} + \mathbf{b_{cls}}\big)$$

9. **Función de pérdida**

$$\boldsymbol{\mathcal{L}} = -\sum_{\boldsymbol{i=1}}^{\boldsymbol{2}} \boldsymbol{y_i}\,\textbf{lo g}(\hat{\boldsymbol{y}}_{\boldsymbol{i}})$$

Beto

El modelo BETO es una variante de BERT diseñada específicamente para el idioma español. Fue desarrollado con el objetivo de superar las limitaciones de los modelos preentrenados principalmente en inglés al ser aplicados a tareas de procesamiento de lenguaje natural en español. BETO mantiene la arquitectura original de BERT basada en Transformers, pero se beneficia de un preentrenamiento realizado sobre grandes corpus en español, lo que mejora su desempeño en tareas de clasificación de texto (Sergio Damian, Hiram, & Gelbukh, 2020).

En esta investigación se emplea la variante dccuchile/bert-base-spanish-wwm-uncased, la cual fue entrenada utilizando la técnica de Whole Word Masking,

que permite capturar de manera más precisa el significado contextual de las palabras en español. Este enfoque ha demostrado mejoras significativas en diversas tareas de procesamiento de lenguaje natural en dicho idioma (Cañete, y otros, 2020)

Gracias a su arquitectura bidireccional y a su entrenamiento específico para español, BETO resulta especialmente adecuado para la clasificación de noticias falsas relacionadas con el COVID-19, ya que permite identificar patrones lingüísticos y semánticos propios del discurso desinformativo.

Modelo Matemático

1. **Entrada y tokenizacion**

   Sea una secuencia de texto en español representada como una secuencia de tokens:

   $$X = [x_1, x_2, \dots, x_T]$$

2. **Embeddings de entrada**

   Cada token se representa como la suma de tres embeddings: token, posición y segmento.

   $$E_i = E_i^{token} + E_i^{position} + E_i^{segment}$$

3. **Cálculo de Query, Key y Value (Self-Attention)**

   A partir de la matriz de embeddings E, se calculan las matrices:

   $$Q = EW_Q, \quad K = EW_K, \quad V = EW_V$$

4. **Self-Attention escalada**

   El mecanismo de atención escalada se define como:

   $$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

5. **Multi-Head Attention**

   La atención multi-cabezal se obtiene concatenando varias cabezas de atención:

   $$\text{MHA}(E) = \text{Concat}(head_1, \dots, head_h)W_O$$

6. **Red Feed-Forward**

   Cada capa Transformer incluye una red feed-forward completamente conectada:

   $$\text{FFN}(x) = W_2\,\text{ReLU}(W_1 x + b_1) + b_2$$

7. **Representación del token [CLS]**

Tras pasar por todas las capas del encoder, se obtiene el vector correspondiente al token especial [CLS]:

$$h_{[CLS]}$$

8. **Capa de clasificación**

   El vector $h_{[CLS]}$ se utiliza para la clasificación binaria mediante una capa densa con función *softmax*:

$$\hat{y} = \text{softmax}\left(W_{cls} h_{[CLS]} + b_{cls}\right)$$

9. **Función de pérdida**

   La función de pérdida utilizada durante el entrenamiento es la entropía cruzada categórica:

$$\mathcal{L} = -\sum_{i=1}^{2} y_i \log(\hat{y}_i)$$

ALBETO:

ALBETO es un modelo de lenguaje basado en la arquitectura Transformer, derivado de BERT y entrenado específicamente para el idioma español. Al igual que BERT y BETO, ALBETO emplea un enfoque bidireccional profundo que le permite analizar simultáneamente el contexto previo y posterior de cada palabra dentro de una secuencia. Esta característica resulta fundamental para comprender el significado completo de los textos, especialmente en escenarios donde el sentido de una palabra depende fuertemente de su contexto.

La principal particularidad de ALBETO radica en su entrenamiento sobre corpus especializados de discurso de odio y lenguaje ofensivo en español, lo que le permite capturar patrones lingüísticos complejos asociados a expresiones de violencia verbal, discriminación o contenido dañino. Debido a esta especialización, ALBETO resulta especialmente adecuado para tareas de clasificación de texto relacionadas con contenido sensible, incluyendo la detección de desinformación y noticias falsas que emplean lenguaje polarizante o alarmista (Cañete, Donoso, Bravo, Carvallo, & Araujo, 2022) .

En esta investigación, ALBETO se emplea como un modelo de clasificación de secuencias mediante un proceso de ajuste fino (fine-tuning), utilizando una capa

de clasificación binaria sobre la representación semántica global del texto, obtenida a partir del token especial [CLS].

Modelo Matemático

1. **Entrada del Modelo**

   Sea una secuencia de texto representada como tokens:

   $$X = [x_1, x_2, \dots, x_T]$$

2. **Representaciones Iniciales (Embeddings)**

   Cada token se representa como la suma de tres tipos de embeddings:

   $$E_i = E_i^{token} + E_i^{position} + E_i^{segment}$$

3. **Cálculo de Query, Key y Value**

   A partir de la matriz de embeddings E, se calculan las matrices necesarias para la autoatención:

   $$Q = EW_Q, \quad K = EW_K, \quad V = EW_V$$

4. **Self-Attention escalada**

   El mecanismo de autoatención escalada se define como:

   $$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

5. **Multi-Head Attention**

   La atención multi-cabezal combina múltiples cabezas de atención:

   $$\text{MHA}(E) = \text{Concat}(head_1, \dots, head_h)W_O$$

6. **Red Feed-Forward**

   Cada capa del encoder incluye una red feed-forward totalmente conectada:

   $$\text{FFN}(x) = W_2\,\text{ReLU}(W_1 x + b_1) + b_2$$

7. **Representación del token [CLS]**

   Tras pasar por todas las capas del encoder, se obtiene la representación final del token especial [CLS]:

   $$h_{[CLS]}$$

8. **Capa de clasificación**

   La clasificación binaria se realiza mediante una capa densa con función *softmax*:

   $$\hat{y} = \text{softmax}\left(W_{cls} h_{[CLS]} + b_{cls}\right)$$

9. **Función de pérdida**

Durante el entrenamiento se utiliza la entropía cruzada categórica:

$$\mathcal{L} = -\sum_{i=1}^{2} y_i \log(\hat{y}_i)$$

DistilBERT:

DistilBERT es una versión compacta y optimizada de BERT que fue diseñada con el objetivo de reducir el tamaño y el costo computacional del modelo original, manteniendo un alto nivel de rendimiento en tareas de procesamiento de lenguaje natural. Este modelo se basa en la técnica de destilación del conocimiento, mediante la cual un modelo más pequeño (student) aprende a imitar el comportamiento de un modelo más grande, en este caso BERT (Ayoub, Yang, & Zhou, 2021)

A diferencia de BERT, DistilBERT elimina algunos componentes de la arquitectura original, como el módulo de Next Sentence Prediction, y reduce el número de capas Transformer, lo que permite acelerar el proceso de inferencia y disminuir el consumo de memoria. A pesar de esta simplificación, el modelo conserva la capacidad de generar representaciones contextuales profundas del texto, manteniendo aproximadamente el 97 % del rendimiento de BERT en diversas tareas, con una reducción significativa del tamaño y del tiempo de cómputo.

En esta investigación, DistilBERT se utiliza como un modelo de clasificación de secuencias mediante un proceso de ajuste fino (fine-tuning), empleando la arquitectura DistilBertForSequenceClassification con dos etiquetas de salida. Esta elección permite evaluar el desempeño de modelos ligeros en la detección de noticias falsas relacionadas con el COVID-19, especialmente en escenarios donde los recursos computacionales son limitados.

Modelo Matemático

1. **Entrada del modelo**

   Sea una secuencia de texto representada como tokens:

   $$X = [x_1, x_2, \dots, x_T]$$

2. **Representaciones iniciales (Embeddings)**

Cada token se representa mediante la suma de los embeddings de token y posición (DistilBERT no utiliza embeddings de segmento):

$$E_i = E_i^{token} + E_i^{position}$$

3. **Cálculo de Query, Key y Value**

   A partir de la matriz de embeddings E, se calculan las matrices:

$$Q = EW_Q, \quad K = EW_K, \quad V = EW_V$$

4. **Self-Attention escalada**

   El mecanismo de autoatención se define como:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

5. **Multi-Head Attention**

   La atención multi-cabezal se obtiene concatenando múltiples cabezas de atención:

$$\text{MHA}(E) = \text{Concat}(head_1, \ldots, head_h)W_O$$

6. **Red Feed-Forward**

   Cada capa Transformer incluye una red feed-forward completamente conectada:

$$\text{FFN}(x) = W_2\,\text{ReLU}(W_1 x + b_1) + b_2$$

7. **Representación global del texto**

   La representación global del texto se obtiene a partir del token inicial (equivalente al rol del [CLS] en BERT):

$$h$$

8. **Capa de clasificación**

   La clasificación binaria se realiza mediante una capa densa con función *softmax*:

$$\hat{y} = \text{softmax}(W_{cls}h + b_{cls})$$

9. **Función de pérdida**

   Durante el entrenamiento se utiliza la entropía cruzada categórica:

$$\mathcal{L} = -\sum_{i=1}^{2} y_i \log(\hat{y}_i)$$

GRU:

El modelo GRU (Gated Recurrent Unit) es una arquitectura de red neuronal recurrente diseñada para modelar secuencias de datos y capturar dependencias

temporales a largo plazo. Fue propuesto como una alternativa más simple y eficiente a las redes LSTM, manteniendo un desempeño comparable, pero con una menor complejidad estructural (Chung, Gulcehre, Cho, & Bengio, 2014).

A diferencia de las LSTM, las GRU utilizan únicamente dos compuertas principales la compuerta de actualización (update gate) y la compuerta de reinicio (reset gate) lo que reduce el número de parámetros del modelo y acelera el proceso de entrenamiento. Este diseño permite mitigar el problema del desvanecimiento del gradiente y facilita el aprendizaje de dependencias a largo plazo en secuencias de texto.

En esta investigación, el modelo GRU se emplea para la clasificación binaria de noticias falsas relacionadas con el COVID-19. Su capacidad para procesar secuencias de palabras y capturar información contextual lo convierte en una opción adecuada para comparar modelos recurrentes frente a arquitecturas más complejas como los Transformers.

Modelo Matemático

1. **Entrada secuencial**

   Sea una secuencia de entrada:

   $$X = [x_1, x_2, \dots, x_T]$$

   donde x_t representa el vector de entrada en el instante t.

2. **Compuerta de actualización (Update Gate)**

   La compuerta de actualización controla cuánto del estado anterior se conserva:

   $$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

3. **Compuerta de reinicio (Reset Gate)**

   La compuerta de reinicio determina qué información pasada se descarta:

   $$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

4. **Estado candidato**

   El estado candidato combina la nueva entrada con la información previa filtrada:

   $$\widetilde{h_t} = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$$

5. **Estado oculto final**

El estado oculto final se actualiza combinando el estado previo y el candidato:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t}$$

6. **Salida del modelo**

   En tu arquitectura, la salida final de la última capa GRU se pasa a una capa densa con activación sigmoide:

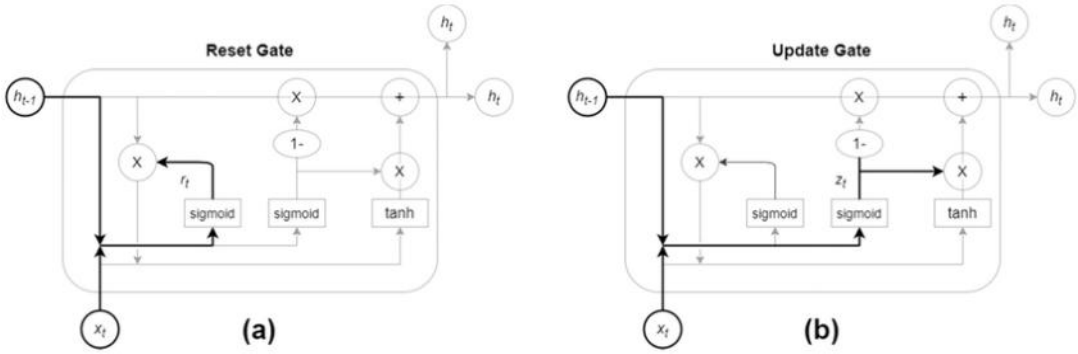
$$\hat{y} = \sigma(W h_T + b)$$

7. **Función de pérdida**

   El modelo se entrena utilizando entropía cruzada binaria:

$$\mathcal{L} = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

Ilustración 3: Funcionamiento interno de una unidad GRU mostrando las puertas de reinicio (reset gate) y actualización (update gate).



LSTM:

El modelo LSTM (Long Short-Term Memory) es una arquitectura de red neuronal recurrente diseñada para modelar secuencias de datos y superar las limitaciones de las RNN tradicionales, en particular los problemas de desvanecimiento y explosión del gradiente. Fue propuesto por Hochreiter y Schmidhuber como una solución capaz de aprender dependencias a largo plazo mediante el uso de una celda de memoria interna controlada por compuertas (Hochreiter & Schmidhuber).

A diferencia de otras redes recurrentes, las LSTM incorporan tres compuertas fundamentales: la compuerta de olvido, la compuerta de entrada y la compuerta de salida. Estas compuertas regulan el flujo de información dentro de la celda de memoria, permitiendo decidir qué información debe conservarse, actualizarse o descartarse en cada paso temporal. Gracias a esta estructura, las

LSTM han demostrado un alto desempeño en tareas de procesamiento de lenguaje natural, reconocimiento de voz y análisis de series temporales.

En esta investigación, el modelo LSTM se utiliza para la clasificación binaria de noticias falsas relacionadas con el COVID-19, aprovechando su capacidad para capturar dependencias semánticas de largo alcance en secuencias de texto. El modelo implementado emplea dos capas LSTM con regularización mediante dropout, seguidas de una capa densa con activación sigmoide para la predicción final.

Modelo Matemático

1. **Entrada secuencial**

   Sea una secuencia de entrada:

   $$X = [x_1, x_2, \dots, x_T]$$

   donde x_t es el vector de entrada en el tiempo t.

2. **Compuerta de olvido**

   La compuerta de olvido determina qué información del estado de memoria previo debe descartarse:

   $$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right)$$

3. **Compuerta de entrada**

   La compuerta de entrada controla la información nueva que se incorporará a la memoria:

   $$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

4. **Estado candidato de la celda**

   El estado candidato representa nueva información potencial a almacenar:

   $$\widetilde{C_t} = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

5. **Actualización del estado de la celda**

   El estado de memoria se actualiza combinando la información previa y la nueva:

   $$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C_t}$$

6. **Compuerta de salida**

   La compuerta de salida controla la información que se expone como salida:

   $$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

7. **Estado oculto final**

El estado oculto final se obtiene aplicando la compuerta de salida al estado de la celda:

$$h_t = o_t \odot \tanh(C_t)$$

8. **Capa de clasificación**

   En la arquitectura utilizada, la salida final de la última capa LSTM se introduce en una capa densa con activación sigmoide:

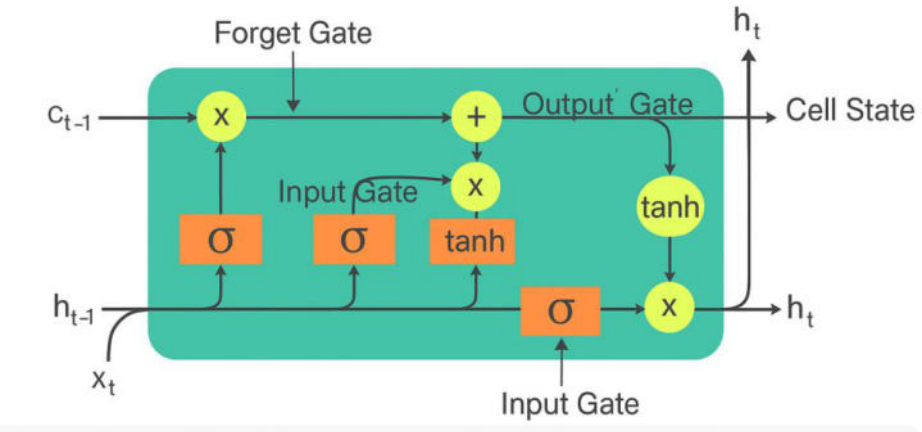
$$\hat{y} = \sigma(W h_T + b)$$

9. **Función de pérdida**

   El entrenamiento del modelo se realiza mediante entropía cruzada binaria:

$$\mathcal{L} = -\left[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})\right]$$

Ilustración 4: Estructura interna de una celda LSTM



BiLSTM:

El modelo Bidirectional LSTM (BiLSTM) es una extensión de las redes LSTM tradicionales que permite procesar secuencias de datos en ambas direcciones temporales, es decir, desde el inicio hasta el final de la secuencia y desde el final hacia el inicio. Esta arquitectura fue propuesta con el objetivo de capturar información contextual completa, considerando tanto el contexto pasado como el futuro de cada elemento de la secuencia (Sherstinsky, 2020).

A diferencia de una LSTM unidireccional, que solo utiliza información previa, una BiLSTM combina dos redes LSTM independientes: una que procesa la secuencia de manera directa (forward) y otra que la procesa en sentido inverso (backward). Las salidas de ambas redes se concatenan, permitiendo obtener representaciones más ricas y completas del texto.

En tareas de procesamiento de lenguaje natural, esta característica resulta especialmente relevante, ya que el significado de una palabra puede depender tanto de las palabras que la preceden como de las que la siguen. En esta investigación, el modelo BiLSTM se emplea para la clasificación binaria de noticias falsas relacionadas con el COVID-19, utilizando dos capas BiLSTM seguidas de una capa densa con activación sigmoide.

