UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TITULO DEL TRABAJO DE TITULACIÓN

APLICACIÓN DE MODELOS TRANSFORMERS PARA CLASIFICAR TEXTOS EN IDIOMA ESPAÑOL

AUTOR

MERCHÁN PÉREZ ERICK LENIN

MODALIDAD DE TITULACIÓN
EXAMEN COMPLEXIVO

Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

TUTOR

ING. LÍDICE VICTORIA HAZ LÓPEZ, Msi.

Santa Elena, Ecuador

Año 2024

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

Ing. José Sánchez Aquino. Mgt.
DIRECTOR DE LA CARRERA

Ing. Lidice Haz López, Msi.
TUTOR

Ing. Shendry Rosero Vásquez, Ms.CC
DOCENTE ESPECIALISTA

Ing. Mónica Jaramillo Infante, Mgt.
DOCENTE GUÍA UIC

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por **MERCHÁN PÉREZ ERICK LENIN**, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 18 días del mes de junio del año 2024

TUTOR

Firmado electrónicamente por:
LIDICE VICTORIA HAZLOPEZ

ING. LIDICE VICTORIA HAZ LÓPEZ, Msi.

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **ERICK LENIN MERCHÁN PÉREZ**

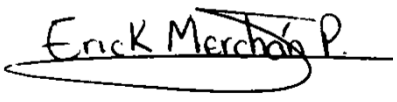
DECLARO QUE:

El trabajo de Titulación, Aplicación de modelos Transformers para clasificar textos en idioma español previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

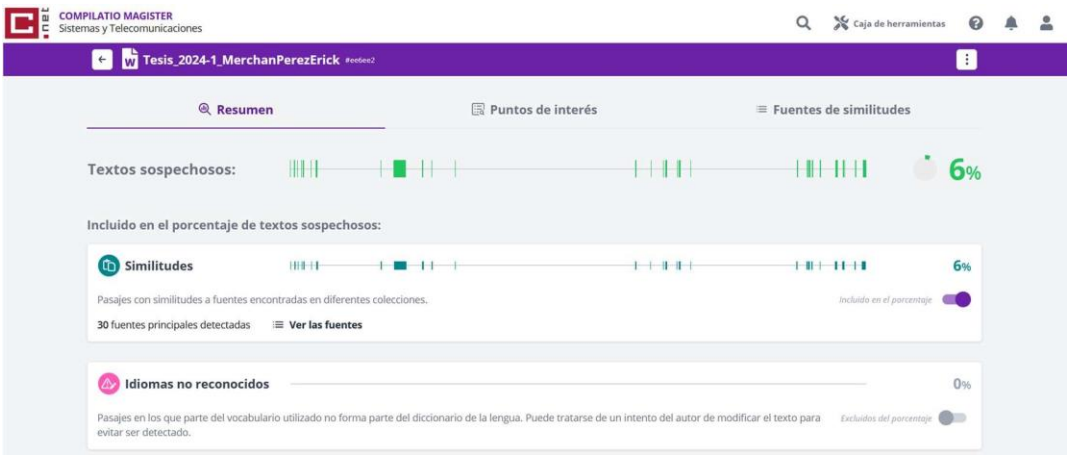
La Libertad, a los 18 días del mes de Junio del año 2024

EL AUTOR



ERICK LENIN MERCHÁN PÉREZ

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
CERTIFICACIÓN DE ANTIPLAGIO



Certifico que después de revisar el documento final del trabajo de titulación denominado "**APLICACIÓN DE MODELOS TRANSFORMERS PARA CLASIFICAR TEXTOS EN IDIOMA ESPAÑOL",** presentado por el estudiante, **MERCHÁN PÉREZ ERICK LENIN** fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 6%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.

TUTOR

Firmado electrónicamente por:
LIDICE VICTORIA HAZ LOPEZ

ING. LIDICE VICTORIA HAZ LÓPEZ, Msi.

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

AUTORIZACIÓN

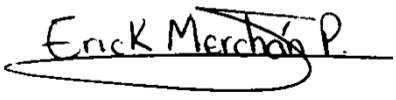
Yo, **ERICK LENIN MERCHÁN PÉREZ**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales del trabajo de titulación con fines de difusión pública, dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

La Libertad, a los 18 días del mes de Junio del año 2024

EL AUTOR



ERICK LENIN MERCHÁN PÉREZ

AGRADECIMIENTO

Quiero expresar mi más profundo agradecimiento primero a Dios que me ha mantenido firme por todo este camino, a la Universidad por darme la oportunidad de convertirme en un profesional de la sociedad, a mis docentes que escogido esta profesional para brindarme su conocimiento y a mi tutora que siempre estuvo apoyando desde el inicio hasta la finalización del proyecto.

Erick Lenin Merchán Pérez

DEDICATORIA

Y le dedico esta tesis a toda mi familia y en especial a mis padres que me apoyado a lo largo de todo este tiempo, realizando un esfuerzo aun mayor sacrificándose día a día para brindarme una vida y educación de calidad.

Erick Lenin Merchán Pérez

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

La llegada de los modelos Transformers ha revolucionado el procesamiento del lenguaje natural (PLN) al introducir un innovador mecanismo de atención capaz de capturar de manera eficiente y simultánea dependencias a largo plazo en secuencias de datos. Este avance arquitectónico ha generado un camino para un progreso significativo en diversas aplicaciones de PLN. En consecuencia, el enfoque de este proyecto radica en aprovechar estos modelos Transformers Pysentimiento para la clasificación de texto en el idioma español. Para lograr este objetivo, se emplearán cuatro modelos distintos: BERT, RoBERTa, DistilBERT y AlBERT, utilizando un conjunto de datos obtenido de la plataforma en línea Kaggle.

Estos conjuntos de datos serán sometidos a un procesamiento previo y posteriormente alimentados a cada modelo para su evaluación. Se llevará a cabo un análisis comparativo de los resultados utilizando diversas métricas, y los hallazgos experimentales revelarán que, después de un adecuado preprocesamiento, el modelo DistilBERT alcanzo una precisión del 78%, un recall del 75%, una exactitud del 75% y una puntuación f1 del 76%. Este resultado establece al modelo DistilBERT como la opción más adecuada para tareas de clasificación de texto en el idioma español.

Palabras claves: PLN, Transformers, Kaggle, BERT, RoBERTa, DistilBERT, AlBERT

ABSTRACT

The advent of Transformer models has revolutionized natural language processing (NLP) by introducing an innovative attention mechanism capable of efficiently and simultaneously capturing long-term dependencies in data sequences. This architectural breakthrough has paved the way for significant progress in various NLP applications. Consequently, the focus of this project lies in leveraging these Transformer models, specifically Pysentimiento, for text classification in Spanish. To achieve this objective, four different models will be employed: BERT, RoBERTa, DistilBERT, and AlBERT, using a dataset obtained from the online platform Kaggle.

These datasets will undergo preprocessing and subsequently be fed into each model for evaluation. A comparative analysis of the results will be conducted using various metrics, and the experimental findings will reveal that, after appropriate preprocessing, the DistilBERT model achieved an accuracy of 78%, a recall of 75%, a precision of 75%, and an F1 score of 76%. This result establishes the DistilBERT model as the most suitable option for text classification tasks in Spanish.

Keywords: PLN, Transformers, Kaggle, BERT, RoBERTa, DistilBERT, AlBERT

INTRODUCCIÓN

Las redes sociales se han convertido en los nuevos espacios virtuales en los que no solo construimos nuestra identidad, además también buscamos relacionarnos. También la usamos como filtro ajustando la cantidad de información que recibimos en función a nuestras preferencias e intereses, la impresión que ha causado las redes sociales ha sido tanta que se ha impuesto un nuevo modelo de relación en donde la presencia física no es necesaria. El uso del internet ha alterado el rumbo en el que percibíamos nuestras vidas, lo que ha llevado a realizarse números estudios para constatar cómo influye ante las emociones.[1]

Un fenómeno que se ha observado en las redes sociales es el de la expresión emocional de los usuarios, que se manifiesta de diversas formas, como reacciones, fotos, publicaciones y comentarios. Estos últimos son especialmente interesantes, ya que pueden revelar aspectos de la personalidad, el estado de ánimo, las intenciones o las opiniones de quien los escribe. Sin embargo, no siempre es fácil interpretar lo que una persona quiere o está tratando de comunicar a través de un comentario, y mucho menos lo que sienten o piensan los millones de usuarios que utilizan este medio tecnológico.[2]

El tema propuesto en este proyecto esta alineado a los objetivos del Plan Nacional de Desarrollo específicamente los siguientes ejes:

Eje 1: Derechos para Todos Durante Toda la Vida

Objetivo 1: Garantizar una vida digna con iguales oportunidades para todas las personas.[3]

Política 1.10 Erradicar toda forma de discriminación y violencia por razones económicas, sociales, culturales, religiosas, etnia, edad, discapacidad y movilidad humana, con énfasis en la violencia de género y sus distintas manifestaciones. [3]

Objetivo 2: Afirmar la interculturalidad y plurinacionalidad, revalorizando las identidades diversas. [3]

Política 2.1 Erradicar la discriminación y la exclusión social en todas sus manifestaciones, especialmente el machismo, la homofobia, el racismo, la xenofobia y otras formas conexas, mediante acciones afirmativas y de reparación integral para la construcción de una sociedad inclusiva. [3]

Eje 2: Economía al Servicio de la Sociedad

Objetivo 5: Impulsar la productividad y competitividad para el crecimiento económico sostenible de manera redistributiva y solidaria. [3]

Política 5.6 Promover la investigación, la formación, la capacitación, el desarrollo y la transferencia tecnológica, la innovación y el emprendimiento, la protección de la propiedad intelectual, para impulsar el cambio de la matriz productiva mediante la vinculación entre el sector público, productivo y las universidades. [3]

CAPÍTULO 1. FUNDAMENTACIÓN

ANTECEDENTES

La tecnología que permite analizar el lenguaje natural y detectar las emociones, actitudes y opiniones de las personas que se comunican en línea tiene múltiples aplicaciones en diferentes sectores. En el ámbito empresarial, se puede utilizar para mejorar la atención al cliente, la gestión de recursos humanos, la innovación y la competitividad. En lo organizacional se puede usar para fomentar la colaboración, la participación, la transparencia y la confianza.

En el ámbito científico, se puede emplear para investigar los fenómenos sociales, psicológicos y culturales que influyen en la comunicación digital. Asimismo, esta tecnología puede contribuir a prevenir y combatir los comportamientos asociales o negativos que pueden afectar la convivencia, el respeto y la seguridad en línea, tales como el acoso, el odio, la desinformación o la manipulación.[4]

En el proyecto de titulación de Joshua Augusto Ávila Vacas y María Belén Pérez Pilco, se trabajó la identificación de comentarios sexistas en español de la red social Twitter utilizando el modelo Transformers para procesamiento de lenguaje natural, donde se pudo determinar que entre los métodos más utilizados para detectar sexismo en textos por mensajes se encuentra las redes neuronales Transformers con los modelos BERT, RoBERTa y el algoritmo de aprendizaje supervisado SVM. Del mismo modo se eligió el modelo pre-entrenado Pysentimiento de Transformers por ser este el mejor valorado y adaptado para la clasificación de texto en idioma español.[5]

En el trabajo realizado por Ricardo Manuel Lazo Vásquez con el tema de clasificación de la personalidad utilizando Procesamiento de Lenguaje Natural y Aprendizaje Profundo para detectar patrones de notas de suicidio en redes sociales, se comprobó la

eficiencia de los modelos Transformers para el trabajo de clasificación de personalidad. Primero se tomó un listado de varias modelos Transformers y del modelo de personalidad MBTI para clasificar texto escrito por computador de forma automática.[6]

Luego se tomó métricas de exactitud, F1-score y AUC para aislar a los mejores candidatos con respecto al conjunto de datos SND tomando como punto de comparación la similitud del perfil de una persona con antecedentes de suicidio. Obteniendo como resultado siete modelos Transformers que puede cumplir con la tarea de detectar patrones de personalidad orientada a suicidio, destacando el modelo Distil RoBERTa el cual obtuvo una exactitud de 57.44% en el resultado de validación del rendimiento.[6]

En el trabajo desarrollado por Sara de Antón Santiago, propone una aplicación de reconocimiento de emociones para personas vulnerables ante la soledad, donde da a conocer su interés por el uso de la inteligencia artificial como bien social, apoyando la lucha contra la soledad y secundar el bienestar emocional con el objetivo de mejorar la calidad de vida de personas vulnerables. Al término del trabajo se pudo determinar las mejores herramientas y modelos para el reconocimiento de emociones y voz.[7]

A partir de la creación de su propio Dataset en español se implementó un sistema de reconocimiento de emociones utilizando un modelo ya creado para luego integrarlo en una aplicación que proporciona respuestas de apoyo emocional utilizando ChatGPT. Para finalmente notificar a los contactos de los usuarios sobre su estado emocional por medio de la aplicación de mensajería instantánea Telegram. Concluyendo así con el trabajo por medio de los resultados obtenidos que es posible implementar un sistema de reconocimiento de emociones por voz con el empleo de modelos de lenguaje como ChatGPT.[7]

DESCRIPCIÓN DEL PROYECTO

El proyecto se fundamenta en el estudio del modelo de aprendizaje profundo Transformers para el procesamiento del lenguaje natural (NLP). El objetivo es la clasificación de características en texto en idioma español usando los modelos pre-entrenado Transformers Pysentimiento. Para lo cual, será necesario realizar una investigación bibliográfica de trabajos previos en el campo del aprendizaje automático, referidos al análisis de emociones en texto; además de conceptualizar el funcionamiento y arquitectura de los modelos Transformers.

El estudio se desarrolla usando la metodología de investigación explicativa y experimental. Para lo cual, se observará el efecto de usar los modelos de la arquitectura Transformers comprándolos entre sí sobre la detección de características identificadas en el texto. Asimismo, se realizarán pruebas experimentales usando los métodos automáticos de modelos pre-entrenados de Transformers. Para realizar estas pruebas, se utilizará Datasets en idioma español extraídos desde la plataforma especializada en la competencia de ciencia de datos Kaggle.

Finalmente, los resultados se presentarán en tablas comparativas que mostrarán las métricas de predicción y probabilidad de cada uno de los modelos para evaluar cuál es el más adecuado para la clasificación de texto en español. Esta investigación ayudará a demostrar que es posible crear sistemas o servicios para la detección de emociones utilizando modelos Transformers. El uso de esta tecnología puede ser valioso en diversos ámbitos, como el empresarial, organizacional y científico, donde se desee monitorear y prevenir comportamientos asociales o negativos identificados en discursos o textos en línea.

OBJETIVOS DEL PROYECTO

OBJETIVO GENERAL

- Aplicar modelos Transformers de aprendizaje profundo analizando mensajes de textos para el Procesamiento del Lenguaje Natural que permita clasificar el texto con base a características predefinidas en idioma español.

OBJETIVOS ESPECIFICOS

- Describir el funcionamiento de las técnicas aprendizaje automático utilizado en la clasificación de textos cortos.
- Aplicar métodos de aprendizaje supervisado para clasificar textos cortos mediante un modelo Transformers pre-entrenado.
- Evaluar los resultados mediante el análisis de métricas de rendimiento de los modelos de aprendizaje profundo.