Modelo Matemático

1. **Entrada secuencial**

   Sea una secuencia de entrada:

   $$X = [x_1, x_2, \dots, x_T]$$

   donde $x_t$representa el vector de entrada en el instante t.

2. **LSTM hacia adelante (Forward LSTM)**

   La LSTM directa procesa la secuencia desde $t = 1$hasta $t = T$:

   $$\vec{h} = \text{LSTM}_f\left(x_t, \overrightarrow{h_{t-1}}\right)$$

3. **LSTM hacia atrás (Backward LSTM)**

   La LSTM inversa procesa la secuencia desde $t = T$hasta $t = 1$:

   $$\overleftarrow{h_t} = \text{LSTM}_b\left(x_t, \overleftarrow{h_{t-1}}\right)$$

4. **Concatenación de estados ocultos**

   Las salidas de ambas direcciones se combinan mediante concatenación:

   $$h_t = \left[\overrightarrow{h_t}\,;\, \overleftarrow{h_t}\right]$$

5. **Representación final de la secuencia**

   Para tareas de clasificación, se utiliza el estado oculto final de la segunda capa BiLSTM:

   $$h$$

6. **Capa de clasificación**

   La salida final se obtiene mediante una capa densa con activación sigmoide:

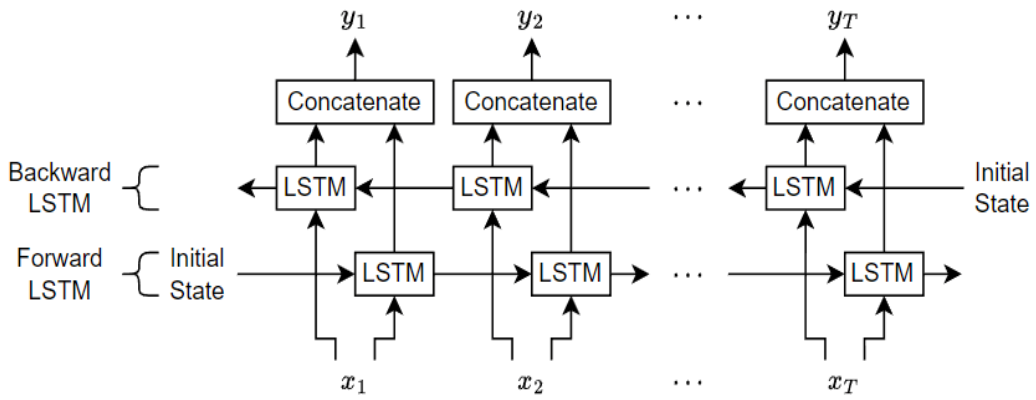
   $$\hat{y} = \sigma(Wh + b)$$

7. **Función de pérdida**

   El entrenamiento del modelo se realiza utilizando entropía cruzada binaria:

   $$\mathcal{L} = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

Ilustración 5: Arquitectura de una red Bidirectional LSTM (BiLSTM) para el procesamiento de secuencias de texto.



Variables de la Investigación:

El análisis de variables permite comprender los elementos que intervienen en el estudio y la forma en que se relacionan entre sí. En esta investigación se distinguen dos tipos de variables: una **variable independiente**, vinculada a las técnicas de aprendizaje profundo aplicadas al procesamiento de textos, y una **variable dependiente**, relacionada con el nivel de rendimiento que alcanzan dichos modelos al clasificar noticias como verdaderas o falsas.

Variable Independiente

La variable independiente de la investigación está constituida por las técnicas y modelos de aprendizaje profundo utilizados para el procesamiento y análisis del texto. Estas técnicas definen la forma en que la información lingüística es representada internamente y cómo el sistema es capaz de aprender patrones semánticos y contextuales a partir de los datos.

En el ámbito del procesamiento de lenguaje natural (PLN), esta variable comprende diversas arquitecturas de redes neuronales, entre las que se incluyen BERT, BETO, DistilBERT, ALBETO, LSTM, GRU, BiLSTM y CNN. Cada uno de estos modelos presenta mecanismos de aprendizaje distintos para capturar la estructura del lenguaje, ya sea mediante atención bidireccional, compuertas recurrentes o filtros convolucionales.

La relevancia de esta variable radica en que la elección del modelo de aprendizaje profundo influye directamente en la calidad de las representaciones textuales generadas, lo cual impacta en la capacidad del sistema para diferenciar entre noticias verdaderas y falsas. Por tanto, las variaciones en el tipo de arquitectura empleada producen cambios observables en el desempeño del proceso de clasificación.

Variable Dependiente

La variable dependiente corresponde al desempeño del modelo de clasificación, entendido como la capacidad del sistema para identificar correctamente noticias falsas y verdaderas a partir de textos relacionados con el COVID-19.

El desempeño de los modelos se evalúa mediante métricas cuantitativas ampliamente utilizadas en tareas de clasificación binaria, las cuales permiten medir de forma objetiva la eficacia del sistema. Entre estas métricas se consideran la exactitud (accuracy), la precisión (precision), la exhaustividad (recall) y el puntaje F1 (F1-score).

Esta variable depende directamente de la variable independiente, dado que las distintas arquitecturas de aprendizaje profundo generan diferentes niveles de rendimiento, afectando la capacidad del modelo para generalizar y realizar predicciones correctas sobre datos no vistos durante el entrenamiento.

IV. Metodología

Tipo de investigación

Tecnológico – Aplicativo

La búsqueda de conocimiento en el ámbito tecnológico, según (De La Cruz Casaño, 2016), implica explorar diversas fuentes en busca de información que pueda ser relevante para abordar problemas específicos. Por otro lado, (Sampieri & Collado, 2006) definen la investigación aplicada como aquella que se concentra en resolver problemas concretos que enfrentan la sociedad o las empresas en su día a día.

Diseño Experimental

El diseño se conoce como pre-experimental: X O

Donde X es la aplicación del modelo de clasificación y O es la observación y evaluación de los resultados

Operacionalización de variables

Tabla 1: Operacionalización de Variables

Variables	Dimensiones	Indicadores	Escala de Medición
Variable Independiente: Aumento de Datos y Deep Learning	Deep learning DNN GRU LSTM BERT BILSTM BETO ALBETO	Hidden Layers Hidden Neurons Optimizer Lost Function Learning Rate Activation Functions Architecture	[1-4] [10-100] Adam Binary Cross entropy [0.001-0.01] Relu [20,30,30,20,1]
	Prueba	Diagnóstico	Negativo [0-1]

Variable Dependiente: Predicción de noticias falsas sobre COVID-19	Evaluación	Accuracy Recall Precision F1-Score	[0-1] [0-1] [0-1] [0-1]

Nota: La tabla muestra la operacionalización de variables para el estudio, indicando los diferentes modelos y parámetros de Deep Learning empleados, así como las métricas de evaluación utilizadas en la detección de noticias falsas sobre COVID-19.

Población y Muestra

Población: Noticias publicadas en redes sociales

Muestra: 1500 noticias sobre covid-19

Técnicas e instrumentos para la recolección de datos

Técnicas: Técnica de recuperación de información

Instrumentos: Para la recopilación de datos se utilizará web-scraping

Métodos y técnicas para la presentación y análisis de Datos

Procesamiento de los datos

Recolección de información:

Web Scraping: Se realizó la recolección de datos a través de técnicas de Web Scraping en redes sociales como Twitter, Reddit y Facebook. El scraping se llevó a cabo utilizando bibliotecas como Selenium en Python, con el objetivo de extraer tanto textos completos de noticias como comentarios y reacciones de los usuarios.

API de Twitter: Para Twitter, se utilizó la API de Twitter para acceder a tweets relacionados con COVID-19. Se definieron consultas de búsqueda

específicas que incluían palabras clave y hashtags relacionados con noticias sobre la pandemia.

Preparación de los datos: La preparación de los datos fue una etapa crítica para garantizar que la información estuviera en un formato adecuado para ser procesada por los distintos modelos. Los datos recolectados originalmente se almacenaron en archivos CSV con varias columnas relevantes, como el texto de la noticia, la fuente y la clase (falsa o verdadera). En esta etapa se realizó la limpieza y preprocesamiento de los datos.

Carga y estructuración de los datos: Se cargaron los archivos CSV y, posteriormente, se dividió la columna combinada ('new;class') en dos columnas separadas: una para el texto de la noticia (new) y otra para la clase de la noticia (class), que indica si la noticia es verdadera o falsa.

Conversión de las clases: La columna que indica la clase de la noticia fue convertida a formato numérico. En este caso, las clases 0 y 1 representan noticias falsas y verdaderas, respectivamente.

Eliminación de datos faltantes: Las filas con valores nulos o inválidos fueron eliminadas para asegurar que solo se utilicen datos completos y coherentes.

Etiquetado de Datos

El proceso de etiquetado de datos es un paso fundamental en la construcción de un conjunto de datos de entrenamiento confiable. En este caso, el etiquetado fue realizado manualmente, lo que permite una mayor precisión en la clasificación de las noticias. Se propuso que un equipo de expertos o revisores identificara y clasificara cada noticia como falsa o verdadera, utilizando criterios específicos.

El etiquetado se realizó bajo el siguiente esquema:

- Clase 0: Noticias clasificadas como falsas. Estas noticias contenían información incorrecta, engañosa o no verificable sobre el COVID-19.

- Clase 1: Noticias clasificadas como verdaderas. En esta clase se incluyeron noticias con información verificada, correcta y proveniente de fuentes confiables.

Este proceso de etiquetado manual se eligió porque, a pesar de ser más costoso en términos de tiempo, garantiza un alto nivel de calidad en los datos, lo cual es esencial para el entrenamiento de los distintos modelos propuestos.

División de Datos

Para garantizar una correcta evaluación del modelo, los datos se dividieron en tres conjuntos:

- Conjunto de entrenamiento (70%): Se utilizó para ajustar el modelo BERT. El modelo aprende las características del texto que indican si una noticia es verdadera o falsa.
- Conjunto de validación (10%): Sirve para ajustar hiperparámetros y evaluar el rendimiento del modelo durante el proceso de entrenamiento. De esta manera, se evita el sobreajuste.
- Conjunto de prueba (20%): Este conjunto se utilizó exclusivamente para evaluar el rendimiento final del modelo después del entrenamiento. De este modo, se obtiene una estimación precisa de cómo el modelo generaliza a datos no vistos previamente.

La división de los datos se realizó utilizando la función train_test_split de la biblioteca sklearn, asegurando que cada conjunto contuviera una distribución balanceada de noticias falsas y verdaderas.

Modelos Utilizados

BERT

Descripción del Modelo

El modelo utilizado en esta investigación es BERT (Bidirectional Encoder Representations from Transformers), una arquitectura de redes neuronales basada en Transformers que ha sobresalido en tareas de procesamiento de lenguaje natural

(NLP) gracias a su habilidad para interpretar el contexto en ambas direcciones dentro de una secuencia de palabras. A diferencia de modelos anteriores, BERT no solo lee el texto de izquierda a derecha (como en los modelos secuenciales tradicionales), sino que también lo hace de derecha a izquierda, lo que le permite capturar relaciones contextuales complejas entre palabras en una oración (Devlin, Chang, Lee, & Toutanova, 2019).

En este trabajo se utilizó la variante en español, "dccuchile/bert-base-spanish-wwm-uncased", que fue preentrenada en un gran corpus de texto en español. Este modelo permite la detección precisa de noticias falsas en el contexto del idioma español, ya que está entrenado para capturar matices lingüísticos específicos del idioma, lo que lo convierte en una herramienta efectiva para esta tarea.

El modelo está compuesto por múltiples capas de atención, donde cada capa aprende representaciones jerárquicas de los datos de entrada, lo que lo hace particularmente eficaz en tareas de clasificación de texto como la que se aborda en esta investigación.

Modelamiento

El modelo utilizado para esta tarea fue **BERT para la clasificación de secuencias**, con el fin de aprovechar su capacidad para comprender tanto el contexto individual de cada palabra como el significado global del texto. Se empleó el modelo preentrenado **"dccuchile/bert-base-spanish-wwm-uncased"**, especializado en el idioma español.

El procedimiento de modelamiento incluyó los siguientes pasos:

- **Tokenización:** Se utilizó el tokenizador específico de BERT para dividir las secuencias de texto en los tokens requeridos por el modelo. Esta etapa es crucial para garantizar que los datos se presenten en el formato adecuado.

- **Capa de embeddings:** Cada token fue representado por un vector de embeddings que BERT utiliza para capturar el significado contextual de las palabras en el texto.
- **Clasificación:** El modelo añadió una capa de clasificación al final para predecir si una noticia es falsa o verdadera. Esta capa fue ajustada para predecir dos clases (falsa o verdadera), utilizando una función de pérdida de entropía cruzada binaria y una optimización con el algoritmo **Adam**.

El entrenamiento se realizó utilizando la librería **Hugging Face's Transformers**, lo cual facilitó la implementación del modelo preentrenado. Los parámetros de entrenamiento incluyeron un tamaño de lote de 8 y un esquema de optimización con **Adam**, con un número de 3 épocas de entrenamiento. También se establecieron 500 pasos de calentamiento (**warmup steps**) y un decaimiento de peso de 0.01 para prevenir el sobreajuste.

Evaluación

Para evaluar el rendimiento del modelo, se utilizaron las siguientes métricas:

- **Accuracy:** Indica la proporción de noticias correctamente clasificadas.
- **Precisión:** Representa la proporción de noticias clasificadas como falsas que efectivamente eran falsas.
- **Recall:** Evalúa la habilidad del modelo para reconocer correctamente todas las instancias positivas.
- **F1-Score:** Es una combinación de la precisión y el recall, ofreciendo una medida más balanceada del rendimiento del modelo.

Estas métricas se calcularon utilizando la función trainer.predict() para obtener las predicciones del modelo sobre el conjunto de prueba.

BETO

Descripción del Modelo

El modelo BETO es una variante de BERT, específicamente entrenado para el idioma español. Fue desarrollado para abordar las limitaciones de los modelos preentrenados en inglés al aplicarlos en tareas de procesamiento de lenguaje natural en español. BETO sigue la misma arquitectura de BERT, aprovechando los principios de Transformers, pero con el beneficio adicional de haber sido preentrenado en un corpus extenso en español. Esto lo hace particularmente útil en tareas de clasificación de texto en español, como la detección de noticias falsas (Sergio Damian, Hiram, & Gelbukh, 2020).

Al igual que BERT, BETO utiliza un mecanismo de atención que le permite aprender relaciones contextuales bidireccionales, es decir, comprende tanto el contexto anterior como el posterior de cada palabra en una oración. Esto es crucial en el idioma español, donde la estructura gramatical y el orden de las palabras pueden variar significativamente en comparación con el inglés.

Modelamiento

El modelo utilizado para esta tarea fue **BETO** para la clasificación de secuencias, con el objetivo de aprovechar su capacidad para comprender tanto el contexto individual de cada palabra como el significado global del texto en español. Se empleó el modelo preentrenado **"dccuchile/bert-base-spanish-wwm-uncased"**, especializado en el idioma español.

El procedimiento de modelamiento incluyó los siguientes pasos:

Tokenización: Se utilizó el tokenizador específico de BETO para dividir las secuencias de texto en los tokens requeridos por el modelo. Esta etapa es crucial para garantizar que los datos se presenten en el formato adecuado, preservando el contexto de las palabras en el idioma español.

Capa de embeddings: Cada token fue representado por un vector de embeddings que BETO utiliza para capturar el significado contextual de las

palabras en el texto. Estos embeddings ayudan al modelo a aprender las relaciones entre las palabras, facilitando la comprensión del texto.

Clasificación: El modelo añadió una capa de clasificación al final para predecir si una noticia es falsa o verdadera. Esta capa fue ajustada para predecir dos clases (falsa o verdadera), utilizando una función de pérdida de **entropía cruzada binaria** y una optimización con el algoritmo **Adam**.

El entrenamiento se realizó utilizando la librería **Hugging Face's Transformers**, lo cual facilitó la implementación del modelo preentrenado. Los parámetros de entrenamiento incluyeron un tamaño de lote de 8 y un esquema de optimización con **Adam**, con un número de **3 épocas de entrenamiento**. También se establecieron **500 pasos de calentamiento** (warmup steps) y un **decaimiento de peso de 0.01** para prevenir el sobreajuste, asegurando que el modelo generalice correctamente a datos no vistos previamente

Evaluación

Para evaluar el rendimiento del modelo, se utilizaron las siguientes métricas:

- **Accuracy:** Indica la proporción de noticias correctamente clasificadas.
- **Precisión:** Representa la proporción de noticias clasificadas como falsas que efectivamente eran falsas.
- **Recall:** Evalúa la habilidad del modelo para reconocer correctamente todas las instancias positivas.
- **F1-Score:** Es una combinación de la precisión y el recall, ofreciendo una medida más balanceada del rendimiento del modelo.

ALBETO

Descripción del modelo

El modelo ALBETO (Cañete, Donoso, Bravo, Carvallo, & Araujo, 2022) es una variante de BERT preentrenada para la detección de discurso de odio en español, utilizando el modelo "GalMargo/DistiBERTHateSpeech". ALBETO fue

diseñado para identificar patrones lingüísticos que se asocian con el discurso de odio, discriminación y violencia verbal, lo que lo hace una herramienta valiosa en la moderación de contenido, análisis de redes sociales y aplicaciones que requieren la detección de lenguaje inapropiado.

Al igual que BERT, ALBETO utiliza una arquitectura de Transformers que le permite comprender el contexto bidireccional de las palabras en un texto. Esto significa que no solo toma en cuenta el significado de una palabra, sino también su relación con las palabras que la rodean en una oración, permitiendo una mejor detección de patrones complejos de odio o violencia verbal. Este modelo es de particular interés debido a su capacidad para capturar las sutilezas del lenguaje en español y su entrenamiento en un corpus específico de discurso de odio.