JUSTIFICACIÓN

Este proyecto de investigación se ha originado debido al aumento de problemas relacionados con la violencia de género en las redes sociales online. Con el avance de las Tecnologías de la Información, las formas de comunicación y relación entre los jóvenes y adultos han experimentado un gran cambio, esto ha llevado a una mayor interconexión y acceso a las redes sociales, que se han convertido en un medio cada vez más prevalente para compartir información, ideas y expresar sentimientos. Sin embargo, esta popularidad ha traído consigo una serie de problemas, proliferación de contenidos ofensivos, el acoso en línea y la difusión de noticias falsas.[5]

Las redes sociales son una herramienta poderosa para conectar con otras personas, informarse y expresarse. Sin embargo, también pueden ser un espacio de hostilidad, intolerancia y agresión hacia quienes tienen una gran visibilidad o influencia. Según la fundación MAPFRE, las redes sociales han generado un fenómeno de odio o hate que afecta especialmente a los influencers. Por eso, nuestro proyecto tiene un enfoque social, que busca analizar y mostrar el nivel de odio y otros indicadores en las personas que más impactan en las redes sociales.[8]

La personalidad oscura se refiere a un conjunto de rasgos que implican una falta de empatía, una tendencia a la manipulación y una actitud hostil hacia los demás. Estas personas suelen expresar comentarios cargados de odio e insensibilidad, que revelan su desprecio por el bienestar ajeno. Según algunos estudios, las personas con personalidad oscura comparten una visión egoísta y maliciosa de las relaciones interpersonales, que los lleva a aprovecharse de los demás y a causarles daño.[9]

ALCANCE DEL PROYECTO

Este proyecto tiene como objetivo aplicar modelos Transformers para la clasificación de texto en español. Se utilizan cuatro modelos específicos: BERT, RoBERTa, DistilBERT y ALBERT. Se realiza una investigación detallada y se explica su funcionamiento en el contexto del procesamiento de lenguaje natural. Los modelos se entrenan y evalúan utilizando conjuntos de datos en español obtenidos de la plataforma Kaggle.

Como lenguaje de programación se emplea Python, y Google Colab como entorno de trabajo. Durante las pruebas, se ajustan los parámetros de aprendizaje de los cuatro modelos para que satisfagan los objetivos del proyecto. Los resultados se presentan en tablas comparativas, facilitando la evaluación de cuál modelo ofrece las mejores características para la detección de emociones en textos.

CAPÍTULO 2. MARCO TEÓRICO Y METODOLOGÍA DEL PROYECTO

MARCO CONCEPTUAL

Inteligencia Artificial

"La inteligencia artificial es una disciplina que combina la ciencia y la ingeniería para estudiar, desde una perspectiva computacional, los procesos y fenómenos que se asocian con la inteligencia. Su objetivo es diseñar y construir sistemas que puedan mostrar este tipo de comportamiento."[10]

Maching Learning

El aprendizaje automático es el proceso por el cual una máquina mejora su desempeño al analizar datos y extraer patrones. Para ello, utiliza un software que se basa en técnicas de inteligencia artificial que simulan el aprendizaje humano, como la práctica y la retroalimentación.[11]

Aprendizaje supervisado

El algoritmo aprende a hacer su tarea usando datos que tienen una etiqueta o una categoría basada en un criterio o una idea específica. Así, puede descubrir patrones que le sirven para analizar y producir un resultado que ya se espera. [12]

Aprendizaje no supervisado

Un modelo predictivo que se entrena de forma parecida al aprendizaje supervisado, pero que no usa datos clasificados o etiquetados, se llama aprendizaje no supervisado. Este tipo de aprendizaje busca encontrar patrones que agrupen los datos según su similitud.[12]

Aprendizaje reforzado

Es una forma de inteligencia artificial que no requiere entrenamiento con datos etiquetados o no etiquetados; el sistema aprende en un entorno donde no hay retroalimentación sobre la salida esperada mediante acciones y los resultados obtenidos, como también señalan, el modelo se fortalece al resolver el problema de la mejor forma. [12]

Deep Learning

En el campo de la inteligencia artificial, el aprendizaje profundo es una técnica que utiliza múltiples capas de unidades de procesamiento no lineales para extraer características de alto nivel de los datos. Entre los tipos más populares de aprendizaje profundo se encuentran las redes neuronales profundas (DNN) y las redes neuronales convencionales (CNN), que han mostrado una gran capacidad para resolver problemas complejos de visión por computadora y reconocimiento de patrones, superando a los métodos clásicos de aprendizaje automático.[13]

Procesamiento de Lenguaje Natural - PLN

El procesamiento del lenguaje natural (PNL) consiste en capacitar a las computadoras para entender, interpretar y producir el lenguaje humano a través de algoritmos de aprendizaje automático como las redes neuronales, los árboles de decisión y las máquinas de soporte vectorial.[14]

PySentimiento

PySentimiento es una biblioteca de código abierto diseñada para el análisis de sentimientos y otras tareas de Procesamiento de Lenguaje Natural en redes sociales. Emplea modelos basados en Transformers para realizar análisis de sentimiento y emoción en español e inglés, además de detectar discursos de odio y reconocer entidades nombradas en ambos idiomas.[5]

Hugging Face

Hugging Face es un destacado startup en el campo del Procesamiento del Lenguaje Natural (NLP) dentro de la ciencia de datos, conocida por sus bibliotecas de código abierto respaldadas por PyTorch y TensorFlow. Estas bibliotecas ofrecen una amplia gama de modelos pre-entrenados que pueden ser ajustados según necesidades específicas.[15]

Kaggle

Kaggle es una plataforma en línea donde científicos de datos se reúnen para participar en competiciones de machine learning y acceder a conjuntos de datos para análisis.[16]

METODOLOGÍA DEL PROYECTO

A continuación, se describen los pasos aplicados para el desarrollo del proyecto:

El proyecto se estructura en varias fases. En la Fase 1, se adquieren los datos del repositorio de Kaggle, aprovechando conjuntos de datos en español y de acceso abierto. Estos datos, que contienen textos a etiquetar para su clasificación, se someterán a un proceso de limpieza y preprocesamiento para garantizar su calidad y facilitar el entrenamiento del modelo.

En la Fase 2, el texto se preprocesa mediante diversas técnicas, como la conversión a minúsculas, la eliminación de signos de puntuación y números, así como la tokenización de palabras individuales y la eliminación de palabras irrelevantes como las stopwords.

En la Fase 3, se desarrolla el modelo utilizando Google Colab como entorno de desarrollo y se utilizan diversas bibliotecas como pandas, numpy, matplotlib, tensorflow, keras y pytorch para el preprocesamiento de datos y la aplicación de los modelos. Se instala la biblioteca "Transformers" y se incorporan modelos pre-entrenados como BERT, DistilBERT, RoBERTa y AlBERT.

En la Fase 4, se extraen características de los textos mediante modelos de lenguaje pre-entrenados incorporados en la fase anterior. Estos modelos permiten obtener representaciones vectoriales de los textos, capturando su semántica y estructura. Se emplean algoritmos de aprendizaje automático supervisado para asignar etiquetas a cada texto según su contenido.

Finalmente, en la Fase 5, se evalúa el modelo utilizando métricas como precisión, recall, F1-score y accuracy. Se emplea la matriz de confusión para visualizar el desempeño del modelo en términos de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos, lo que proporciona una comprensión integral de su rendimiento.

METODOLOGIA DE INVESTIGACIÓN

El estudio se desarrollará siguiendo una metodología de investigación explicativa y experimental, lo que implica un enfoque riguroso para comprender y evaluar el efecto de diferentes direcciones de modelado en la detección de características en el texto. En este sentido, se llevará a cabo un análisis comparativo entre los modelos de la arquitectura Transformers, con el objetivo de explorar y entender cómo cada enfoque aborda la tarea de identificar emociones en el contenido textual.