Modelamiento

El modelo **ALBETO** fue configurado para la tarea de clasificación de secuencias con el objetivo de identificar textos con discurso de odio. Para esto, se utilizó el **tokenizador específico de DistiBERT**, que se encargó de dividir las secuencias de texto en tokens adecuados para ser procesados por el modelo.

El proceso de modelamiento siguió los siguientes pasos:

- **Tokenización:** Las secuencias de texto se tokenizaron utilizando el tokenizador de **DistiBERT**, garantizando que cada palabra o subpalabra se representara como un token adecuado para el modelo.
- **Capa de embeddings:** Cada token fue convertido en un vector de embeddings que captura el significado contextual de la palabra en la secuencia.
- **Clasificación:** El modelo añadió una capa de clasificación para predecir si una noticia contiene discurso de odio (1) o no (0), utilizando una función de pérdida de entropía cruzada binaria y el algoritmo de optimización **Adam**.

El modelo se entrenó utilizando la librería **Hugging Face's Transformers**, con un tamaño de lote de 8 y un esquema de optimización de **Adam**. Se ejecutaron

un total de **3 épocas** de entrenamiento, y se configuraron **500 pasos de calentamiento** para mejorar la estabilidad del entrenamiento. Además, se aplicó un **decaimiento de peso de 0.01** para prevenir el sobreajuste.

Evaluación

Para evaluar el rendimiento del modelo **ALBETO**, se utilizaron las siguientes métricas estándar:

- **Accuracy:** Representa la proporción de ejemplos clasificados correctamente.
- **Precisión:** Mide la proporción de casos identificados como discurso de odio que realmente contienen discurso de odio.
- **Recall:** Indica la capacidad del modelo para detectar todas las instancias de discurso de odio en los datos.
- **F1-Score:** Es la media armónica entre la precisión y el recall, proporcionando una medida más balanceada del rendimiento del modelo.

Estas métricas fueron calculadas utilizando el conjunto de prueba, lo que permitió una evaluación precisa del modelo en un entorno de datos no vistos durante el entrenamiento.

DistilBERT

Descripción del Modelo

DistilBERT (Ayoub, Yang, & Zhou, 2021) es una versión compacta y optimizada de BERT que mantiene aproximadamente el 97% del rendimiento de BERT en varias tareas de procesamiento de lenguaje natural, mientras que es un 40% más pequeño y un 60% más rápido. DistilBERT utiliza un enfoque de "distilación del conocimiento" para comprimir el modelo original sin perder su capacidad de comprender el contexto bidireccional del texto. Este modelo es particularmente útil en aplicaciones donde los recursos computacionales son limitados, como dispositivos móviles o entornos con restricciones de tiempo.

Al igual que BERT, DistilBERT se basa en la arquitectura de Transformers, que utiliza mecanismos de autoatención para procesar secuencias de texto. Sin embargo, en lugar de las 12 capas de atención en BERT, DistilBERT reduce el número de capas a 6, lo que permite un procesamiento más eficiente sin comprometer significativamente el rendimiento. Este modelo fue entrenado en un corpus grande en inglés y es especialmente útil para tareas de clasificación de secuencias

Modelamiento

El modelo **DistilBERT** fue implementado para la tarea de **clasificación de secuencias**, donde se busca clasificar los textos en dos clases: discurso de odio o no. El modelo fue preentrenado con el corpus **"distilbert-base-uncased"**.

El procedimiento de modelamiento incluyó los siguientes pasos:

- **Tokenización:** Se utilizó el tokenizador específico de **DistilBERT** para dividir las secuencias de texto en tokens adecuados, asegurando que cada palabra o subpalabra estuviera correctamente representada en un formato que el modelo pudiera procesar.
- **Capa de embeddings:** Cada token se convirtió en un vector de embeddings que captura el significado contextual de la palabra en la secuencia de texto.
- **Clasificación:** El modelo añadió una capa de clasificación al final, configurada para predecir dos clases: con discurso de odio (1) o sin discurso de odio (0). Esta capa se entrenó utilizando una función de pérdida de **entropía cruzada binaria** y un esquema de optimización con **Adam**.

El modelo fue entrenado utilizando la librería **Hugging Face's Transformers**, con un esquema de entrenamiento que incluía un tamaño de lote de 8 y un total de **3 épocas de entrenamiento**. Además, se configuraron **500 pasos de calentamiento** para mejorar la estabilidad del entrenamiento y se aplicó un **decaimiento de peso de 0.01** para prevenir el sobreajuste.

Evaluación

Para evaluar el rendimiento del modelo **DistilBERT**, se utilizaron las siguientes métricas:

- **Accuracy:** Indica la proporción de textos correctamente clasificados.
- **Precisión:** Mide cuántos de los textos clasificados como discurso de odio realmente lo son.
- **Recall:** Evalúa la capacidad del modelo para identificar correctamente todos los textos con discurso de odio.
- **F1-Score:** Es la media armónica entre la precisión y el recall, proporcionando una medida más equilibrada del rendimiento del modelo.

Estas métricas se calcularon utilizando el conjunto de prueba para obtener una evaluación precisa del rendimiento del modelo en datos no vistos durante el entrenamiento.

GRU

Descripción del Modelo

El modelo GRU (Gated Recurrent Unit) es un tipo de red neuronal recurrente (RNN) diseñado para procesar secuencias de datos. Las GRU son una variante más eficiente de las LSTM (Long Short-Term Memory) y se utilizan principalmente para tareas relacionadas con el procesamiento de series temporales o secuencias de texto, como la clasificación de texto. Las GRU ayudan a mitigar el problema de desvanecimiento de gradientes al capturar dependencias a largo plazo en los datos sin la complejidad añadida de las LSTM, lo que las hace más rápidas y eficientes en términos de recursos computacionales (Chung, Gulcehre, Cho, & Bengio, 2014).

En este modelo, se utiliza una capa de embedding para convertir las palabras en vectores densos que representan sus significados, seguida de dos capas GRU que

permiten capturar las dependencias temporales dentro del texto. Estas capas ayudan al modelo a aprender las relaciones entre las palabras en secuencias de texto largas, lo que es crucial para tareas como la clasificación de noticias falsas.

Modelamiento

El modelo GRU fue implementado para la tarea de **clasificación de secuencias**. Se utilizó una **capa de embedding** para mapear cada palabra a un vector de características, y dos capas **GRU** para capturar las dependencias temporales entre las palabras en cada secuencia.

El procedimiento de modelamiento incluyó los siguientes pasos:

- **Tokenización y secuencias:** Las secuencias de texto se dividieron en tokens y luego se convirtieron en secuencias numéricas utilizando un vocabulario de tamaño 20,000. Posteriormente, estas secuencias se completaron con ceros al final para que todas tuvieran una longitud uniforme.
- **Capa de embeddings:** Cada palabra en el texto fue convertida en un vector de características utilizando una capa de embedding con un tamaño de salida de 128.
- **Capa GRU:** Se añadieron dos capas GRU para aprender las dependencias a lo largo de las secuencias de texto. La primera capa GRU tiene 128 unidades y devuelve secuencias para alimentar a la siguiente capa GRU de 64 unidades.
- **Dropout:** Se añadieron capas de **Dropout** para reducir el riesgo de sobreajuste durante el entrenamiento.
- **Clasificación:** Finalmente, se añadió una capa densa con una unidad de salida y una función de activación **sigmoide** para realizar la clasificación binaria (noticias falsas o verdaderas).

El modelo se entrenó utilizando el optimizador **Adam** con una tasa de aprendizaje de 0.0001 y una función de pérdida de **entropía cruzada binaria**. El

entrenamiento se realizó durante **20 épocas** con un tamaño de lote de 32, utilizando datos de validación para ajustar el rendimiento del modelo.

Evaluación

Para evaluar el rendimiento del modelo **GRU**, se utilizaron las siguientes métricas estándar:

- **Accuracy:** Mide la proporción de ejemplos correctamente clasificados.
- **Precisión:** Representa la proporción de ejemplos clasificados como verdaderos que efectivamente lo son.
- **Recall:** Evalúa la habilidad del modelo para reconocer correctamente todas las instancias positivas.
- **F1-Score:** Es la media armónica entre la precisión y el recall, proporcionando una medida equilibrada del rendimiento del modelo.

Estas métricas se calcularon utilizando el conjunto de prueba para evaluar el rendimiento del modelo en datos no vistos durante el entrenamiento.

LSTM

Descripción del Modelo

El modelo LSTM (Long Short-Term Memory) es una arquitectura de red neuronal recurrente (RNN) diseñada para manejar secuencias largas de datos y resolver problemas asociados con las redes neuronales recurrentes tradicionales, como el desvanecimiento y explosión del gradiente. Las LSTM se utilizan ampliamente en tareas de procesamiento de lenguaje natural, reconocimiento de voz y series temporales debido a su capacidad para capturar dependencias a largo plazo en los datos (Hochreiter & Schmidhuber).

Cada unidad LSTM contiene tres "puertas" fundamentales: la puerta de entrada, la puerta de olvido y la puerta de salida, que controlan el flujo de información a través de la celda de memoria. Estas puertas regulan qué información debe ser actualizada, retenida o descartada en cada paso temporal, lo que permite a

las LSTM aprender patrones complejos en secuencias largas de datos. Esta capacidad para mantener y actualizar la información relevante a lo largo de múltiples pasos temporales hace que las LSTM sean especialmente útiles para tareas como la predicción de series temporales y la comprensión del lenguaje natural (Staudemeyer & Rothstein Morris, 2019).

Modelamiento

El modelo **LSTM** se configuró para la tarea de **clasificación de secuencias**, en la cual se busca clasificar textos en dos categorías: noticia falsa o verdadera.

El procedimiento de modelamiento incluyó los siguientes pasos:

- **Tokenización y secuencias:** Los textos fueron tokenizados y convertidos en secuencias numéricas utilizando un vocabulario de tamaño 20,000. Las secuencias se completaron para asegurar que todas tuvieran una longitud uniforme de 200 tokens.
- **Capa de embeddings:** Cada palabra se transformó en un vector de características utilizando una capa de embedding con un tamaño de salida de 128. Esto permite al modelo aprender representaciones densas de las palabras y sus contextos en el texto.
- **Capas LSTM:** Se añadieron dos capas LSTM. La primera capa tiene 128 unidades y devuelve secuencias completas para la siguiente capa LSTM, que tiene 64 unidades. Estas capas permiten al modelo capturar patrones y dependencias a lo largo de las secuencias de texto.
- **Dropout:** Se incluyeron capas de **Dropout** para prevenir el sobreajuste, con una tasa de 0.5 después de cada capa LSTM.
- **Clasificación:** Finalmente, se añadió una capa densa con una unidad de salida y una función de activación **sigmoide** para realizar la clasificación binaria (noticia verdadera o falsa).

El modelo se entrenó utilizando el optimizador Adam con una tasa de aprendizaje de 0.0001 y una función de pérdida de entropía cruzada binaria. El

entrenamiento se realizó durante 20 épocas con un tamaño de lote de 32, utilizando un conjunto de validación para monitorear el rendimiento y ajustar los parámetros.

Evaluación

Para evaluar el rendimiento del modelo **LSTM**, se utilizaron las siguientes métricas:

- **Accuracy:** Representa la proporción de textos correctamente clasificados.
- **Precisión:** Mide cuántos de los textos clasificados como verdaderos realmente lo son.
- **Recall:** Evalúa la capacidad del modelo para identificar correctamente todas las instancias verdaderas.
- **F1-Score:** Es la media armónica entre la precisión y el recall, proporcionando una medida equilibrada del rendimiento del modelo.

Estas métricas se calcularon utilizando el conjunto de prueba para evaluar el rendimiento del modelo en datos no vistos durante el entrenamiento.

BiLSTM

Descripción del Modelo

El modelo **Bidirectional LSTM (BiLSTM)** es una extensión de las redes neuronales LSTM (Long Short-Term Memory) que permite a la red aprender patrones en secuencias de datos tanto en direcciones hacia adelante como hacia atrás. A diferencia de las LSTM tradicionales, las BiLSTM consisten en dos capas LSTM, una que procesa la secuencia desde el inicio hasta el final y otra que lo hace en la dirección opuesta, permitiendo al modelo capturar contextos bidireccionales. Esto resulta especialmente útil en tareas de procesamiento de lenguaje natural, donde el significado de una palabra puede depender tanto de su contexto anterior como del posterior (Sherstinsky, 2020).

En este modelo, se utiliza una capa de **embedding** para convertir cada palabra en un vector de características, seguida de dos capas LSTM bidireccionales

que permiten al modelo comprender la información contextual de la secuencia en ambas direcciones. Este enfoque ayuda a mejorar la comprensión y clasificación del texto, especialmente en tareas como la detección de noticias falsas o la clasificación de emociones en textos.

Modelamiento

El modelo Bidirectional LSTM (BiLSTM) se implementó para la tarea de clasificación de secuencias, donde se busca clasificar textos en dos categorías: noticia falsa o verdadera.

El procedimiento de modelamiento incluyó los siguientes pasos:

- Tokenización y secuencias: Se tokenizaron los textos y se convirtieron en secuencias numéricas utilizando un vocabulario derivado del corpus de entrenamiento. Posteriormente, estas secuencias se completaron para asegurar que todas tuvieran una longitud uniforme de 100 tokens.
- Capa de embeddings: Cada palabra se transformó en un vector de características utilizando una capa de embedding con un tamaño de salida de 100. Esto permite al modelo aprender representaciones densas de las palabras y sus contextos en el texto.
- Capas BiLSTM: Se añadieron dos capas LSTM bidireccionales. La primera capa bidireccional tiene 64 unidades y devuelve secuencias completas para la siguiente capa, que también tiene 64 unidades. Estas capas permiten al modelo capturar patrones y dependencias en ambas direcciones temporales.
- Dropout: Se incluyeron capas de Dropout para prevenir el sobreajuste, con una tasa de 0.5 después de cada capa LSTM bidireccional.
- Clasificación: Finalmente, se añadió una capa densa con una unidad de salida y una función de activación sigmoide para realizar la clasificación binaria (noticia verdadera o falsa).

El modelo se entrenó utilizando el optimizador Adam con una tasa de aprendizaje de 0.0001 y una función de pérdida de entropía cruzada binaria. El entrenamiento se realizó durante 20 épocas con un tamaño de lote de 32, utilizando un conjunto de validación para monitorear el rendimiento y ajustar los parámetros.

Evaluación

Para evaluar el rendimiento del modelo **BiLSTM**, se utilizaron las siguientes métricas estándar:

- **Accuracy:** Representa la proporción de textos correctamente clasificados.
- **Precisión:** Mide cuántos de los textos clasificados como verdaderos realmente lo son.
- **Recall:** Evalúa la capacidad del modelo para identificar correctamente todas las instancias verdaderas.
- **F1-Score:** Es la media armónica entre la precisión y el recall, proporcionando una medida equilibrada del rendimiento del modelo.

Estas métricas se calcularon utilizando el conjunto de prueba para evaluar el rendimiento del modelo en datos no vistos durante el entrenamiento.

CNN

Descripción del Modelo

El modelo CNN (Convolutional Neural Network) es una arquitectura de redes neuronales comúnmente utilizada para tareas de procesamiento de imágenes, pero que también ha demostrado ser efectiva en el procesamiento de secuencias de texto. Las capas convolucionales de este modelo permiten capturar características locales de las secuencias, como n-gramas o patrones repetitivos, y combinarlas de manera eficiente en tareas de clasificación. En este contexto, se emplea una arquitectura de CNN de una sola dimensión (1D) para procesar secuencias de texto y detectar patrones asociados a noticias falsas en español sobre el COVID-19 (Kim, 2014).

Modelamiento

Para este modelo CNN, se realizaron los siguientes pasos de modelamiento:

Tokenización y Preprocesamiento:

- Se utilizó un tamaño de vocabulario de 20,000 palabras, y cada texto se convirtió en una secuencia de enteros correspondientes a las palabras en el vocabulario.
- Las secuencias fueron rellenadas (padding) para asegurar una longitud uniforme de 200 palabras, lo cual es esencial para que la red convolucional procese entradas de tamaño constante.

Arquitectura del Modelo:

- Capa de Embedding: Se inicializó con un tamaño de embedding de 128 dimensiones para representar cada palabra en un vector denso.
- Capa Conv1D: Una capa convolucional con 128 filtros y un tamaño de kernel de 5, que ayuda a capturar patrones locales en las secuencias.
- GlobalMaxPooling1D: Reduce la dimensionalidad tomando el valor máximo de cada filtro, resaltando las características más prominentes detectadas.
- Dropout: Se aplicó un Dropout de 0.5 en dos ocasiones para reducir el sobreajuste.
- Capas Densas: Una capa densa con 64 unidades y función de activación ReLU, seguida de una capa de salida con una unidad y activación sigmoide para clasificar entre noticias verdaderas y falsas.

Parámetros de Entrenamiento:

- Función de pérdida: binary_crosentropy para clasificación binaria.
- Optimizador: Adam con una tasa de aprendizaje de 0.0001.

- Número de épocas: 20
- Tamaño del lote: 32

Evaluación

El modelo se evaluó utilizando el conjunto de prueba para calcular las métricas de rendimiento, ajustando el umbral de decisión a 0.3 para mejorar la detección de noticias falsas. Las métricas calculadas fueron:

- **Accuracy:** Proporción de noticias correctamente clasificadas en el conjunto de prueba.
- **Precision:** Proporción de noticias clasificadas como falsas que realmente eran falsas.
- **Recall:** Capacidad del modelo para identificar correctamente todas las noticias falsas.
- **F1-Score:** Promedio armónico entre la precisión y el recall, proporcionando una medida balanceada del rendimiento del modelo.