Este enfoque metodológico nos permitirá no solo evaluar la efectividad relativa de cada modelo en la tarea propuesta, sino también comprender mejor los mecanismos subyacentes que impulsan su desempeño y discernir las posibles ventajas y limitaciones de cada enfoque en el contexto específico de la detección de emociones.

TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Como técnica para el proceso de recopilación de los datos esenciales para este proyecto, nuestra estrategia se enfocará exclusivamente en la adquisición de conjuntos de datos provenientes de fuentes abiertas disponibles en la web. En particular, optaremos por

descargar estos conjuntos de datos de plataformas reconocidas y confiables como Kaggle, que ofrece una amplia variedad de conjuntos de datos de alta calidad para una variedad de dominios y aplicaciones.

Para el desarrollo de los modelos propuestos, se emplea un conjunto de datos que recoge comentarios de redes sociales, clasificándolos en seis tipos diferentes de emociones. Este Dataset, que contiene dos columnas (Emociones, Texto). Este Dataset está disponible y se puede acceder a el desde: https://www.kaggle.com/datasets/ishantjuyal/emotions-in-text/data

CAPÍTULO 3. PROPUESTA

ARQUITECTURA DEL SISTEMA

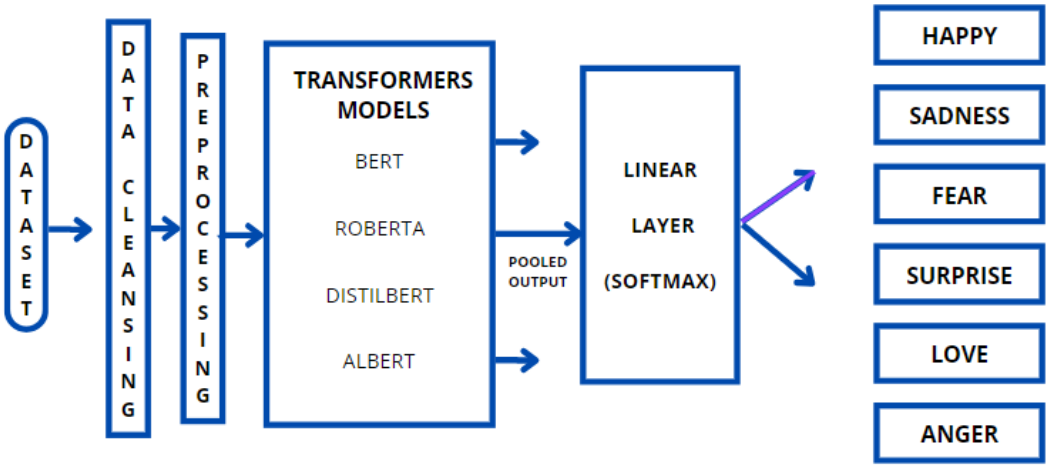


Figura 1. *Arquitectura del Sistema. Elaboración propia*

• **Conjunto de datos:** Para la obtención de datos, se recurrirá a un conjunto de datos descargado de fuentes en línea, el cual será vinculado al proyecto desde Google Drive para su uso posterior. Este enfoque garantiza la disponibilidad de datos relevantes y actualizados para el análisis y modelado del proyecto.

• **Limpieza de datos:** Una parte esencial del proceso de preparación de datos, conocida como data cleansing, se llevará a cabo en esta etapa. Aquí, se procederá a eliminar palabras, símbolos y emoticones que no aportan valor al análisis o que podrían introducir ruido en los resultados. Este paso asegura la coherencia y la calidad de los datos antes de proceder a su procesamiento.

• **Preprocesamiento:** Tras la limpieza de datos, se llevará a cabo una fase de preprocesamiento para preparar los datos de manera óptima para su posterior análisis. Durante esta etapa, se visualizarán y se explorarán las características de los datos para

comprender mejor su estructura y distribución, lo que facilitará la selección de técnicas de modelado apropiadas.

• **Modelado:** En esta fase, se realizará la construcción y el entrenamiento de diferentes modelos de la arquitectura Transformers. Estos modelos serán fundamentales para la tarea de clasificación de datos, y se explorarán diversas configuraciones y parámetros para obtener los mejores resultados posibles.

• **Salida:** Utilizando la capa de softmax como parte del proceso de modelado, se generará una salida múltiple que permitirá identificar y clasificar las diferentes características presentes en los datos procesados. Este paso final proporcionará información valiosa para comprender las tendencias y los patrones dentro de los datos analizados.

PASOS DE PREPOCESAMIENTO DE TEXTO

Para el procesamiento de los datos, es necesario realizar una limpieza previa que elimine o corrija los valores erróneos, incompletos o irrelevantes. De esta forma, se obtienen datos más puros y de mayor calidad, que facilitan el entrenamiento del modelo y mejoran la precisión de los resultados, a este proceso se lo conoce como tokenización. Existen diversas técnicas para realizar esta limpieza, que se explicarán a continuación:

Conversión a minúsculas: La función tiene como objetivo convertir una cadena de texto en español. Esta herramienta es muy útil para estandarizar texto o para comparaciones que no distinguen entre mayúsculas y minúsculas ***nuevo_texto = texto.lower().*** [17]

Eliminación de signos de puntuación: Se dará un conjunto de caracteres en una expresión regular y lo que hará la función es buscar cada uno de estos y remplazarlos por espacios en blancos ***nuevo_texto = re.sub(regex , ' ', nuevo_texto).*** [17]

Eliminación de números: La siguiente expresión regular busca remplazar todos los dígitos en el texto por espacios en blanco y se almacenara en la variable **nuevo_texto. nuevo_texto = re.sub("\d+", ' ', nuevo_texto).**[17]

Eliminación de espacios en blancos múltiples: Este código utiliza una expresión regular para reemplazar múltiples espacios en blanco en el texto por un solo espacio en blanco ***nuevo_texto = re.sub("\\s+", ' ', nuevo_texto).*** [17]

Tokenización de palabras individuales: La siguiente expresión utiliza el método **Split()** para dividir el texto almacenado en la variable **nuevo_texto** en una lista de palabras, utilizando el espacio en blanco como separador ***nuevo_texto = nuevo_texto.split(sep = ' ').*** [17]

Eliminación de tokens con longitud: Este código utiliza una comprensión de listas para filtrar los elementos de la lista y retener solo aquellos tokens cuya longitud sea mayor que 1 eliminando así palabras de una sola letra como artículos, preposiciones, etc ***nuevo_texto = [token for token in nuevo_texto if len(token) > 1].*** [17]

Eliminación de stop_words: Para este modo de limpieza de datos necesitamos antes crear un diccionario en el cual se encontrará una lista de palabras que usaremos como filtro y eliminaremos del Dataset aquellas se encuentren en la lista descartando así palabra que no tengan un contexto o valor para el modelo ***nuevo_texto = [token for token in nuevo_texto if token not in stop_words].*** [17]

EXTRACCIÓN DE CARACTERÍSTICAS

Los Transformers revolucionaron el Procesamiento del Lenguaje Natural desde su creación, introduciendo dos propuestas innovadoras: los mecanismos de self-attention para mantener el contexto de las palabras generadas a largo plazo y las representaciones de posición para cada palabra ingresada al modelo. Estas características permiten procesar el texto en paralelo sin afectar el orden de la oración. Presentado originalmente en 2017, el modelo incluye codificadores para las entradas de texto y decodificadores para los textos generados.[18]

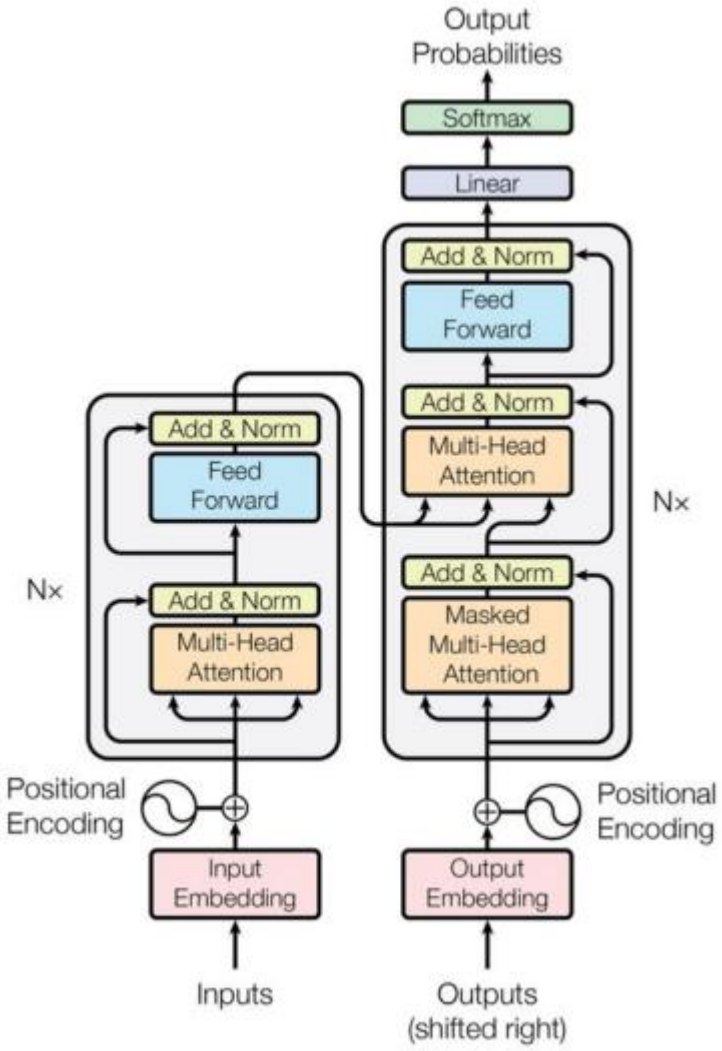


Figura 2. *Arquitectura del Modelo Transformers. [18]*

Modelo BERT

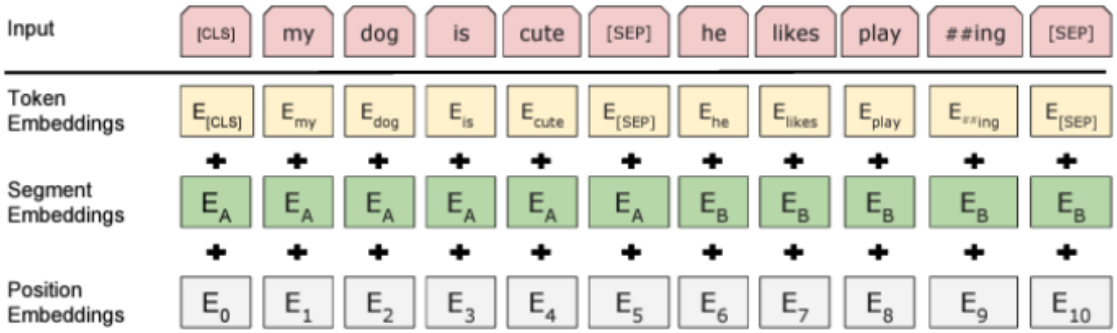


Figura 3. *Funcionamiento del proceso de entrada al modelo BERT.[19]*

La Figura 2 muestra cómo se procesan las oraciones para la entrada al modelo BERT. El proceso comienza con los "Inputs", que son los textos que se ingresarán al modelo, añadiendo tokens especiales como "CLS" para marcar el inicio y "SEP" para la separación en la oración. Los "Token Embeddings" representan las entradas textuales en forma de vectores.

Los "Segment Embeddings" indican la segmentación de las frases dentro de la oración, asignando el grupo "A" a una frase y el grupo "B" a otra, ambas dentro de la misma oración. Finalmente, los "Position Embeddings" señalan las posiciones relativas de cada palabra en la oración. La suma de estos tres tipos de "Embeddings" constituye la entrada al modelo BERT.[20]

Salidas de codificación de BERT

De acuerdo con la especificación de Hugging Face, los resultados de un modelo consisten en salidas de datos estructuradas que contienen la información generada por el modelo. Esta información puede organizarse en tuplas o diccionarios para su posterior utilización. Dentro de cada modelo Transformer, se identifican tres tipos fundamentales de salidas. Tomando como ejemplo a BERT, se pueden destacar las siguientes:[20]

- Output [0] Last hidden state: El último estado oculto es la representación contextualizada final de cada token, refiriéndose al estado oculto final asociado con cada token de entrada, después de pasar por las 12 capas del modelo BERT. [21]

- Output [1] Pooler output: La salida de la capa de "Pooling" es una representación incorporada del texto completo, comúnmente utilizada para tareas de clasificación de texto o para obtener representaciones fijas del texto. [21]

- Output [2] Hidden state: El estado oculto consiste en las 12 capas de atención presentes en el modelo BERT. Dependiendo del modelo, el número de capas puede variar. Cada una de estas capas tiene su propio estado oculto, que contiene información sobre la representación contextual de los textos ingresados.[21]

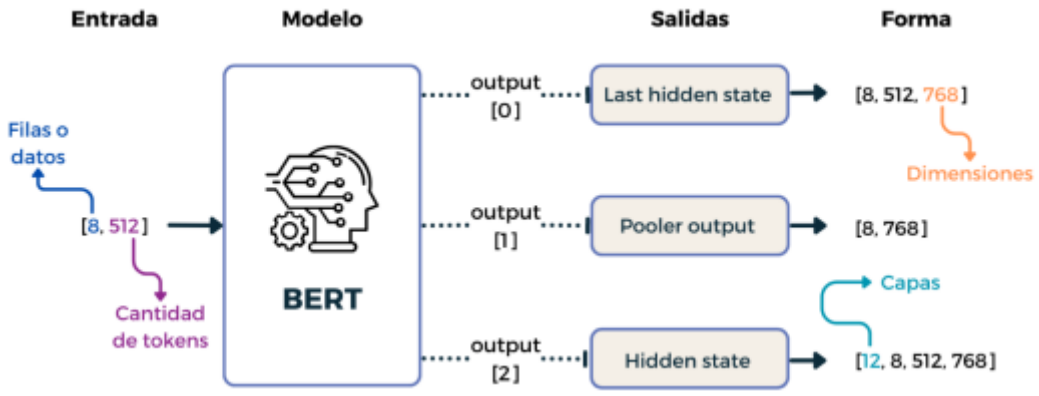


Figura 4. *Representación de las salidas del modelo BERT. [20]*

En la Figura 3, se muestra cómo los vectores difieren en cada salida. El valor "8" representa las filas o datos y varía según la cantidad de textos que se ingresan al modelo, mientras que el "512" varía dependiendo de la longitud máxima establecida para los tokens. Por otro lado, las dimensiones "768" son una característica fija del modelo BERT, indicando el tamaño del vector de características resultante de la codificación del texto

ingresado. Esto significa que cada palabra o token se representa con un vector de 768 dimensiones, permitiendo capturar una amplia gama de información semántica y contextual. Por último, el valor "12" corresponde a las capas o codificadores presentes en el modelo BERT de Transformers.[20]