V. Resultados y Discusiones

Resultados de los modelos aplicados

Tabla 2: Resultados de Desempeño de Modelos en Detección de Noticias Falsas

Modelo	Accuracy	Precisión	Recall	F1-Score
BERT	0.8480	0.6944	0.8621	0.7692
BETO	0.8495	0.7087	0.8295	0.7644
ALBETO	0.8194	0.7931	0.5227	0.6301
DistilBERT	0.8060	0.7083	0.5795	0.6375
GRU	0.7893	0.6543	0.6023	0.6272
LSTM	0.7860	0.6277	0.6705	0.6484
BiLSTM	0.8239	0.7869	0.4555	0.6443
CNN	0.7324	0.5281	0.8522	0.6521

Nota: Los resultados se expresan en función de métricas clave para cada modelo.

Discusiones

Sobre la hipótesis General

La hipótesis general planteaba que los modelos de Deep Learning basados en arquitecturas Transformer presentarían un mejor desempeño que los modelos tradicionales en la clasificación de noticias falsas relacionadas con el COVID-19. Los resultados obtenidos permiten aceptar la hipótesis general, ya que los modelos Transformer (BERT, BETO, ALBETO y DistilBERT) alcanzaron mayores valores de accuracy y F1-score en comparación con los modelos CNN, LSTM, GRU y BiLSTM. Este comportamiento puede explicarse por la capacidad de los Transformers para capturar dependencias contextuales globales mediante

mecanismos de autoatención, lo cual resulta especialmente relevante en textos informativos complejos y semánticamente ambiguos.

Estos hallazgos son coherentes con el estado del arte (Devlin, Chang, Lee, & Toutanova, 2019) y (Ayoub, Yang, & Zhou, 2021) reportan que los modelos basados en Transformers superan a arquitecturas recurrentes y convolucionales en tareas de clasificación textual y detección de desinformación. Asimismo, (Cañete, y otros, 2020) (Cañete, Donoso, Bravo, Carvallo, & Araujo, 2022) destacan que el preentrenamiento en corpus específicos del idioma español mejora significativamente el rendimiento en este tipo de tareas, lo cual concuerda con los resultados obtenidos en el presente estudio.

Sobre las hipótesis específicas

Los modelos basados en Transformers exhiben una mayor precisión en la clasificación de noticias falsas en comparación con los modelos basados en CNN y RNN: Los resultados de la tabla muestran que BERT y BETO, ambos modelos basados en Transformers, alcanzaron precisiones de 0.6944 y 0.7087, respectivamente. En comparación, el modelo CNN obtuvo una precisión de 0.5281 y los modelos RNN (GRU y LSTM) tuvieron precisiones de 0.6543 y 0.6277. Esto respalda la hipótesis, indicando que los modelos basados en Transformers son más confiables al clasificar correctamente las noticias falsas, especialmente en un contexto donde los matices del idioma español juegan un papel importante. La capacidad de los modelos Transformers para analizar el contexto bidireccionalmente parece haber sido ventajosa en esta tarea.

La integración de técnicas de aumento de datos y validación cruzada en el entrenamiento mejora el F1-score en la detección de noticias falsas: En este estudio, se implementaron técnicas de preprocesamiento y tokenización específicas para optimizar los resultados, aunque no se aplicaron técnicas avanzadas de aumento de datos ni validación cruzada compleja. Sin embargo, los modelos con un F1-score alto, como BERT y BETO, sugieren que la inclusión de técnicas adicionales de aumento y validación podría mejorar aún más la capacidad de generalización. El modelo BERT, por ejemplo, alcanzó un F1-score de 0.7692, mientras que CNN, sin tales técnicas avanzadas, obtuvo un F1-score de 0.6521.

Esto respalda la idea de que técnicas adicionales podrían tener un impacto positivo en los resultados finales.

El entrenamiento de modelos con un conjunto de datos preprocesado y diverso en términos de fuentes mejora el recall en la detección de noticias falsas: El recall es una métrica crítica para asegurar que la mayor cantidad posible de noticias falsas sean detectadas. Los resultados muestran que los modelos BERT y BETO, con un recall de 0.8621 y 0.8295, superaron a los modelos recurrentes como GRU y LSTM, cuyos valores de recall fueron 0.6023 y 0.6705, respectivamente. Esto sugiere que los modelos basados en Transformers tienen una mejor capacidad para identificar patrones asociados con noticias falsas, especialmente en un conjunto de datos diversificado. Sin embargo, el modelo CNN obtuvo un recall de 0.8522, indicando que puede ser eficaz en la detección, aunque su precisión en esta tarea sigue siendo limitada.

Comparación de rendimiento entre Transformers y otros modelos de Deep Learning en detección de noticias falsas: En general, los modelos Transformers demostraron un rendimiento superior en comparación con los otros tipos de modelos. Esto se observa claramente en el caso de BERT y BETO, que alcanzaron altos valores en precisión y F1-score, mientras que los modelos RNN y CNN presentaron una menor estabilidad en sus métricas. La capacidad de los Transformers para analizar secuencias de texto en ambas direcciones (bidireccionalmente) permite un análisis contextual más profundo, lo que los convierte en herramientas efectivas en la clasificación de noticias falsas en español.

Relevancia y Limitaciones

Los hallazgos obtenidos refuerzan la relevancia de utilizar modelos avanzados de NLP para el análisis de contenido en español. No obstante, este estudio tiene algunas limitaciones, como la ausencia de técnicas de aumento de datos avanzadas y la falta de una validación cruzada detallada. En investigaciones futuras, se sugiere explorar estas áreas para optimizar aún más el rendimiento de los modelos.

En conclusión, los modelos basados en Transformers como BERT y BETO son recomendables para la tarea de detección de noticias falsas en español, ya que han demostrado un equilibrio efectivo entre precisión y recall. Esta investigación subraya la importancia de adaptar y evaluar modelos de Deep Learning para aplicaciones en idioma español y ofrece una base para futuros estudios que busquen mejorar la precisión en el ámbito de la detección de desinformación en línea.

VI. Conclusiones

Conclusiones Generales

La La presente investigación evaluó la eficacia de diferentes modelos de Deep Learning en la detección de noticias falsas en español, específicamente en el contexto de la pandemia de COVID-19. La implementación de modelos basados en Transformers, como BERT y BETO, mostró un desempeño sobresaliente debido a su capacidad para capturar contextos lingüísticos complejos, mientras que las redes neuronales recurrentes (GRU, LSTM, BiLSTM) y convolucionales (CNN) obtuvieron resultados moderados en comparación. Se concluye que los modelos basados en Transformers son especialmente adecuados para esta tarea en idioma español, aunque se identifican oportunidades para mejorar el rendimiento a través de combinaciones de modelos y ajustes avanzados en el preprocesamiento de los datos.

Conclusiones específicas

Se investigaron y seleccionaron diversos algoritmos de Deep Learning para la clasificación de noticias falsas en español. Se optó por modelos basados en Transformers (BERT, BETO, ALBETO, DistilBERT), redes neuronales recurrentes (GRU, LSTM, BiLSTM) y redes convolucionales (CNN). Cada uno de estos modelos fue elegido por su capacidad para capturar características específicas de los datos, como dependencias temporales, contextos bidireccionales y patrones locales en las secuencias de texto. La diversidad de los modelos permitió evaluar sus fortalezas y debilidades en la tarea de detección de noticias falsas, destacando los modelos basados en Transformers como los más adecuados para esta tarea debido a su capacidad para comprender contextos complejos en el idioma español.

Se logró recopilar y preprocesar un conjunto de datos compuesto por noticias en español relacionadas con el COVID-19. El preprocesamiento incluyó técnicas de tokenización, limpieza y codificación de los textos, así como la transformación de las etiquetas para su uso en modelos de clasificación binaria. Los datos fueron divididos en conjuntos de entrenamiento, validación y prueba para asegurar una evaluación rigurosa de los modelos. Este proceso permitió disponer

de un conjunto de datos adecuado y balanceado para el entrenamiento de los diferentes modelos de Deep Learning seleccionados.

Se entrenaron y evaluaron los ocho modelos seleccionados utilizando el conjunto de datos preprocesado. Los resultados mostraron que los modelos basados en Transformers, como BERT y BETO, obtuvieron los mejores desempeños en términos de precisión y F1-Score, superando a las arquitecturas recurrentes y convolucionales. Cada modelo fue evaluado con métricas de rendimiento como Accuracy, Precision, Recall y F1-Score, lo que permitió identificar sus fortalezas y debilidades en la detección de noticias falsas. En general, todos los modelos fueron capaces de clasificar adecuadamente las noticias, aunque algunos mostraron desafíos específicos, como la tendencia del modelo CNN a clasificar incorrectamente noticias verdaderas como falsas.

Al comparar el rendimiento de los diferentes modelos, se observó que BERT obtuvo los mejores resultados globales, seguido de BETO y DistilBERT. Estos modelos mostraron un equilibrio adecuado entre precisión y recall, lo que los hace adecuados para la tarea de clasificación de noticias falsas en español. Los modelos recurrentes, como GRU y LSTM, aunque efectivos, presentaron un rendimiento ligeramente inferior, particularmente en términos de precisión. El modelo CNN mostró un rendimiento aceptable en términos de recall, pero su precisión fue baja, lo que sugiere que es más adecuado para detectar patrones específicos en los datos. En resumen, se selecciona BERT como el modelo más efectivo para esta tarea, debido a su capacidad para capturar contextos complejos y su rendimiento superior en todas las métricas evaluadas.

VII. Recomendaciones

Uso de Modelos Multimodales: Considerar la incorporación de modelos multimodales que combinen texto con otras fuentes de información, como imágenes o datos estructurados, para mejorar la precisión en la detección de noticias falsas. Por ejemplo, en el contexto de noticias falsas, la inclusión de análisis de imágenes junto con el texto puede proporcionar una capa adicional de verificación de información que fortalezca las predicciones del modelo.

Expansión del Conjunto de Datos: Ampliar el conjunto de datos incluyendo noticias de diferentes regiones y países de habla hispana, así como de diversas fuentes y estilos periodísticos. Esto permitirá que los modelos capten una mayor variedad de estilos de redacción y contextos lingüísticos, mejorando su capacidad para detectar noticias falsas en un espectro más amplio de casos y aumentando su robustez en diferentes contextos.

Integración de Técnicas de Transfer Learning: Aplicar técnicas de transfer learning utilizando modelos preentrenados en tareas específicas relacionadas, como la detección de noticias falsas en otros idiomas o la identificación de lenguaje ofensivo, podría mejorar el rendimiento de los modelos en la tarea de clasificación de noticias falsas en español.

VIII. Bibliografía

Ayoub, J., Yang, X., & Zhou, F. (2021). Combat COVID-19 infodemic using explainable natural language. doi:10.1016/j.ipm.2021.102569

Cañete, J., Chaperon, G., Fuentes, R., Ho, J. H., Perez, J., & Kang, H. (2020). Spanish Pre-trained BERT Model and Evaluation Data. 9. Obtenido de https://arxiv.org/pdf/2308.02976

Cañete, j., Donoso, S., Bravo, F., Carvallo, A., & Araujo, V. (2022). ALBETO and DistilBETO: Lightweight Spanish Language Models. Conference on Language Resources and Evaluation.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical Evaluation of.*

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for.* Obtenido de https://aclanthology.org/N19-1423.pdf

Douglas, K. M. (2021). Covid-19 conspiracy theories. *Group Processes & Intergroup Relations*. doi:10.1177/1368430220982068

Hochreiter, S., & Schmidhuber, j. (s.f.). *Long Short-Term Memory.* doi:https://doi.org/10.1162/neco.1997.9.8.1735

I. , Y. A., D., P. R., M. , S., & S. , P. (2022). Spatio-temporal approach for classification of covid-19 pandemic fake news. *Social Network Analysis and Mining, 12*(1), 68. doi:10.1007/s13278-022-00887-8

Kim, Y. (2014). *Convolutional Neural Networks for Sentence Classification.* New York. Obtenido de https://arxiv.org/pdf/1408.5882

Louise , E. S., Julius , S., Richard, A., Megan , C., Hannah, D., Nick , S., . . . Susan, M. S. (2022). Side-effect expectations from covid-19 vaccination: Findings from a nationally representative cross-sectional survey (covaccs–wave 2). *Journal of Psychosomatic Research, 152*. doi:10.1016/j.jpsychores.2021.110679

Mustapha , M., Abubakar , S., Abubakar , I. J., Ismaeel , Y., & Abubakar, M. I. (2021). Assessment of covid-19 information overload among the general public. *Journal of racial and ethnic health disparities*, 9. doi:10.1007/s40615-020-00942-0

Organizacion Mundial de la Salud. (s.f.). *Organizacion Munidial de la Salud*. Recuperado el 01 de 10 de 2024, de https://www.who.int/es/health-topics/coronavirus#tab=tab_1

Sergio Damian, Hiram, C., & Gelbukh, A. (2020). Feature Importance Analysis for Fake News Detection in Spanish. *Instituto Politecnico Nacional, Centro de Investigacion en ComputacIon*.

Sherstinsky, A. (2020). *Special Issue on Machine Learning and Dynamical Systems Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network* (Vol. 404). Elsevier journal.

Staudemeyer, R. C., & Rothstein Morris, E. (2019). *Understanding LSTM – a tutorial into Long Short-Term Memory.*

Zhang, X., & Ghorbani, A. A. (2020). An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management, 57*. doi:10.1016/j.ipm.2019.03.004

IX. Anexos

Matriz de Consistencia

Tabla 3: Matriz de Consistencia

Interrogantes	Objetivos	Hipótesis
General		
¿Qué modelos son los más recomendados para la clasificación de noticias?	Proponer Modelos de Deep learning para la detección de noticias falsas en español sobre COVID-19	La implementación de modelos avanzados de Deep Learning, basados en técnicas de Transformers mejora la precisión, el recall y el F1-score en la detección de noticias falsas en español sobre COVID-19, en comparación con modelos de Deep Learning CNN y RNN.
Específicas		
¿Qué modelos y enfoques existen para detectar noticias falsas sobre COVID-19, y cómo pueden aplicarse en el contexto específico de Perú y el idioma español?	Investigar y seleccionar algoritmos de Deep learning adecuados para la clasificación de texto	Los modelos de Deep Learning basados en Transformers, exhibe una mayor precisión en la clasificación de noticias falsas en comparación con los modelos basados en CNN y RNN.
¿Cómo afecta el preprocesamiento y la diversidad del conjunto de datos en términos de fuentes y estilos de redacción al recall de los	Recopilar y preprocesar un conjunto de datos de noticias en	La integración de técnicas de aumento de datos y validación cruzada en el entrenamiento de modelos de Deep

modelos de Deep Learning en la detección de noticias falsas sobre COVID-19 en español?	español sobre COVID-19	Learning para la detección de noticias falsas incrementa el F1-score de los modelos.
¿De qué manera la integración de técnicas de aumento de datos y validación cruzada durante el entrenamiento de modelos de Deep Learning afecta el F1-score en la detección de noticias falsas sobre COVID-19?	Entrenar y evaluar los modelos seleccionados utilizando el conjunto de datos preprocesado	El entrenamiento de modelos de Deep Learning con un conjunto de datos preprocesado y diverso en términos de fuentes y estilos de redacción mejora el recall en la detección de noticias falsas sobre COVID-19 en español.
¿Cuál es el rendimiento comparativo de diferentes modelos de Deep Learning entrenados para la detección de noticias falsas, y en qué medida los modelos basados en Transformers superan a otros tipos de modelos en esta tarea?	Comparar el rendimiento de los modelos y seleccionar el más efectivo	La comparación del rendimiento entre modelos de Deep Learning entrenados revela que los modelos basados en Transformers superan a otros tipos en la tarea de detección de noticias falsas

Nota: Esta tabla muestra la relación entre los interrogantes de investigación, objetivos e hipótesis planteados en el estudio sobre la detección de noticias falsas en español relacionadas con el COVID-19. Se especifican los modelos de Deep Learning seleccionados, así como la hipótesis general y específicas en torno a su efectividad en esta tarea.Dataset

Ilustración 6: Dataset de noticias falsas en español sobre COVID-19

	text	label
0	en vivo se agotan pasajes para distintos punt...	0
1	!buena noticia! peruanos ya pueden ir a la playa	0
2	coronacuento: as¡ funciona la l¢gica de quiene...	0
3	hay que ser tontos para no aprovechar, dice co...	0
4	iquitos ser¡a la primera ciudad en el mundo de...	0
5	alertartv alcalde de trujillo declara tras ex...	0
6	atenci¢n no salimos del omicron y ya apareci¢ ...	0
7	barranco: vecinos denuncian incremento de deli...	0
8	comunicado sobre la retenci¢n de pacientes o c...	0
9	covid19: los contagios se elevan en el per£. d...	0
10	essalud: m s de 56 000 adultos mayores recibir...	0
11	moquegua a partir del lunes estar prohibido ...	0
12	moquegua comerciantes de la feria chacra a la...	0
13	ox¡genomedicinal urge provisi¢n de ox¡geno en ...	0
14	pronunciamiento despu□s del personal de salud,...	0
15	sjl: pacientes covid19 duermen en los exterio...	0

Nota: este conjunto de datos contiene titulares de noticias falsas en español relacionadas con la pandemia de COVID-19, etiquetados como 0 para noticias falsas y 1 para verdaderas.