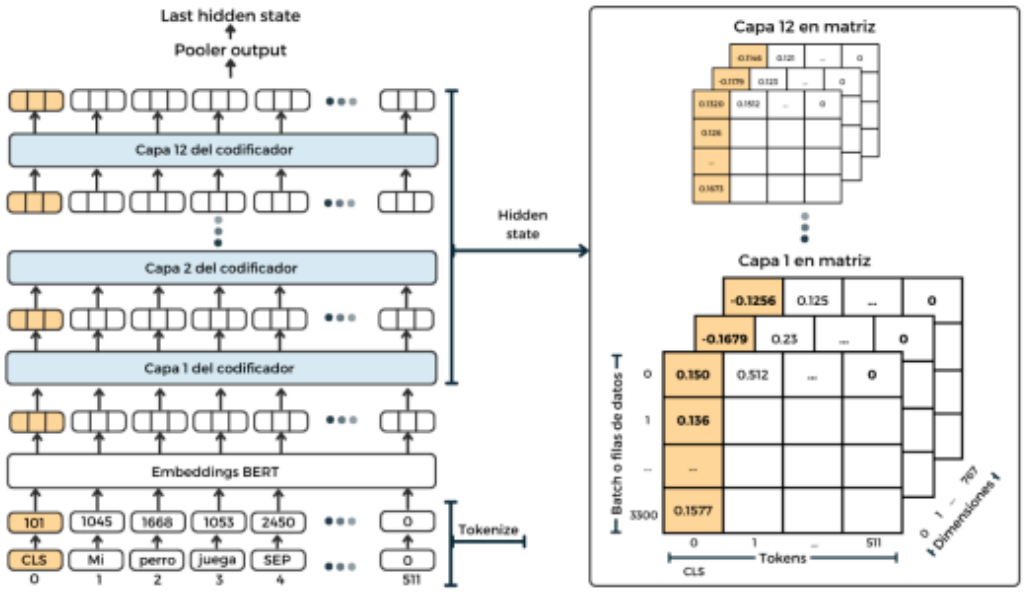


Figura 5. *Estructura interna del modelo BERT. [20]*

De acuerdo con las observaciones de Peters, las 12 capas del modelo pueden dividirse en dos secciones clave. En la primera sección, se describe que las capas inferiores del modelo revelan la estructura sintáctica del texto, mientras que, en la segunda sección, las capas superiores capturan información semántica más compleja. [22]

MODELOS DE CLASIFICADORES BASADOS EN TRANSFORMERS

Modelo BERT

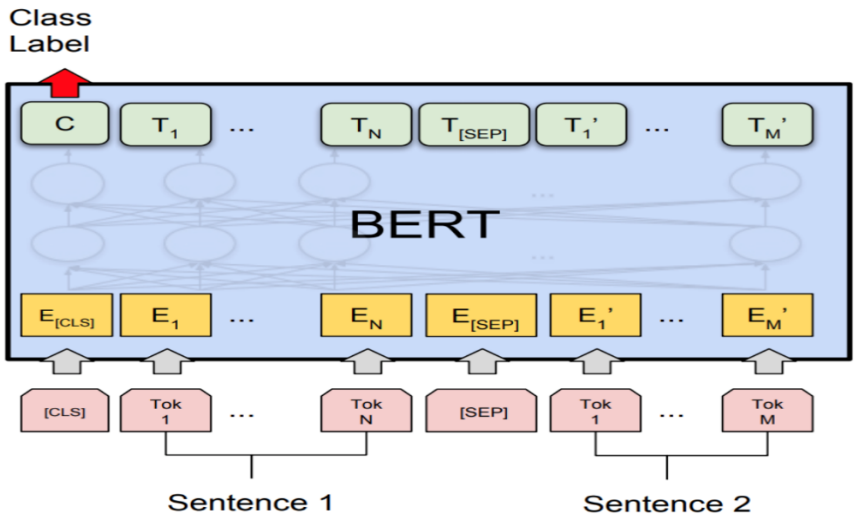


Figura 6. *Modelo BERT. [23]*

El modelo BERT fue propuesto en "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" por Jacob Devlin, Ming-Wei Chang, Kenton Lee y Kristina Toutanova. Es un transformador bidireccional pre-entrenado utilizando una combinación de objetivos de modelado de lenguaje enmascarado y predicción de próxima oración en un corpus grande que comprende el Corpus de Libros de Toronto y Wikipedia.[24]

BERT es conceptualmente simple y empíricamente poderoso. Obtiene nuevos resultados de última generación en once tareas de procesamiento del lenguaje natural, incluido el aumento de la puntuación GLUE al 80,5% (mejora absoluta del 7,7%), precisión de MultiNLI al 86,7% (mejora absoluta del 4,6%), SQuAD v1.1 Prueba de respuesta a preguntas F1 a 93,2 (mejora absoluta de 1,5 puntos) y Prueba SQuAD v2.0 F1 a 83,1 (mejora absoluta de 5,1 puntos).[24]

MODELO BERT		
HIPERPARAMETRO	**VALOR**	**DESCRIPCIÓN**
max_length	80	Longitud máxima de la secuencia de tokens
config.output_hidden_states	False	No devolverá los estados ocultos de cada etapa
optimizer	Adam	Ajusta los pesos del modelo durante el entrenamiento
Learning_rate	1e-05	Determina el tamaño de los pasos que da el optimizador durante la actualización de los pesos
kernel_initializer	TruncatedNormal	Inicializador de los pesos
epochs	5	Duración y calidad del entrenamiento
Función de activación	Linear	Incorpora no linealidad al modelo y mejora la capacidad de aprendizaje
Función de salida	Softmax	Clasifica la salida de los modelos
Units	6	Cantidad de variables de clasificación
Función de perdida	CategoricalCrossentropy	Mide la diferencia entre las predicciones del modelo y las etiquetas verdaderas

Tabla 1.

Tabla 2. *Tabla de Hiperparametros del Modelo BERT. Elaboración propia*

Modelo DistilBERT

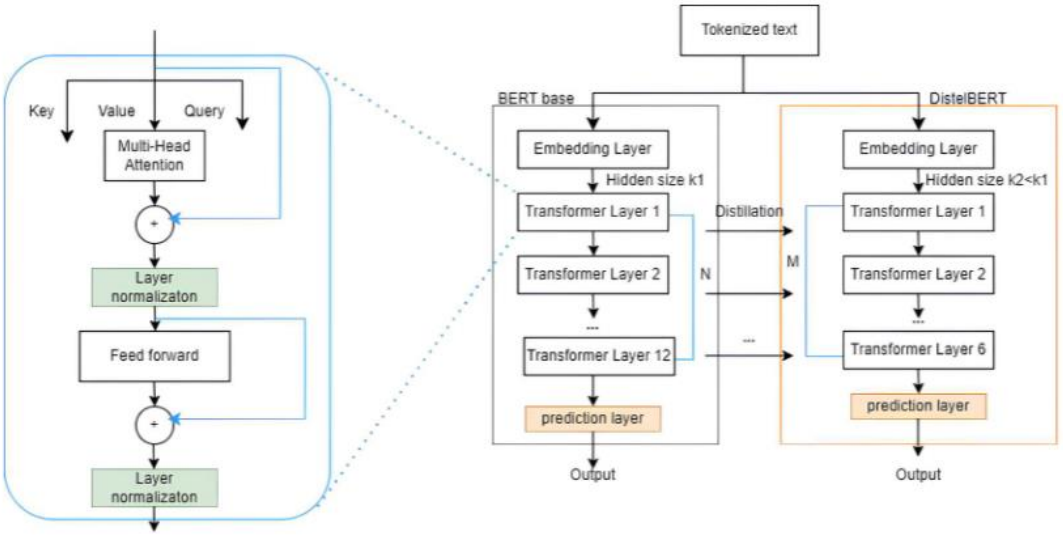


Figura 7. *Modelo DistilBERT. [25]*

DistilBERT es un modelo de transformador pequeño, rápido, económico y liviano entrenado mediante la destilación de la base BERT. Tiene un 40% menos de parámetros que bert-base-uncased, se ejecuta un 60% más rápido y conserva más del 95% del rendimiento de BERT según lo medido en el punto de referencia de comprensión del lenguaje GLUE.[26]

MODELO DistilBERT		
HIPERPARAMETRO	**VALOR**	**DESCRIPCIÓN**
max_length	80	Longitud máxima de la secuencia de tokens
config.output_hidden_states	False	No devolverá los estados ocultos de cada etapa
optimizer	Adam	Ajusta los pesos del modelo durante el entrenamiento
Learning_rate	5e-05	Determina el tamaño de los pasos que da el optimizador durante la actualización de los pesos
kernel_initializer	TruncatedNormal	Inicializador de los pesos
epochs	3	Duración y calidad del entrenamiento
Función de activación	Linear	Incorpora no linealidad al modelo y mejora la capacidad de aprendizaje
Función de salida	Softmax	Clasifica la salida de los modelos
Units	6	Cantidad de variables de clasificación
Función de perdida	CategoricalCrossentropy	Mide la diferencia entre las predicciones del modelo y las etiquetas verdaderas

Tabla 3.

Tabla 4. *Tabla de Hiperparametros del Modelo DistilBERT. Elaboración propia*

Modelo RoBERTa

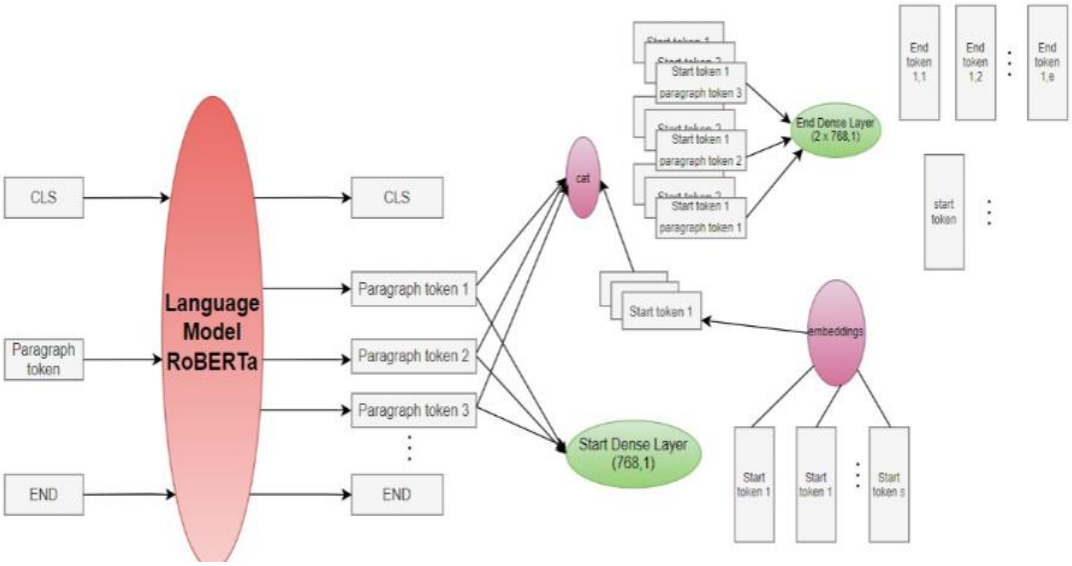


Figura 8. *Modelo RoBERTa. [25]*

El modelo RoBERTa fue propuesto en RoBERTa: A Robustly Optimized BERT Pretraining Approach por Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer y Veselin Stoyanov. Se basa en el modelo BERT de Google lanzado en 2018 y modifica hiperparámetros clave, eliminando el objetivo de pre-entrenamiento de la siguiente oración y entrenando con mini lotes y tasas de aprendizaje mucho más grandes.[27]

MODELO RoBERTa		
HIPERPARAMETRO	**VALOR**	**DESCRIPCIÓN**
max_length	80	Longitud máxima de la secuencia de tokens
config.output_hidden_states	False	No devolverá los estados ocultos de cada etapa
optimizer	Adam	Ajusta los pesos del modelo durante el entrenamiento
Learning_rate	1e-05	Determina el tamaño de los pasos que da el optimizador durante la actualización de los pesos
kernel_initializer	TruncatedNormal	Inicializador de los pesos
epochs	5	Duración y calidad del entrenamiento
Función de activación	Linear	Incorpora no linealidad al modelo y mejora la capacidad de aprendizaje
Función de salida	Softmax	Clasifica la salida de los modelos
Units	6	Cantidad de variables de clasificación
Función de perdida	CategoricalCrossentropy	Mide la diferencia entre las predicciones del modelo y las etiquetas verdaderas

Tabla 5.

Tabla 6. *Tabla de Hiperparametros del Modelo RoBERTa. Elaboración propia*

Modelo AlBERT

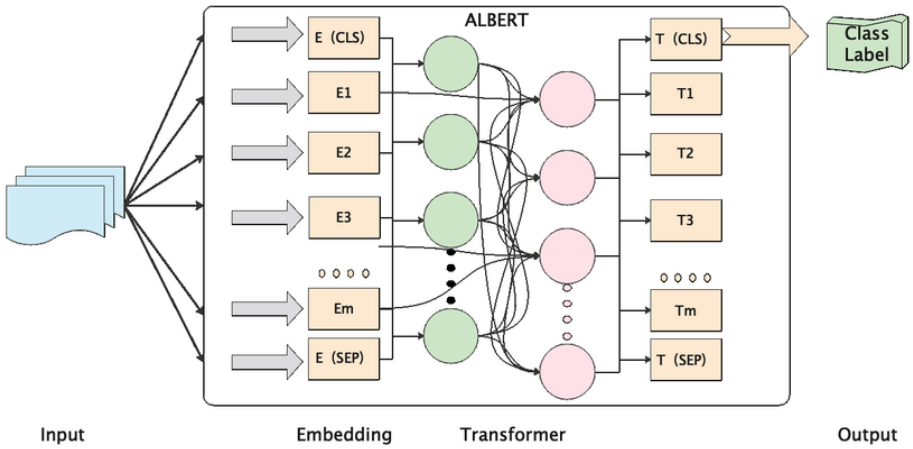


Figura 9. *Modelo AlBERT. [28]*

El modelo ALBERT fue propuesto en ALBERT: A Lite BERT for Self-supervised Learning of Language Representations por Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut. Presenta dos técnicas de reducción de parámetros para reducir el consumo de memoria y aumentar la velocidad de entrenamiento de BERT:

- Dividir la matriz de incrustación en dos matrices más pequeñas.
- Usando capas repetidas divididas entre grupos.[29]

Tabla 7.

MODELO AlBERT		
HIPERPARAMETRO	**VALOR**	**DESCRIPCIÓN**
max_length	80	Longitud máxima de la secuencia de tokens
config.output_hidden_states	False	No devolverá los estados ocultos de cada etapa
optimizer	Adam	Ajusta los pesos del modelo durante el entrenamiento
Learning_rate	5e-05	Determina el tamaño de los pasos que da el optimizador durante la actualización de los pesos
kernel_initializer	TruncatedNormal	Inicializador de los pesos
epochs	5	Duración y calidad del entrenamiento
Función de activación	Linear	Incorpora no linealidad al modelo y mejora la capacidad de aprendizaje
Función de salida	Softmax	Clasifica la salida de los modelos
Units	6	Cantidad de variables de clasificación
Función de perdida	CategoricalCrossentropy	Mide la diferencia entre las predicciones del modelo y las etiquetas verdaderas

Tabla 8.