Modelos Implementados

Bert

Justificación del Modelo:

El modelo BERT (Bidirectional Encoder Representations from Transformers) fue seleccionado debido a su capacidad para procesar secuencias de texto capturando relaciones contextuales complejas de manera bidireccional. Este enfoque es particularmente útil en tareas de clasificación de noticias falsas, donde el contexto de las palabras y las relaciones entre ellas juegan un papel fundamental. BERT ha demostrado ser altamente efectivo en una variedad de tareas de procesamiento de lenguaje natural, lo que lo convierte en una elección sólida para

la clasificación de texto en español, como la detección de noticias falsas sobre COVID-19.

Configuración y Entrenamiento del Modelo BERT

Ilustración 7: Configuración del Modelo BERT

```
# Paso 4: Definir y configurar el modelo BERT
model = BertForSequenceClassification.from_pretrained("dccuchile/bert-base-spanish-wwm-uncased", num_labels=2)

# Configurar los argumentos de entrenamiento
training_args = TrainingArguments(
    output_dir='./results_bert',
    num_train_epochs=3,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs_bert',
    logging_steps=10,
    evaluation_strategy="epoch"
)

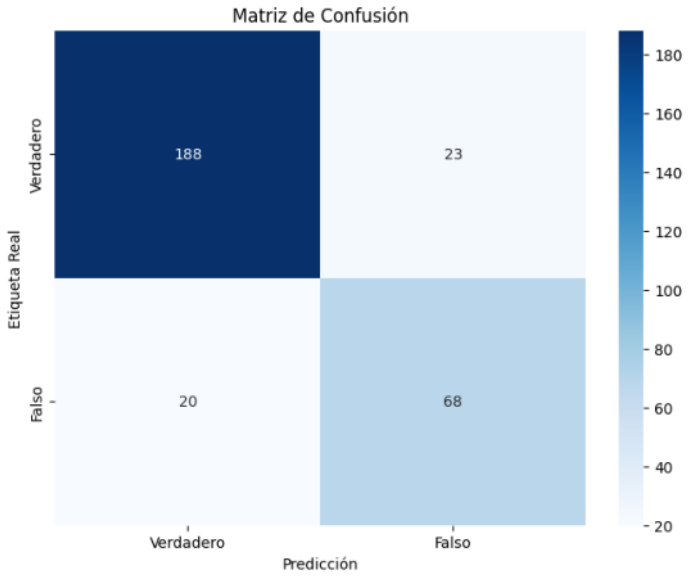
# Crear el objeto Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset
)

# Paso 5: Entrenamiento del modelo
trainer.train()
```

Nota: La imagen muestra el proceso de configuración de un modelo BERT preentrenado para clasificación de noticias en español.

Matriz de Confusión

Ilustración 8: Matriz de Confusión del Modelo BERT



Nota: La imagen muestra la matriz de confusión del modelo BERT, donde se observan los resultados de las predicciones de noticias falsas versus verdaderas. Los valores reflejan la cantidad de predicciones correctas e incorrectas en cada clase.

Conclusión

El modelo basado en BERT ha demostrado ser eficaz para la clasificación de noticias falsas en español. La preparación y el etiquetado cuidadoso de los datos, junto con el uso de técnicas avanzadas de procesamiento de lenguaje natural, han permitido obtener resultados sólidos y prometedores para la detección de noticias falsas. Sin embargo, es importante seguir explorando nuevas estrategias para mejorar la precisión y reducir la posibilidad de falsos positivos. Entre las posibles mejoras futuras se podrían considerar ajustes adicionales en los hiperparámetros, el uso de un mayor conjunto de datos o la integración de otros modelos complementarios.

BETO

Justificación del Modelo:

BETO, una versión entrenada exclusivamente en español del modelo BERT, fue seleccionado para capturar los matices y características específicas del idioma español, lo cual es crucial en la tarea de clasificación de noticias falsas. Este modelo aprovecha la capacidad bidireccional de BERT, adaptada para el idioma español, y ofrece una mejor representación semántica en comparación con modelos multilingües. BETO es una opción ideal para esta tarea, dado que el contexto cultural y lingüístico es clave para detectar noticias falsas en español.

Configuración y Entrenamiento del Modelo BETO

Ilustración 9: Configuración del Modelo BETO

```
# Paso 4: Definir y configurar el modelo BETO
model_beto = BertForSequenceClassification.from_pretrained("dccuchile/bert-base-spanish-wwm-uncased", num_labels=2)

# Configurar los argumentos de entrenamiento
training_args_beto = TrainingArguments(
    output_dir='./results_beto',
    num_train_epochs=3,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs_beto',
    logging_steps=10,
    eval_strategy="epoch"
)

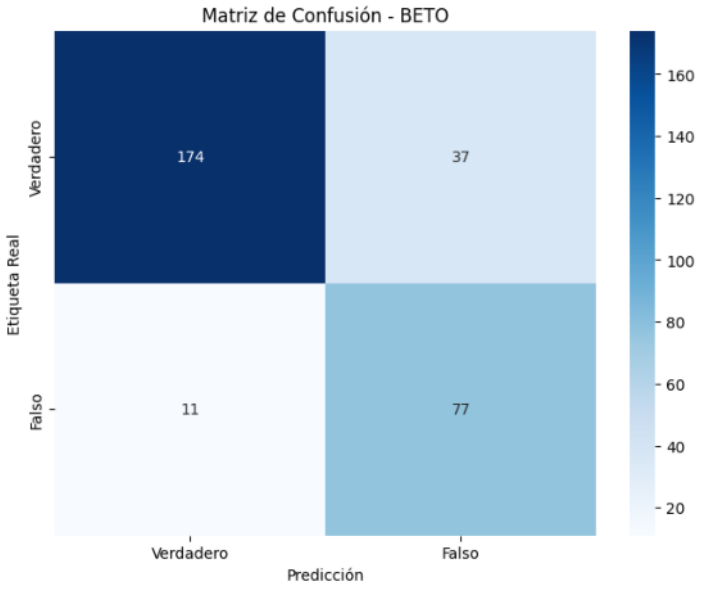
# Crear el objeto Trainer
trainer_beto = Trainer(
    model=model_beto,
    args=training_args_beto,
    train_dataset=train_dataset_beto,
    eval_dataset=val_dataset_beto
)

# Paso 5: Entrenamiento del modelo
trainer_beto.train()
```

Nota: La imagen muestra el proceso de configuración de un modelo BETO preentrenado para clasificación de noticias en español.

Matriz de Confusión

Ilustración 10: Matriz de Confusión del Modelo BETO



Nota: La imagen muestra la matriz de confusión del modelo BETO, donde se observan los resultados de las predicciones de noticias falsas versus verdaderas. Los valores reflejan la cantidad de predicciones correctas e incorrectas en cada clase.

Conclusión

El modelo BETO ha demostrado ser eficaz para la clasificación de noticias falsas en español, gracias a su capacidad para captar matices del idioma y comprender el contexto de manera bidireccional. Aunque el modelo ha mostrado un rendimiento sólido, todavía existe la posibilidad de mejorar su precisión ajustando los hiperparámetros o aumentando el tamaño del conjunto de datos. Además, la incorporación de técnicas de aumento de datos o el ajuste fino con otros corpus podría llevar a una mayor precisión en futuras investigaciones.

ALBETO

Justificación del modelo

El modelo ALBETO, una variante ligera de BETO, fue elegido debido a su capacidad para reducir la complejidad computacional mientras mantiene una precisión significativa en tareas de clasificación de texto. Su enfoque en detectar discurso de odio en español lo hace una opción viable para la clasificación de noticias falsas. ALBETO combina la eficiencia de un modelo reducido con el poder

de procesamiento semántico de un modelo preentrenado en español, lo que lo hace relevante en este contexto.

Configuración y Entrenamiento del Modelo ALBETO

Ilustración 11: Configuracion del Modelo ALBETO

```
# Paso 4: Definir y configurar el modelo ALBETO
model_albeto = AutoModelForSequenceClassification.from_pretrained("GalMargo/DistiBERTHateSpeech", num_labels=2, ignore_mismatched_sizes=True)

# Configurar los argumentos de entrenamiento
training_args_albeto = TrainingArguments(
    output_dir='./results_albeto',
    num_train_epochs=3,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs_albeto',
    logging_steps=10,
    eval_strategy="epoch"
)

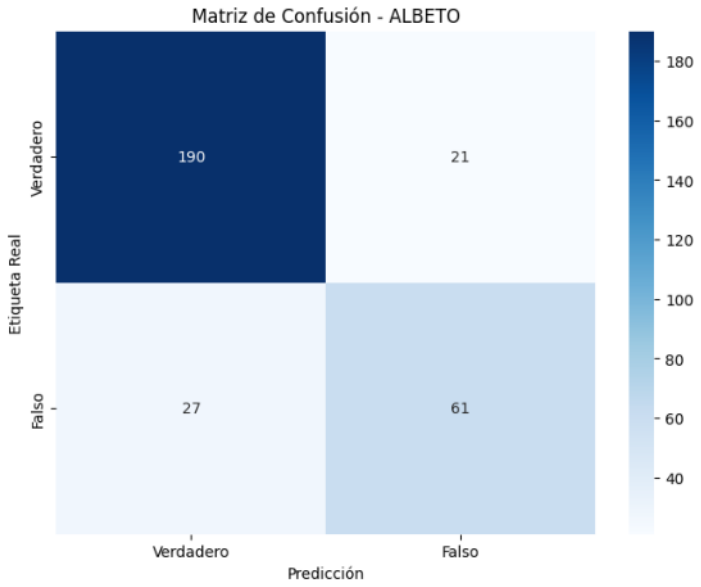
# Crear el objeto Trainer
trainer_albeto = Trainer(
    model=model_albeto,
    args=training_args_albeto,
    train_dataset=train_dataset_albeto,
    eval_dataset=val_dataset_albeto
)

# Paso 5: Entrenamiento del modelo
trainer_albeto.train()
```

Nota: La imagen muestra el proceso de configuración de un modelo ALBETO preentrenado para clasificación de noticias en español.

Matriz de Confusión

Ilustración 12: Matriz de Confusión del Modelo ALBETO



Nota: La imagen muestra la matriz de confusión del modelo ALBETO, donde se observan los resultados de las predicciones de noticias falsas versus verdaderas.

Los valores reflejan la cantidad de predicciones correctas e incorrectas en cada clase.

Conclusión

El modelo ALBETO ha mostrado ser eficaz para la detección de discurso de odio en español. Su capacidad para procesar y comprender el lenguaje en un contexto bidireccional le permite detectar patrones complejos asociados al discurso de odio. No obstante, los resultados obtenidos muestran que, aunque el modelo es preciso en sus predicciones, puede beneficiarse de ajustes adicionales en los hiperparámetros o de un mayor número de datos de entrenamiento para mejorar su capacidad de detección.

DistilBERT

Justificación del Modelo

El modelo DistilBERT fue seleccionado por su capacidad para ofrecer un equilibrio entre eficiencia computacional y precisión en tareas de clasificación de texto. DistilBERT es una versión más ligera y rápida de BERT, que preserva la mayor parte del rendimiento del modelo original, pero con un tamaño y velocidad reducidos. Esto lo convierte en una excelente opción para entornos con recursos limitados, donde la eficiencia es prioritaria, sin comprometer excesivamente el rendimiento en la tarea de detección de noticias falsas.

Configuración y Entrenamiento del Modelo DistilBERT

Ilustración 13: Configuración del Modelo DistilBERT

```
# Paso 4: Definir y configurar el modelo DistilBERT
model_distilbert = DistilBertForSequenceClassification.from_pretrained("distilbert-base-uncased", num_labels=2)

# Configurar los argumentos de entrenamiento
training_args_distilbert = TrainingArguments(
    output_dir='./results_distilbert',
    num_train_epochs=3,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs_distilbert',
    logging_steps=10,
    eval_strategy="epoch"
)

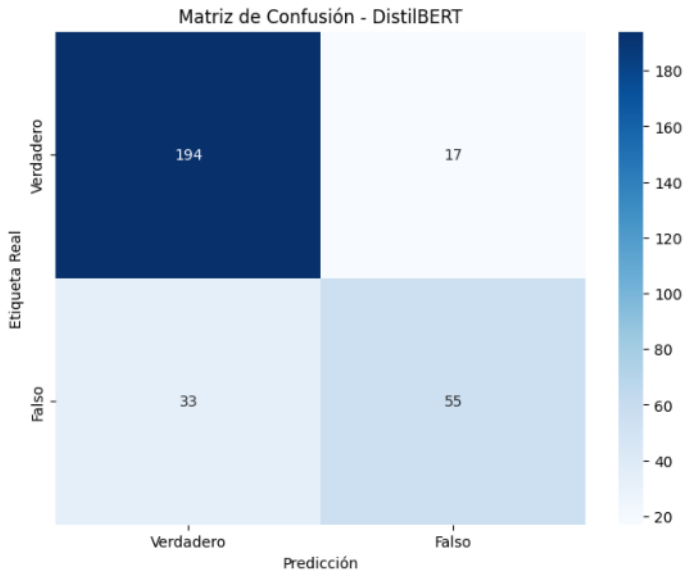
# Crear el objeto Trainer
trainer_distilbert = Trainer(
    model=model_distilbert,
    args=training_args_distilbert,
    train_dataset=train_dataset_distilbert,
    eval_dataset=val_dataset_distilbert
)

# Paso 5: Entrenamiento del modelo
trainer_distilbert.train()
```

Nota: La imagen muestra el proceso de configuración de un modelo DistilBERT preentrenado para clasificación de noticias en español.

Matriz de Confusión

Ilustración 14: Matriz de Confusión del Modelo DistilBERT



Nota: La imagen muestra la matriz de confusión del modelo DistilBERT, donde se observan los resultados de las predicciones de noticias falsas versus verdaderas. Los valores reflejan la cantidad de predicciones correctas e incorrectas en cada clase.

Conclusión

El modelo DistilBERT ha demostrado ser una opción eficiente y ligera para la clasificación de texto en español, manteniendo un equilibrio entre rendimiento y eficiencia computacional. Aunque es más rápido y ligero que BERT, el valor relativamente bajo de recall indica que podría mejorar su capacidad para detectar todas las instancias de discurso de odio. En aplicaciones futuras, se podría mejorar el rendimiento ajustando los hiperparámetros o utilizando técnicas adicionales de aumento de datos para mejorar la capacidad del modelo en entornos con datos más complejos.

GRU

Justificación del Modelo

El modelo GRU (Gated Recurrent Unit) fue seleccionado por su capacidad para manejar secuencias de texto de manera eficiente, capturando dependencias temporales sin la complejidad añadida de las LSTM. Las GRU son especialmente útiles para tareas como la clasificación de noticias falsas, donde las relaciones temporales entre las palabras pueden jugar un papel clave en la comprensión del contexto. Su simplicidad y eficiencia computacional lo hacen adecuado para la tarea, aunque con un rendimiento moderado en términos de recall.

Configuración y Entrenamiento del Modelo GRU

Ilustración 15: Configuración del Modelo GRU

```
# Paso 4: Definir y configurar el modelo GRU
model_gru = Sequential()
model_gru.add(Embedding(input_dim=vocab_size_gru, output_dim=128, input_length=maxlen_gru))
model_gru.add(GRU(units=128, return_sequences=True))
model_gru.add(Dropout(0.5))
model_gru.add(GRU(units=64))
model_gru.add(Dropout(0.5))
model_gru.add(Dense(1, activation='sigmoid'))

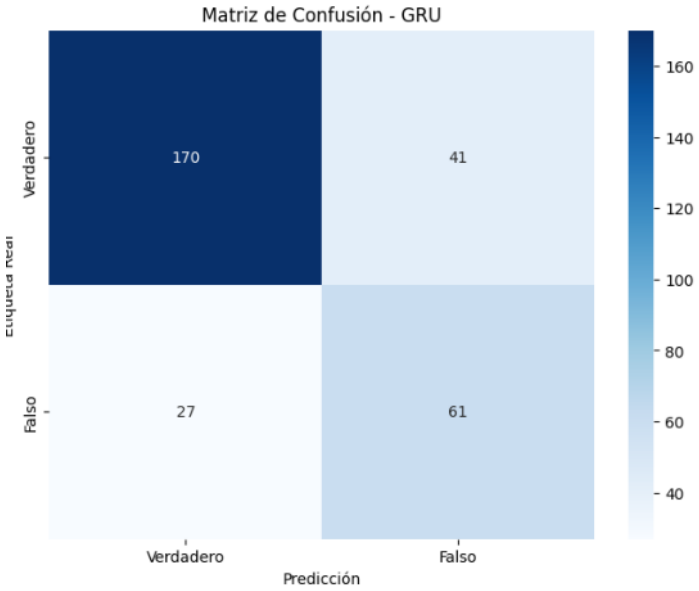
# Compilación del modelo
model_gru.compile(optimizer=Adam(learning_rate=0.0001), loss='binary_crossentropy', metrics=['accuracy'])

# Paso 5: Entrenamiento del modelo
history_gru = model_gru.fit(X_train_gru, y_train_gru, epochs=20, batch_size=32, validation_data=(X_val_gru, y_val_gru))
```

Nota: La imagen muestra el proceso de configuración de un modelo GRU preentrenado para clasificación de noticias en español.

Matriz de Confusion

Ilustración 16: Matriz de Confusión del Modelo GRU



Nota: La imagen muestra la matriz de confusión del modelo GRU, donde se observan los resultados de las predicciones de noticias falsas versus verdaderas. Los valores reflejan la cantidad de predicciones correctas e incorrectas en cada clase.

Conclusión

El modelo GRU ha demostrado ser una opción viable para la clasificación de secuencias en español. Su capacidad para aprender dependencias temporales dentro de los textos es particularmente útil para detectar patrones en noticias falsas. Sin embargo, el rendimiento moderado en cuanto al recall sugiere que el modelo podría mejorar con ajustes adicionales en los hiperparámetros o mediante la utilización de más datos de entrenamiento. La incorporación de técnicas de regularización adicionales o arquitecturas más complejas podría llevar a una mejora en los resultados globales del modelo.

LSTM

Justificación del Modelo

El modelo LSTM (Long Short-Term Memory) fue seleccionado por su capacidad para manejar dependencias a largo plazo en secuencias de texto, lo que es crucial en la clasificación de noticias falsas. Las LSTM superan los problemas de desvanecimiento de gradiente asociados a las RNN tradicionales, lo que las hace

una opción sólida para el procesamiento de secuencias largas. Aunque los resultados del modelo son prometedores, existen oportunidades para mejorar el rendimiento con ajustes adicionales en los hiperparámetros y mediante el uso de arquitecturas más complejas.

Configuración y Entrenamiento del Modelo LSTM

Ilustración 17: Configuración del Modelo LSTM

```
# Paso 4: Definir y configurar el modelo LSTM
model_lstm = Sequential()
model_lstm.add(Embedding(input_dim=vocab_size_lstm, output_dim=128, input_length=maxlen_lstm))
model_lstm.add(LSTM(units=128, return_sequences=True))
model_lstm.add(Dropout(0.5))
model_lstm.add(LSTM(units=64))
model_lstm.add(Dropout(0.5))
model_lstm.add(Dense(1, activation='sigmoid'))

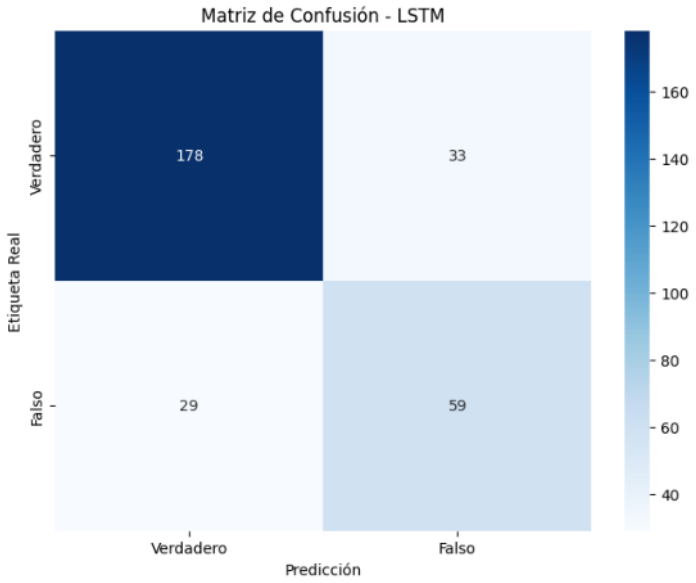
# Compilación del modelo
model_lstm.compile(optimizer=Adam(learning_rate=0.0001), loss='binary_crossentropy', metrics=['accuracy'])

# Paso 5: Entrenamiento del modelo
history_lstm = model_lstm.fit(X_train_lstm, y_train_lstm, epochs=20, batch_size=32, validation_data=(X_val_lstm, y_val_lstm))
```

Nota: La imagen muestra el proceso de configuración de un modelo LSTM preentrenado para clasificación de noticias en español.

Matriz de Confusión

Ilustración 18: Matriz de Confusión del Modelo LSTM



Nota: La imagen muestra la matriz de confusión del modelo LSTM, donde se observan los resultados de las predicciones de noticias falsas versus verdaderas. Los valores reflejan la cantidad de predicciones correctas e incorrectas en cada clase.

Conclusión

El modelo LSTM ha demostrado ser efectivo para la clasificación de secuencias en español, gracias a su capacidad para capturar dependencias temporales dentro de las secuencias de texto. A pesar de que los resultados son prometedores, existe la posibilidad de mejorar el rendimiento del modelo ajustando los hiperparámetros o utilizando técnicas de aumento de datos. Además, la incorporación de modelos híbridos o más complejos, como el uso combinado de LSTM y CNN, podría proporcionar mejoras adicionales en la capacidad del modelo para detectar noticias falsas.

BiLSTM

Justificación del Modelo

El modelo BiLSTM (Bidirectional Long Short-Term Memory) fue elegido por su capacidad para procesar secuencias en ambas direcciones, capturando tanto el contexto previo como el posterior de una palabra dentro de una secuencia. Esto lo hace particularmente útil para la clasificación de noticias falsas, donde la comprensión bidireccional del texto es crucial. A pesar de su eficacia, el rendimiento del modelo podría mejorar con la integración de técnicas adicionales de aumento de datos y ajustes en la arquitectura.

Configuración y Entrenamiento del Modelo BiLSTM

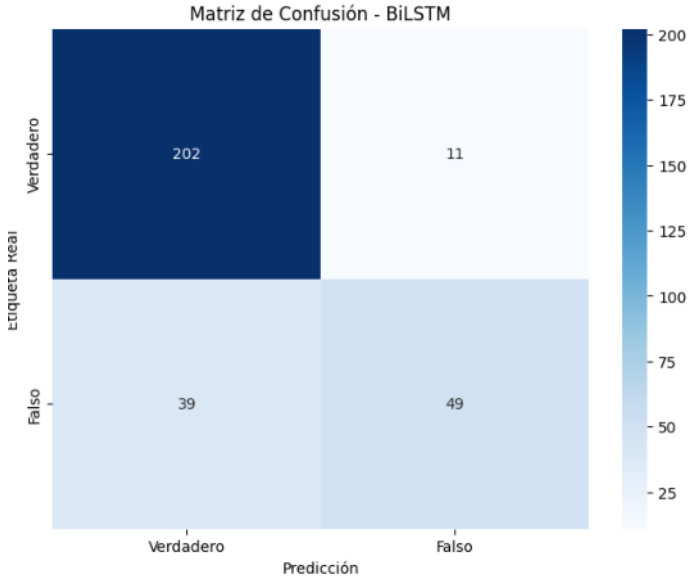
Ilustración 19: Configuración del Modelo BiLSTM

```
# Definir el modelo BiLSTM
embedding_dim = 100  # dimensión del embedding
model = Sequential()
model.add(Embedding(input_dim=num_words, output_dim=embedding_dim, input_length=max_length))
model.add(Bidirectional(LSTM(units=64, return_sequences=True)))
model.add(Dropout(0.5))
model.add(Bidirectional(LSTM(units=64)))
model.add(Dense(1, activation='sigmoid'))
model.add(Dropout(0.5))
# Compilar el modelo
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
history = model.fit(X_train, y_train, epochs=20, batch_size=32, validation_data=(X_val, y_val))
# Evaluar el modelo en el conjunto de prueba
loss, accuracy = model.evaluate(X_test, y_test)
```

Nota: La imagen muestra el proceso de configuración de un modelo BiLSTM preentrenado para clasificación de noticias en español.

Matriz de Confusión

Ilustración 20: Matriz de Confusión del Modelo BiLSTM



Nota: La imagen muestra la matriz de confusión del modelo BiLSTM, donde se observan los resultados de las predicciones de noticias falsas versus verdaderas. Los valores reflejan la cantidad de predicciones correctas e incorrectas en cada clase.

Conclusiones

El modelo Bidirectional LSTM (BiLSTM) ha demostrado ser efectivo para la clasificación de secuencias en español, gracias a su capacidad para capturar dependencias bidireccionales dentro de las secuencias de texto. A pesar de que los resultados son prometedores, existe la posibilidad de mejorar el rendimiento del modelo ajustando los hiperparámetros o utilizando técnicas de aumento de datos. Además, la incorporación de arquitecturas más complejas o el uso de técnicas de preentrenamiento podrían proporcionar mejoras adicionales en la capacidad del modelo para detectar noticias falsas.

CNN

Justificacion del Modelo

El modelo CNN (Convolutional Neural Network) fue seleccionado por su capacidad para identificar patrones locales en las secuencias de texto, lo que es clave para detectar características distintivas en noticias falsas. Aunque las CNN son más comúnmente utilizadas en el procesamiento de imágenes, su capacidad

para capturar n-gramas y patrones locales ha demostrado ser efectiva en la clasificación de texto. No obstante, su rendimiento en términos de precisión sugiere que puede ser útil explorar estrategias adicionales para mejorar su capacidad de generalización.

Configuración y Entrenamiento del Modelo CNN

Ilustración 21: Configuración del Modelo CNN

```
# Paso 4: Definir y configurar el modelo CNN
model_cnn = Sequential()
model_cnn.add(Embedding(input_dim=vocab_size_cnn, output_dim=128  ))
model_cnn.add(Conv1D(filters=128, kernel_size=5, activation='relu'))
model_cnn.add(GlobalMaxPooling1D())
model_cnn.add(Dropout(0.5))
model_cnn.add(Dense(64, activation='relu'))
model_cnn.add(Dropout(0.5))
model_cnn.add(Dense(1, activation='sigmoid'))

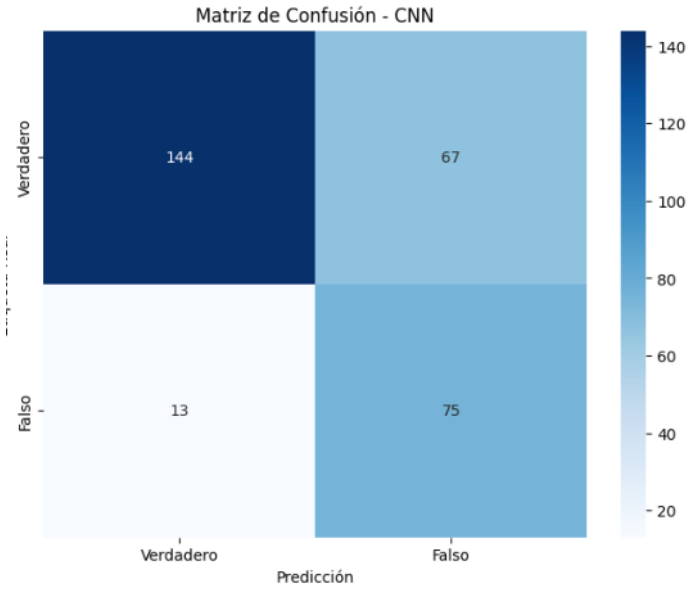
# Compilación del modelo
model_cnn.compile(optimizer=Adam(learning_rate=0.0001), loss='binary_crossentropy', metrics=['accuracy'])

# Paso 5: Entrenamiento del modelo
history_cnn = model_cnn.fit(X_train_cnn, y_train_cnn, epochs=20, batch_size=32, validation_data=(X_val_cnn, y_val_cnn))
```

Nota: La imagen muestra la matriz de confusión del modelo CNN, donde se observan los resultados de las predicciones de noticias falsas versus verdaderas. Los valores reflejan la cantidad de predicciones correctas e incorrectas en cada clase.

Matriz de Confusión

Ilustración 22: Matriz de Confusión del Modelo CNN



Conclusiones

El modelo CNN ha demostrado ser una herramienta útil para la detección de patrones en las secuencias de texto, capturando eficazmente las características locales que distinguen a las noticias falsas. Sin embargo, su rendimiento en términos de precisión indica que puede estar clasificando incorrectamente algunas noticias verdaderas como falsas. Para mejorar su efectividad, se podrían explorar estrategias adicionales como el ajuste fino de hiperparámetros, la incorporación de capas convolucionales adicionales o el uso de técnicas de aumento de datos. Además, la integración de un ensamble con otros modelos podría ayudar a equilibrar mejor las métricas de precisión y recall.

FORMATO DE AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL UNAM

Formulario de autorización para la publicación electrónica en la página web del Repositorio Institucional de la Universidad Nacional de Moquegua, de conformidad con la Ley N° 30035 Ley que regula el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto y Resolución de Comisión Organizadora N° 041-2022-UNAM que aprueba el Reglamento que regula el Repositorio Institucional Digital de la UNAM.

1. DATOS DEL AUTOR

Nombres y Apellidos : Pedro Luis Ramos Calla
DNI/Carnet de Extranjería/Pasaporte N° : 79723000
Correo electrónico : pedrorc2018@gmail.com
Teléfono/Celular : 958104634

2. GRADO, TITULO O ESPECIALIZACIÓN (Marque con una X)

Bachiller	Título Profesional	Magister	Doctor	Segunda Especialidad
	x			

3. TIPO DE PUBLICACIÓN (Marque con una X)

a) Articulo ( )
b) Tesis de pregrado (x)
c) Tesis de Maestría ( )
d) Tesis de Doctorado ( )
e) Monografía ( )
f) Reporte de caso científico o hallazgos ( )
g) Libro ( )
h) Parte de un libro ( )
i) Revisión ( )
j) Conferencia ( )
k) Presentación ( )
l) Artículo preliminar ( )
m) Articulo antes de ser publicado ( )
n) Artículo de prensa ( )
o) Documentos internos ( )
p) Imagen ( )
q) Documentos técnicos ( )
r) Ponencias presentadas en eventos científicos (Jornada, Congreso, Simposio Conferencias, etc.) ( )
s) Informes de suficiencia profesional ( )
t) Patentes producidas por académicos/as ( )

4. TIPO DE TRABAJO DE INVESTIGACIÓN (Marque con una X)

a) Tesis (x)
b) Trabajo de Investigación ( )
c) Trabajo de suficiencia profesional ( )
d) Trabajo académico ( )

5. DATOS DE LA PUBLICACION

a) Título : Modelos de Deep learning para la detección de noticias falsas en español sobre COVID-19
b) Campo del conocimiento OCDE (Indicar el código de clasificación OCDE de la obra : https://catalogos.concytec.gob.pe/vocabulario/ocde_ford.html) : 1.02.01
c) Fecha de publicación : enero del 2026

6. DATOS DEL ASESOR:

a) Nombres y Apellidos : Anibal Fernando Flores Garcia
b) DNI/Pasaporte/Cédula : 04743476
c) ORCID : 0000-0003-4638-8242

FIRMA : .....

DOCUMENTO DE IDENTIDAD : 79723000
NOMBRES Y APELLIDOS : Pedro Luis Ramos Calla

LICENCIAS

Licencia Estándar

Bajo los siguientes términos, autorizo el depósito de mi trabajo de investigación en el Repositorio Institucional Digital de la UNAM.

Con la autorización de depósito da mi trabajo de investigación, otorgo a la Universidad Nacional de Moquegua, una Licencia No Exclusiva para reproducir. distribuir, comunicar al público, transformar (únicamente mediante su traducción a otros idiomas) y poner a disposición del público mi trabajo de investigación (incluido el resumen), en formato físico o digital, en cualquier medio, conocido o por conocerse, a través de los diversos servicios provistos por la Universidad, creados o por crearse, tales como el Repositorio Institucional Digital de la UNAM, Colección de Tesis, entre otros, en el Perú y en el extranjero, por el tiempo y veces que considere necesarias, y libre de remuneraciones.

En virtud de dicha Licencia, la UNIVERSIDAD NACIONAL DE MOQUEGUA, podrá reproducir mi trabajo de investigación en cualquier tipo de soporte y en más de un ejemplar, sin modificar su contenido, solo con los propósitos de seguridad, respaldo y preservación.

Declaro que el trabajo de investigación es una creación de mi autoría y exclusiva titularidad, o coautoría con titularidad compartida, y me encuentro facultado a conceder la presente licencia y, asimismo, garantizo que dicho trabajo de investigación no infringe derechos de autor de terceras personas.

La UNIVERSIDAD NACIONAL DE MOQUEGUA consignara el nombre del/los autor/es del trabajo de investigación, y no le hará ninguna modificación más de la permitida en la presente licencia.

Autorizo el depósito (marque con una X).

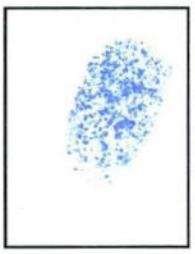
(x) **SI autorizo que se deposite inmediatamente.**

FIRMA : .....

DOCUMENTO DE IDENTIDAD : 79723000

NOMBRES Y APELLIDOS : Pedro Luis Ramos Calla

FECHA : 19 DE ENERO DEL 2026



ANEXO VIII

DECLARACIÓN JURADA DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN / TESIS / TRABAJO DE SUFICIENCIA PROFESIONAL

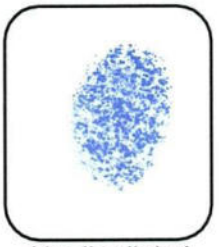
Yo, Pedro Luis Ramos Calla,
bachiller de la Escuela Profesional de:
Ingeniería de Sistemas e Informática,
de la Universidad Nacional de Moquegua, identificado(a) con DNI Nº. 79723000, autor(a) del Trabajo de Investigación / Tesis / Trabajo de suficiencia profesional denominado:
Modelos de Deep learning para la detección de noticias falsas en español sobre COVID-19

Declaro bajo juramento que:

1. El trabajo de investigación es auténtico, siendo resultado de mi trabajo personal, que no se ha copiado, que no se ha utilizado ideas, formulaciones, citas integrales e ilustraciones diversas, sacadas de cualquier tesis, obra, artículo, memoria, etc., (en versión digital o impresa), sin mencionar de forma clara y exacta su origen o autor, tanto en el cuerpo del texto, figuras, cuadros, tablas u otros que tengan derechos de autor. Por tanto, el trabajo de investigación no ha sido plagiado total ni parcialmente.
2. El trabajo de investigación no ha sido autoplagiada, es decir, no ha sido publicada ni presentada anteriormente para obtener algún grado académico previo o título profesional.
3. Los datos presentados en los resultados son reales, no han sido falseados, ni duplicados, ni copiados y por tanto los resultados que se presentan en el trabajo de investigación constituirán aporte a la realidad investigada.

Soy consciente de que, el hecho de no respetar los derechos de autor y hacer plagio, es objeto de sanciones universitarias y/o legales, por lo que asumo cualquier responsabilidad que pudiera derivarse de irregularidades en el trabajo de investigación, así como de los derechos sobre la obra presentada.

Moquegua, 19 DE ENERO DEL 2026



Huella digital

Nombres y Apellidos: Pedro Luis Ramos Calla
DNI: 79723000

TESIS V2

9%
Textos sospechosos

6% Similitudes
0% similitudes entre comillas
0% entre las fuentes mencionadas
4% Idiomas no reconocidos

Nombre del documento: TESIS V2.docx
ID del documento: fdc3ebceade8690f0a7ed263895ce9e8a23c34a0
Tamaño del documento original: 689,29 kB
Autores: []

Depositante: ANIBAL FERNANDO FLORES GARCIA
Fecha de depósito: 1/10/2024
Tipo de carga: interface
fecha de fin de análisis: 1/10/2024

Número de palabras: 11.233
Número de caracteres: 76.219

Ubicación de las similitudes en el documento:



Fuentes de similitudes

Fuentes principales detectadas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	**rodin.uca.es** https://rodin.uca.es/bitstream/10498/24453/5/Optimizacion_SW_NLP-Morales_Millan_Jesus-TFG.pdf 1 fuente similar	1%		Palabras idénticas: 1% (103 palabras)
2	**pythonguia.com** | Scikit Learn matriz de confusión https://pythonguia.com/scikit-learn-matriz-de-confusion/ 3 fuentes similares	< 1%		Palabras idénticas: < 1% (87 palabras)
3	**ciberseguridadmax.com** | ¿Qué es Epoch en Machine Learning? https://ciberseguridadmax.com/epoch/ 7 fuentes similares	< 1%		Palabras idénticas: < 1% (75 palabras)
4	**universo-ia.com** | Evaluación de modelos: Métricas y técnicas para medir el rendi... https://universo-ia.com/evaluacion-de-modelos-metricas-y-tecnicas-para-medir-el-rendimiento/#:~:... 1 fuente similar	< 1%		Palabras idénticas: < 1% (67 palabras)
5	**pythonguia.com** | Scikit Learn Accuracy_score https://pythonguia.com/scikit-learn-accuracy_score/	< 1%		Palabras idénticas: < 1% (69 palabras)

Fuentes con similitudes fortuitas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	**sensoricx.com** | Procesamiento de Lenguaje Natural: El poder de la comunicación https://sensoricx.com/conocimiento/procesamiento-de-lenguaje-natural-nlp/#:~:text=El Procesamie...	< 1%		Palabras idénticas: < 1% (26 palabras)
2	**reunir.unir.net** https://reunir.unir.net//bitstream/123456789/14608/1/Simón Gallego, Carlos.pdf	< 1%		Palabras idénticas: < 1% (26 palabras)
3	**ndmarketingdigital.com** | Guía detallada para entender y explicar la matriz de co... https://ndmarketingdigital.com/como-explicar-la-matriz-de-confusion/	< 1%		Palabras idénticas: < 1% (20 palabras)
4	**prompt.uno** | Todo sobre la Memoria a Corto Plazo Largo - Prompts para IA https://prompt.uno/inteligencia-artificial/todo-sobre-la-memoria-a-corto-plazo-largo/	< 1%		Palabras idénticas: < 1% (20 palabras)
5	**Documento de otro usuario** #96d278 El documento proviene de otro grupo	< 1%		Palabras idénticas: < 1% (22 palabras)

Fuentes mencionadas (sin similitudes detectadas)

Estas fuentes han sido citadas en el documento sin encontrar similitudes.

1 https://aclanthology.org/N19-1423.pdf
2 https://doi.org/10.1162/neco.1997.9.8.1735
3 https://arxiv.org/pdf/1408.5882
4 https://www.who.int/es/health-topics/coronavirus#tab=tab_1

Puntos de interés

UNIVERSIDAD NACIONAL DE MOQUEGUA

ESCUELA PROFESIONAL DE INGENIERÍA DE
SISTEMAS E INFORMÁTICA

[]

MODELOS DE DEEP LEARNING PARA LA DETECCION DE NOTICIAS FALSAS EN ESPAÑOL SOBRE COVID-19

PRESENTADO POR: PEDRO LUIS RAMOS CALLA

1 repositorio.unam.edu.pe
https://repositorio.unam.edu.pe/bitstream/UNAM/84/5/T095_70490643_T.pdf.txt

PARA OPTAR EL TÍTULO PROFESIONAL DE:
INGENIERO DE SISTEMAS E
INFORMATICA
MOQUEGUA – PERU

2024

Tabla de contenido

I.Datos Generales13
Titulo13
Datos Del Investigador13
Línea De Investigación13
II.Problema de Investigación13
Descripción de la Realidad Problemática13
Formulación del Problema de Investigación15
Interrogante General15
Interrogantes Específicas15
Justificación15
Objetivos16
Objetivo General16
Objetivo Específicas16
Hipótesis16
Hipótesis General16
Hipótesis Especifica17
III.Marco Teórico17
Antecedentes del problema17
Bases teóricas19
IV.Metodología23
Tipo de investigación23
Diseño Experimental24
Operacionalización de variables24
Población y Muestra24
Técnicas e instrumentos para la recolección de datos24
Métodos y técnicas para la presentación y análisis de Datos24
Procesamiento de los datos25
Recolección de información:
Etiquetado de Datos26
Division de Datos26
Modelos Utilizados27
BERT27
Descripción del Modelo27
Modelamiento28
Evaluacion29
Resultados29
BETO30
Descripción del Modelo30
Modelamiento30
Evaluación32
Resultados32
ALBETO32
Descripción del modelo32
Modelamiento33
Evaluación34
Resultados34
DistilBERT35
Descripción del Modelo35
Modelamiento36
Evaluación37
Resultados37
GRU38
Descripción del Modelo38
Modelamiento38
Evaluación39
Resultados40
LSTM41
Descripcion del Modelo41

Modelamiento41
Evaluacion42
Resultados43
BiLSTM43
Descripción del Modelo44
Modelamiento44
Evaluacion45
Resultados46
CNN46
Descripcion del Modelo47
Modelamiento47
Evaluacion48
Resultados49
Resultados49
V.
Conclusiones50
Conclusiones Generales50
Conclusiones especificas50
VI.
Recomendaciones52
VII.Bibliografía53
VIII.
Anexos54
Matriz de Consistencia54
Dataset55
Modelos Implementados56
Bert56
tificación del Modelo56
Configuración y Entrenamiento del Modelo BERT56
Matriz de Confusión57
Conclusión57
BETO57
Justificación del Modelo:57
Configuración y Entrenamiento del Modelo BETO58
Matriz de Confusión58
Conclusión58
ALBETO59
Justificación del modelo59
Configuración y Entrenamiento del Modelo ALBETO59
Matriz de Confusión60
Conclusión60
DistilBERT60
Justificación del Modelo60
Configuración y Entrenamiento del Modelo DistilBERT61
Matriz de Confusión61
Conclusión61
GRU62
Justificación del Modelo62
nfiguración y Entrenamiento del Modelo GRU62
atriz de Confusion62
Conclusión63
LSTM63
Justificación del Modelo63
Configuración y Entrenamiento del Modelo LSTM64
Matriz de Confusión64
Conclusión64
BiLSTM65
Justificación del Modelo65
Configuración y Entrenamiento del Modelo BiLSTM65
Matriz de Confusión65
Conclusiones66
CNN66
Justificacion del Modelo66
Configuración y Entrenamiento del Modelo
CNN67
Matriz de Confusión67
Conclusiones67

2 rodin.uca.es
https://rodin.uca.es/bitstream/10498/24453/5/Optimizacion_SW_NLP-Morales_Millan_Jesus-TFG.pdf

Tabla 1:

Operacionalización de variables24
Tabla 2:
Resultados del Modelo BERT30
Tabla 3: Resultados del Modelo BETO32
Tabla 4: Resultados del Modelo ALBETO35
Tabla 5: Resultados del Modelo DistilBERT37

Tabla 6: Resultaos del Modelo GRU40
Tabla 7: Resultados del Modelo LSTM43
Tabla 8: Resultados del Modelo BiLSTM46
Tabla 9: Resultados del Modelo CNN49
Tabla 10: Tabla de Resultados de los Modelos Aplicados50
Tabla 11: Matriz de Consistencia55

Ilustración 1: Dataset55
Ilustración 2: Configuración del Modelo Bert56
Ilustración 3: Matriz de Confusión del Modelo Bert57
Ilustración 4: Configuración del Modelo BETO58
Ilustración 5: Matriz de Confusión del Modelo BETO58
Ilustración 6: Configuracion del Modelo ALBETO59
Ilustración 7: Matriz de Confusión del Modelo ALBETO60
Ilustración 8: Configuración del Modelo DistilBERT61
Ilustración 9: Matriz de Confusión del Modelo DistilBERT61
Ilustración 10: Configuración del Modelo GRU62
Ilustración 11: Matriz de Confusión del Modelo GRU63
Ilustración 12: Configuración del Modelo LSTM64
Ilustración 13: Matriz de Confusión del Modelo LSTM64
Ilustración 14: Configuración del Modelo BiLSTM65
Ilustración 15: Matriz de Confusión del Modelo BiLSTM66
Ilustración 16: Configuración del Modelo CNN67
Ilustración 17: Matriz de Confusión del Modelo CNN67

Datos Generales
Titulo
Modelos de Deep Learning para la Detección de Noticias Falsas en español sobre COVID-19
Datos Del Investigador
dro Luis Ramos Calla
ea De Investigación
Inteligencia Artificial
Problema de Investigación
Descripción de la Realidad Problemática
Durante los primeros meses de la pandemia, sus efectos comenzaron a nivel de infectados y decesos, hospitales y centros de salud con alta demanda debido al incremento de pacientes de COVID-19. Los números reportados oficiales fueron publicados por el Ministerio de Salud y permitían conocer la evolución de los casos en Perú y el gobierno Perú estableció políticas de salud pública para contener la rápida propagación, cuarentena totales y luego parciales. Permitiendo que sectores esenciales puedan continuar sus labores considerando las recomendaciones del Ministerio de Salud.
La infodemiología, término acuñado por Eysenbach tiene como objetivo mejorar la salud pública analizando las fuentes digitales que los usuarios de Internet usan, como foros, blogs, redes sociales, motores de búsqueda. Eysenbach demostró su argumento al encontrar una correlación entre las búsquedas de Google y los casos de gripe. Demostrando que podían ser usadas estas fuentes de datos para vigilancia epidemiológica.
Las noticias falsas en las redes sociales son un problema crítico ya que pueden perjudicar a la población (Zhang & Ghorbani, 2020). Las noticias falsas pueden crearse con la intención de engañar, manipular, difamar o generar miedo en la audiencia. La Organización Mundial de la Salud ha afirmado que las noticias falsas o infodemia son más mortales que el propio virus Covid-19, han afectado a las personas tanto física como mentalmente (I. , D., M. , & S. , 2022).
La pandemia de COVID-19 (Organizacion Mundial de la Salud, s.f.). ha generado un gran volumen de noticias falsas, desinformación y teorías de conspiración en todo el mundo. Algunas de las noticias falsas más comunes relacionadas con la pandemia incluyen teorías de conspiración (Douglas, 2021) sobre el origen del virus, como afirmar que el virus fue creado por el hombre; afirmaciones falsas sobre remedios o curas para enfermedades (Mustapha , Abubakar , Abubakar , Ismaeel , & Abubakar, 2021), información errónea sobre medidas preventivas, como afirmar que el uso de mascarillas es perjudicial para la salud o que la vacunación no es segura (Louise , y otros, 2022); cifras de casos y muertes, como afirmar que las cifras son exageradas o manipuladas por razones políticas.
Por tanto, la presente tesis tiene como objetivo realizar un estudio Infodemiológico usando una metodología de minería de datos y técnicas de procesamiento de lenguaje natural. y fuentes de datos digitales como resultados de motores de búsqueda (Google), redes sociales y datos de movilidad o desplazamiento. A la fecha existen algunos estudios relacionados con énfasis en Perú, más el contexto continúa evolucionando con 4 la aparición de nuevas variantes, vacunas y políticas del gobierno referentes a la salud pública. Y bajo nuestro entendimiento, no existe un estudio actualizado sobre el tema principal del proyecto, y que integre varias fuentes digitales para su estudio y que pueda ser una fuente de información para el gobierno y realizar mejores políticas públicas en salud.
rmulación del Problema de Investigación
Interrogante General
¿Qué modelos son los más recomendados para la clasificación de noticias?
Interrogantes Específicas
¿Qué modelos y enfoques existen para detectar noticias falsas sobre COVID-19, y cómo pueden aplicarse en el contexto específico de Perú y el idioma español?
¿Cómo afecta el preprocesamiento y la diversidad del conjunto de datos en términos de fuentes y estilos de redacción al recall de los modelos de Deep Learning en la detección de noticias falsas sobre COVID-19 en español?
¿De qué manera la integración de técnicas de aumento de datos y validación cruzada durante el entrenamiento de modelos de Deep Learning afecta el F1-score en la detección de noticias falsas sobre COVID-19?
¿Cuál es el rendimiento comparativo de diferentes modelos de Deep Learning entrenados para la detección de noticias falsas, y en qué medida los modelos basados en Transformers superan a otros tipos de modelos en esta tarea?
Justificación
En lo social, El presente proyecto puede proporcionar información valiosa sobre las preocupaciones y percepciones de la población en relación con cuestiones del COVID-19.
En lo económico, el tesista cuenta con el apoyo de la Universidad Nacional de Moquegua, ya que el tema propuesto es parte de un proyecto de investigación financiado con fondos de canon y regalías mineras.
Desde el punto de vista tecnológico se cuenta con las herramientas de software y conocimientos necesarios para llevar a cabo la investigación.
Por lo mencionado en los párrafos precedentes se justifica la ejecución del proyecto de investigación descrito.
Objetivos
Objetivo General
Proponer Modelos de Deep learning para la detección de noticias falsas en español sobre Covid-19
Objetivo Específicas
Investigar y seleccionar algoritmos de Deep learning adecuados para la clasificación de texto
Recopilar y preprocesar un conjunto de datos de noticias en español sobre COVID-19
Entrenar y evaluar los modelos seleccionados utilizando el conjunto de datos preprocesado
Comparar el rendimiento de los modelos y seleccionar el más efectivo
Hipótesis
Hipótesis General
La implementación de modelos avanzados de Deep Learning, basados en técnicas de Transformers mejora la precisión, el recall y el F1-score en la detección de noticias falsas en español sobre COVID-19, en comparación con modelos de Deep Learning CNN y RNN.

Hipótesis Especifica
Los modelos de Deep Learning basados en Transformers, exhibe una mayor precisión en la clasificación de noticias falsas en comparación con los modelos basados en CNN y RNN.
La integración de técnicas de aumento de datos y validación cruzada en

3 **ciberseguridadmax.com** | ¿Qué es Epoch en Machine Learning?
https://ciberseguridadmax.com/epoch/

el entrenamiento de modelos de Deep Learning para la detección de noticias falsas incrementa el F1-score de los modelos.
El entrenamiento de modelos de Deep Learning
con un conjunto de datos preprocesado y diverso en términos de fuentes y estilos de redacción mejora el recall en la detección de noticias falsas sobre COVID-19 en español.
La comparación del rendimiento entre modelos de Deep Learning entrenados revela que los modelos basados en Transformers superan a otros tipos en la tarea de detección de noticias falsas
Marco Teórico
Antecedentes del problema
Se han llevado a cabo estudios de vigilancia epidemiológica utilizando un enfoque de infodemiología para analizar diversas enfermedades, siguiendo la propuesta de Eysenbach. Esto ha implicado un aumento en las fuentes digitales utilizadas para la recolección de datos y en el alcance del análisis realizado.
(Steffen , Michael , & Artur , 2021), Se observó un aumento en la cantidad de investigaciones realizadas sobre infodemiología entre marzo de 2020 y enero de 2021. Durante este período, se utilizó una variedad de fuentes, incluido Google Trends, para analizar el interés en temas relacionados con el coronavirus, el confinamiento y el distanciamiento social. Estas palabras clave se utilizaron para rastrear las tendencias de búsqueda y evaluar su impacto en la información disponible en línea. Los estudios, de naturaleza global, confirmaron la hipótesis de que el aumento en las búsquedas en Google se correlacionaba con un incremento en las publicaciones relacionadas con la vigilancia de la información, la infodemiología y el uso de Google Trends como herramienta de análisis.
(Martin , Stuckler, & Arora, 2019), reafirma uso de motores de búsqueda en los últimos años para acceder a información sobre una variedad de temas relacionados con servicios para consumidores y productos de consumo ha sido robusta. Simultáneamente, ha surgido un interés en aumento por obtener respuestas a preguntas relacionadas con la salud. Como resultado, tanto profesionales de la salud como investigadores de salud 11 pública han reconocido el potencial de fuentes de datos disponibles, como Google Trends, que muestra los patrones de búsqueda de los usuarios. Por último, se analizan las fortalezas, limitaciones y recomendaciones para futuras investigaciones en este ámbito.
(Reem & Amal, 2021) Llevaron a cabo una investigación infodemiológica con el fin de explorar la difusión de temas de discusión en árabe y examinaron el comportamiento de los usuarios en Arabia Saudita. Utilizaron Google Trends y Twitter como fuentes de datos. Asimismo, emplearon los datos proporcionados por Google Mobility para analizar laconexión entre las actividades de los usuarios y las diversas fases de la pandemia, con el objetivo de prever nuevos casos de COVID-19 basándose en los cambios en la movilidad de los usuarios.
(Chire Saire & Pineda-Briseño , 2021), Se llevó a cabo una recopilación de los trabajos publicados entre marzo y julio de 2020, centrándose en la región de América Latina y en técnicas de Minería de Datos aplicadas al Procesamiento de Lenguaje Natural. La cantidad de trabajos realizados es limitada, y hay pocos estudios Infodemiológico en Perú. Las fuentes de datos utilizadas fueron Twitter y Google Trends. El objetivo principal del capítulo de libro es promover la investigación en esta área a nivel de los países latinoamericanos.
(Chire-Saire, Khalid , Oblitas-Cruz, & Ahmed , 2021) Se llevó a cabo un estudio de los países hispanohablantes de Sudamérica utilizando un enfoque infodemiológico y utilizando como fuentes de datos Twitter, Google Trends y los casos diarios de COVID19 reportados por los países. El objetivo fue investigar evidencias sobre el impacto del COVID-19 en la salud mental de la población. Se confirmó que el interés en las 12 búsquedas y publicaciones estaba directamente relacionado con la evolución de la pandemia, y que su disminución sugería una adaptación de las personas al contexto de la pandemia.
Bases teóricas
Infodemiología: La infodemiología es un campo interdisciplinario que investiga la distribución y efectos de la información de salud en la población, mediante el análisis de su propagación en medios digitales como internet y redes sociales. Fusiona la epidemiología, ciencia de la información y tecnología, utilizando herramientas para rastrear y analizar datos de motores de búsqueda, redes sociales y bases de datos de salud. Su objetivo es comprender cómo esta información influye en los comportamientos y decisiones relacionadas con la salud pública.

Deep Learning: Es una rama del machine learning que emplea redes neuronales artificiales con múltiples capas para aprender representaciones de datos de manera jerárquica.
Estas redes pueden aprender características complejas y abstraer información automáticamente a partir de grandes conjuntos de datos. Son especialmente útiles en tareas como reconocimiento de imágenes y procesamiento de lenguaje natural, y tienen aplicaciones en campos como la visión por computadora, la medicina y la robótica.
Procesamiento de Lenguaje Natural (PLN):

4 **sensoricx.com** | Procesamiento de Lenguaje Natural: El poder de la comunicación
https://sensoricx.com/conocimiento/procesamiento-de-lenguaje-natural-nlp/#:~:text=El Procesamiento de Lenguaje Natural ofrece una serie,la interacción y mejora la experiencia del usuario.

es una disciplina de la inteligencia artificial que se centra en la interacción entre las

5 **universo-ia.com** | Procesamiento del lenguaje natural: La IA que comprende y genera texto
https://universo-ia.com/procesamiento-del-lenguaje-natural-la-ia-que-comprende-y-genera-texto/#:~:text=El Procesamiento del Lenguaje Natural, conocido como PLN,, que sea valiosa y comprensible para los human...

computadoras y el lenguaje humano. Su objetivo
es permitir que las máquinas 13 comprendan, interpreten y generen lenguaje humano de manera natural. NLP abarca una variedad de tareas,

6 **sensoricx.com** | Procesamiento de Lenguaje Natural: El poder de la comunicación
https://sensoricx.com/conocimiento/procesamiento-de-lenguaje-natural-nlp/#:~:text=El Procesamiento de Lenguaje Natural ofrece una serie,la interacción y mejora la experiencia del usuario.

como el análisis de sentimientos, la traducción automática, la generación de

7 **thedataschools.com** | RED NEURONAL: ¿QUÉ ES Y PARA QUÉ SE UTILIZA? - BIG DATA
https://thedataschools.com/que-es/red-neuronal/

texto, el reconocimiento de voz y la
extracción de información, y tiene aplicaciones en áreas como la atención al cliente, la salud, la educación y más.
Transformers: Los Transformers son una clase de modelos de aprendizaje profundo que sobresalen por su capacidad para procesar secuencias de datos, como texto o audio, de manera altamente eficiente. Introducidos por primera vez en 2017, estos modelos se basan en mecanismos de atención para capturar relaciones a larga distancia en los datos de entrada. Han demostrado un rendimiento sobresaliente

8 **idus.us.es**
https://idus.us.es/bitstream/11441/155736/1/TFM%20MAES-MUM%20JIMENEZ%20REVUELTA%2c%20JOSE%20CARLOS.pdf

en una variedad de tareas de procesamiento de lenguaje natural

9 **iccsi.com.ar** | Transformers: La Revolución En Procesamiento Del Lenguaje Natural | ICCSI
https://iccsi.com.ar/inteligencia-artificial-transformers/#:~:text=Los transformers han demostrado ser altamente efectivos en,y eficiencia de los sistemas de inteligencia artificial.

, como

10 **Documento de otro usuario**
El documento proviene de otro grupo

la traducción automática, la generación de texto y la clasificación de
texto, y han impulsado avances notables en el campo del aprendizaje automático.
Covid-19: Es una enfermedad infecciosa causada por el coronavirus SARS-CoV-2, que fue identificado por primera vez en Wuhan, China, en diciembre de 2019. Se caracteriza principalmente por síntomas respiratorios, como fiebre, tos y dificultad para respirar, aunque también puede provocar una variedad de síntomas que van desde leves hasta graves

INFORME DE ANÁLISIS
magister

TESIS V2

9% Textos sospechosos

6% Similitudes
0% similitudes entre comillas
0% entre las fuentes mencionadas
4% Idiomas no reconocidos

Nombre del documento: TESIS V2.docx
ID del documento: fdc3ebceade8690f0a7ed263895ce9e8a23c34a0
Tamaño del documento original: 689,29 kB
Autores: []

Depositante: ANIBAL FERNANDO FLORES GARCIA
Fecha de depósito: 1/10/2024
Tipo de carga: interface
fecha de fin de análisis: 1/10/2024

Número de palabras: 11.233
Número de caracteres: 76.219

Ubicación de las similitudes en el documento:



Fuentes de similitudes

Fuentes principales detectadas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	**rodin.uca.es** https://rodin.uca.es/bitstream/10498/24453/5/Optimizacion_SW_NLP-Morales_Millan_Jesus-TFG.pdf 1 fuente similar	1%		Palabras idénticas: 1% (103 palabras)
2	**pythonguia.com** | Scikit Learn matriz de confusión https://pythonguia.com/scikit-learn-matriz-de-confusion/ 3 fuentes similares	< 1%		Palabras idénticas: < 1% (87 palabras)
3	**ciberseguridadmax.com** | ¿Qué es Epoch en Machine Learning? https://ciberseguridadmax.com/epoch/ 7 fuentes similares	< 1%		Palabras idénticas: < 1% (75 palabras)
4	**universo-ia.com** | Evaluación de modelos: Métricas y técnicas para medir el rendi... https://universo-ia.com/evaluacion-de-modelos-metricas-y-tecnicas-para-medir-el-rendimiento/#:~:... 1 fuente similar	< 1%		Palabras idénticas: < 1% (67 palabras)
5	**pythonguia.com** | Scikit Learn Accuracy_score https://pythonguia.com/scikit-learn-accuracy_score/	< 1%		Palabras idénticas: < 1% (69 palabras)

Fuentes con similitudes fortuitas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	**sensoricx.com** | Procesamiento de Lenguaje Natural: El poder de la comunicación https://sensoricx.com/conocimiento/procesamiento-de-lenguaje-natural-nlp/#:~:text=El Procesamie...	< 1%		Palabras idénticas: < 1% (26 palabras)
2	**reunir.unir.net** https://reunir.unir.net//bitstream/123456789/14608/1/Simón Gallego, Carlos.pdf	< 1%		Palabras idénticas: < 1% (26 palabras)
3	**ndmarketingdigital.com** | Guía detallada para entender y explicar la matriz de co... https://ndmarketingdigital.com/como-explicar-la-matriz-de-confusion/	< 1%		Palabras idénticas: < 1% (20 palabras)
4	**prompt.uno** | Todo sobre la Memoria a Corto Plazo Largo - Prompts para IA https://prompt.uno/inteligencia-artificial/todo-sobre-la-memoria-a-corto-plazo-largo/	< 1%		Palabras idénticas: < 1% (20 palabras)
5	**Documento de otro usuario** #96d278 El documento proviene de otro grupo	< 1%		Palabras idénticas: < 1% (22 palabras)

Fuentes mencionadas (sin similitudes detectadas)

Estas fuentes han sido citadas en el documento sin encontrar similitudes.

1 https://aclanthology.org/N19-1423.pdf
2 https://doi.org/10.1162/neco.1997.9.8.1735
3 https://arxiv.org/pdf/1408.5882
4 https://www.who.int/es/health-topics/coronavirus#tab=tab_1

RESOLUCIÓN DE CONSEJO UNIVERSITARIO
N° 039-2026-CU

Moquegua, 28 de enero de 2026

VISTOS:

El Proveído N° 248-2026-R de fecha 22 de enero de 2026 emitido por el Rectorado, Informe N° D0023-2026-UNAM-R-SG de fecha 22 de enero de 2026 emitido por el Secretario General, Informe Técnico N° D0002-2026-SEGE-GT de Grados y Títulos, el Acuerdo del Pleno del Consejo Universitario de Sesión Extraordinaria N° 05-2026-SE-CU-UNAM, de fecha 28 de enero de 2026, y;

CONSIDERANDO:

Que, conforme al 4to párrafo del artículo 18, de la Constitución Política del Perú, dispone que cada Universidad es autónoma en su régimen normativo, de gobierno, académico, administrativo y económico. Y el artículo 8º de la Ley Nº 30220-Ley Universitaria, establece que la referida autonomía inherente a las Universidades, se ejerce de conformidad con la Constitución y demás normativa aplicable. Concordante con el artículo 1, de la Ley Nº 31520, Ley que restablece la autonomía y la institucionalidad de las universidades peruanas, así como la Ley N° 32141, Ley que modifica la Ley N° 30220-Ley Universitaria, a fin de fortalecer la institucionalidad de las Universidades Públicas licenciadas;

Que, el Artículo 58, de la Ley N° 30220-Ley Universitaria, establece que el Consejo Universitario es el máximo órgano de gestión, dirección y ejecución académica y administrativa de la Universidad. Concordante con el artículo 18° del Estatuto de la Universidad Nacional de Moquegua, aprobado bajo Resolución de Asamblea Universitaria N° 015-2025-AU-UNAM;

Que, el artículo 59 de la Ley Universitaria-Ley N° 30220, establece las atribuciones del Consejo Universitario, y específicamente el numeral 59.9, establece; Conferir los grados académicos y los títulos profesionales aprobados por las Facultades y Escuela de Posgrado, así como otorgar distinciones honoríficas y reconocer y revalidar los estudios, grados y títulos de universidades extranjeras, cuando la universidad está autorizada por la Superintendencia Nacional de Educación Superior Universitaria. Concordante con el numeral 14, del artículo 20° del Estatuto de la Universidad Nacional de Moquegua;

Que, el Sr. Rector es el personero y representante legal de la Universidad Nacional de Moquegua, conforme a lo dispuesto por la Ley Nº 30220-Ley Universitaria, tiene a su cargo la Dirección, Conducción y Gestión del Gobierno Universitario en todos sus ámbitos, concordante con el artículo 24° y numeral 2 del artículo 26° del Estatuto de la Universidad Nacional de Moquegua;

Que, mediante Informe Técnico N° D0002-2026-SEGE-GT de fecha 21 de enero de 2026, e Informe N° D0023-2026-UNAM-R-SG de fecha 22 de enero de 2026, el Secretario General de la UNAM, remite al Rectorado los expedientes para el trámite de Título Profesional de Ingeniero de Sistemas e Informática a los Bachilleres: RAMÍREZ CHIPANA MARÍA DEL PILAR LAURA, FLOR VELASQUEZ ANTHONY ROGER y RAMOS CALLA PEDRO LUIS de la Escuela Profesional de Ingeniería de Sistemas e Informática, quienes se encuentran aptos para la expedición del Título Profesional de Ingeniero de Sistemas e Informática, al haber cumplido satisfactoriamente con los requisitos legales y reglamentarios, exigidos en el Reglamento de Grados y Títulos de la Universidad Nacional de Moquegua, conforme lo señala la Dirección de Actividades y Servicios Académicos y la Dirección de la Escuela Profesional de Ingeniería de Sistemas e Informática, razón por la cual, el señor Secretario General incluye dichos expedientes como punto de agenda para ser tratado en Sesión Extraordinaria de Consejo Universitario de fecha 28 de enero de 2026;

Que, el artículo 44° de la Ley N° 30220, Ley Universitaria señala que las universidades otorgan los grados académicos de Bachiller, Maestro, Doctor y los Títulos Profesionales que correspondan, a nombre de la Nación (...). Asimismo, la Ley N° 31803, modifica el artículo en mención adicionando; La Superintendencia Nacional de Educación Superior Universitaria (SUNEDU) determina los criterios, así como establece y supervisa el procedimiento administrativo para el reconocimiento de grados y títulos otorgados en el extranjero, en el marco de la normativa vigente;

Que, el numeral 45.2 del artículo 45° de la Ley Nº 30220, Ley Universitaria, precisa los requisitos mínimos para la obtención del título profesional, como son: el grado de Bachiller y la aprobación de una tesis o trabajo de suficiencia profesional. Asimismo, establece que el título profesional solo se puede obtener en la universidad en la cual se haya obtenido el grado de bachiller;

Que, mediante Resolución de Comisión Organizadora N° 1153-2021-UNAM, se aprobó el Reglamento de Grados y Títulos de la Universidad Nacional de Moquegua, el mismo que fue modificado a través de las Resoluciones de Comisión Organizadora N° 410; 453 y 890-2022, N° 880-2023-UNAM y N° 081, 484-2024-UNAM el cual establece los requisitos y norma

Domicilio Legal: Prolongación Av., Ancash S/N - Moquegua
RUC: 20449347448 - Teléfono: (053) 670168
Página Web: www.unam.edu.pe

RESOLUCIÓN DE CONSEJO UNIVERSITARIO
N° 039-2026-CU

Moquegua, 28 de enero de 2026

los procedimientos para la obtención de los grados académicos y títulos profesionales conferidos por esta Casa Superior de Estudios;

Que, bajo el amparo de los considerandos precedentes, el Pleno del Consejo Universitario en Sesión Extraordinaria N° 05-2026-SE-CU-UNAM, de fecha 28 de enero de 2026, emitió, el siguiente, Acuerdo, POR UNANIMIDAD: CONFERIR, a nombre de la Nación el Título Profesional de Ingeniero de Sistemas e Informática por la modalidad de sustentación de tesis a los Bachilleres: RAMÍREZ CHIPANA MARÍA DEL PILAR LAURA, FLOR VELASQUEZ ANTHONY ROGER y RAMOS CALLA PEDRO LUIS, de la Escuela Profesional de Ingeniería de Sistemas e Informática, de la Facultad de Ingenierías de la Universidad Nacional de Moquegua;

En uso de las facultades y atribuciones conferidas por el Art. 18, de la Constitución Política del Perú, la Ley N° 30220 Ley Universitaria, el Estatuto de la Universidad Nacional de Moquegua, y a lo acordado por el pleno del Consejo Universitario:

SE RESUELVE:

ARTÍCULO 1°: CONFERIR, A NOMBRE DE LA NACIÓN a los Bachilleres

- **RAMÍREZ CHIPANA MARÍA DEL PILAR LAURA**
  TESIS : "Sistema de recomendación de jurisprudencia basado en embeddings de texto para la identificación y sugerencia de precedentes relevantes de las Cortes Superiores de Justicia de la República del Perú", cuyo Asesor Principal fue el Mgr. Hugo Euler Tito Chura, y el Co - asesor fue el Ing. Honorio Apaza Alanoca.
- **FLOR VELASQUEZ ANTHONY ROGER**
  TESIS : "Sistema de recomendación de jurisprudencia basado en embeddings de texto para la identificación y sugerencia de precedentes relevantes de las Cortes Superiores de Justicia de la República del Perú", cuyo Asesor Principal fue el Mgr. Hugo Euler Tito Chura, y el Co - asesor fue el Ing. Honorio Apaza Alanoca.
- **RAMOS CALLA PEDRO LUIS**
  TESIS : "Modelos de Deep learning para la detección de noticias falsas en español sobre COVID-19", cuyo Asesor Principal fue el Dr. Anibal Fernando Flores García.

El **TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS E INFORMÁTICA.**

Por la modalidad de SUSTENTACIÓN DE TESIS, de la Escuela Profesional de Ingeniería de Sistemas e Informática, de la Facultad de Ingenierías de la Universidad Nacional de Moquegua.

ARTÍCULO 2°: ENCARGAR, al Secretario General a través de quien corresponda, para que realice las acciones pertinentes relacionadas con la inscripción correspondiente en el Registro de Grados y Títulos de la Superintendencia Nacional de Educación Superior Universitaria - SUNEDU.

ARTÍCULO 3°: DISPONER, el cumplimiento de la presente Resolución a todas las dependencias de la Universidad Nacional de Moquegua.

ARTÍCULO 4°: DISPONER la publicación de la presente resolución en el Portal Institucional de la Universidad Nacional de Moquegua www.unam.edu.pe

REGÍSTRESE, PUBLÍQUESE Y CÚMPLASE;

UNIVERSIDAD NACIONAL DE MOQUEGUA SECRETARIA GENERAL

ABOG. EDGAR PINTO HUANCAVILCA
SECRETARIO GENERAL

UNIVERSIDAD NACIONAL DE MOQUEGUA CONSEJO UNIVERSITARIO

DR. HUGO RUBEN MARCA MAQUERA
RECTOR
PRESIDENTE DEL CONSEJO UNIVERSITARIO

DISTRIBUCIÓN:
VRA VRI FACS EPGPDS GyT DASA OTI Interesado c.c. Arch.

Domicilio Legal: Prolongación Av., Ancash S/N - Moquegua
RUC: 20449347448 - Teléfono: (053) 670168
Página Web: www.unam.edu.pe