Tabla 9. *Tabla de Hiperparametros de AlBERT. Elaboración propia*

MODELOS TRANSFORMERS				
Característica	**BERT**	**RoBERTa**	**AlBERT**	**DistilBERT**
Nombre	Bidireccional Encoder Representations from Transformers	Robustly optimized BERT approach	A Lite BERT	Distilled BERT
Publicado	Google AI Languaje	Facebbok AI	Google Research	Hugging Face
Fecha publicación	Octubre 2018	Julio 2019	Septiembre 2019	Octubre 2019
Base de datos	Libros, Wikipedia	Libros, Wikipedia, Common Crawl	Libros, Wikipedia	Libros, Wikipedia
Arquitectura	Transformers bidireccional	Bert, optimización en pre-entrenamiento	Bert, parámetros compartidos y factorización matricial	Bert, más ligero
Parámetros	BERT base: 110 M, BERT large: 340 M	RoBERTa base: 125 M, RoBERTa large: 335 M	AlBERT base: 12 M, AlBERT large: 18 M	DistilBERT: 66M
Pre-entrenamiento	Mascara de lenguaje y predicciones de la siguiente oración	Mascara de lenguaje	Mascara de lenguaje, predicción de la siguiente oración	Mascara de lenguaje
Velocidad entrenamiento	Lenta	Velocidad optimizada	Mas rápida y eficiente	Mas rápida y eficiente
Memoria requerida	Alta	Alta	Menor	Menor
Rendimiento	Solido en multitareas de NLP	Mejor rendimiento debido a más datos y más pasos de entrenamiento	Eficiente con parámetros reducidos y rendimiento competitivo	Rendimiento competitivo con menos recursos
Uso Principal	Tareas generales de NLP	Tareas generales de NLP	NLP, con limitaciones de recursos	NLP, con limitaciones de recursos

Tabla 10. *Tabla comparativa de Modelos. Elaboración propia*

EVALUACIÓN DEL MODELO MÉTRICAS

Recall (Recuperación): Es la proporción de ejemplos relevantes que fueron correctamente identificados por el modelo entre todos los ejemplos relevantes en los datos. Se calcula como la relación entre verdaderos positivos (TP) y la suma de verdaderos positivos y falsos negativos (FN). Un alto recall indica que el modelo identifica la mayoría de los casos relevantes en los datos.[6]

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

Precisión (Precisión): Representa la proporción de ejemplos clasificados como positivos que son verdaderamente positivos entre todos los ejemplos clasificados como positivos por el modelo. Se calcula como la relación entre verdaderos positivos (TP) y la suma de verdaderos positivos y falsos positivos (FP). Una alta precisión indica que el modelo no clasifica erróneamente muchas muestras como positivas.[6]

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

Accuracy (Exactitud): Es la medida global de la precisión del modelo y representa la proporción de predicciones correctas sobre el total de predicciones. Se calcula como la relación entre verdaderos positivos y verdaderos negativos (TN) y la suma de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos.[6]

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

F1 Score: Es una métrica que combina precisión y recall en un solo valor. Es la media armónica de precisión y recall, y proporciona un equilibrio entre ambas métricas. El F1 Score es especialmente útil cuando hay un desequilibrio entre las clases, ya que tiene en cuenta tanto falsos positivos como falsos negativos.[6]

$$\boldsymbol{F_1 Score = \frac{2\ x\ precision\ x\ recall}{recall + precision}}$$

Matriz de confusión

Es una herramienta que permite visualizar el desempeño de un modelo de clasificación al comparar las predicciones del modelo con los valores reales de las clases en un conjunto de datos. Esta matriz es especialmente útil para entender el rendimiento del modelo en

términos de verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN).[6]

- **Verdaderos Positivos (TP):** Las muestras que el modelo predijo correctamente como positivas.
- **Falsos Positivos (FP):** Las muestras que el modelo predijo incorrectamente como positivas cuando en realidad eran negativas.
- **Verdaderos Negativos (TN):** Las muestras que el modelo predijo correctamente como negativas.
- **Falsos Negativos (FN):** Las muestras que el modelo predijo incorrectamente como negativas cuando en realidad eran positivas.

Prediction		
Label	TP	FN
	FP	TN

Tabla 11.

Tabla 12. *Matriz de confusión. Elaboración propia*

TRANSFORMERS				
MODELO	**PRECISION**	**RECALL**	**ACCURACY**	**F1-SCORE**
BERT	0.65	0.35	0.35	0.44
RoBERTa	0.79	0.54	0.54	0.63
DistilBERT	0.78	0.75	0.75	0.76
AlBERT	0.30	0.33	0.33	0.50

Tabla 13. *Resultados. Elaboración propia*

En la tabla 7 se presentan los resultados de las métricas obtenidas durante el entrenamiento de diversos modelos. De estos resultados, se puede destacar al modelo RoBERTa situada ligeramente por encima del promedio de los porcentajes obtenidos. Por otro lado, los modelos BERT y ALBERT mostraron un rendimiento significativamente inferior al esperado en las métricas evaluadas.

En contraste, el modelo DistilBERT demostró ser el más adaptable y estable, alcanzando un promedio superior al 75% en las métricas. Cabe mencionar que el entrenamiento de todos los modelos se realizó con una duración 3-5 epoch, las variaciones en los resultados entre los modelos se deben a las diferencias en sus arquitecturas y parámetros característicos. A continuación, se presentará una tabla de los resultados del entrenamiento de los modelos según la clasificación de las emociones.

METRICAS DE MODELO BERT				
Emociones	**precision**	**recall**	**f1-score**	**support**
TRISTEZA	0.34	0.35	0.35	1212
ENOJO	0.00	0.00	0.00	0
AMAR	0.00	0.00	0.00	0
SURPRESA	0.00	0.00	0.00	0
MIEDO	0.00	0.00	0.00	0
FELIZ	0.76	0.34	0.47	3080
Weighted avg	0.65	0.35	0.44	
acurracy	0.35		**Total**	4292

Tabla 14. *Métricas por clase del Modelo BERT. Elaboración propia*

Tabla 15.

METRICAS DE MODELO RoBERTa				
Emociones	**precision**	**recall**	**f1-score**	**support**
TRISTEZA	0.80	0.47	0.60	2101
ENOJO	0.15	0.62	0.24	149
AMAR	0.00	0.00	0.00	0
SURPRESA	0.00	0.00	0.00	0
MIEDO	0.07	0.75	0.13	52
FELIZ	0.86	0.59	0.70	1990
Weighted avg	0.79	0.54	0.63	
acurracy	0.54		**Total**	4292

Tabla 16. *Métricas por clase del Modelo RoBERTa. Elaboración propia*

Tabla 17.

METRICAS DE MODELO DistilBERT				
Emociones	**precision**	**recall**	**f1-score**	**support**
TRISTEZA	0.85	0.73	0.79	1423
ENOJO	0.75	0.74	0.75	578
AMAR	0.30	0.69	0.42	147
SURPRESA	0.57	0.67	0.62	156
MIEDO	0.62	0.78	0.69	414
FELIZ	0.85	0.78	0.81	1574
Weighted avg	0.78	0.75	0.76	
acurracy	0.75		**Total**	4292

Tabla 18. *Métricas por clase del Modelo DistilBERT. Elaboración propia*

METRICAS DE MODELO AlBERT				
Emociones	**precision**	**recall**	**f1-score**	**support**
TRISTEZA	0.00	0.00	0.00	0
ENOJO	0.00	0.00	0.00	0
AMAR	0.00	0.00	0.00	0
SURPRESA	0.00	0.00	0.00	0
MIEDO	0.00	0.00	0.00	0
FELIZ	1.00	0.33	0.50	4292
Weighted avg	1.00	0.33	0.50	
acurracy	0.33		**Total**	4292

Tabla 19. *Métricas por clase del Modelo AlBERT. Elaboración propia*

CONCLUSIONES

- Describir el funcionamiento de las técnicas de aprendizaje automático ha permitido comprender en detalle los principios y métodos que subyacen a esta tarea fundamental en el procesamiento del lenguaje natural. Se ha explorado cómo los modelos avanzados Transformers, implementan mecanismos de atención para capturar dependencias contextuales a lo largo de las secuencias de texto.

- La aplicación de modelos Transformers pre-entrenados en este proyecto ha demostrado que, aunque se basan en la misma arquitectura de red neuronal orientada al procesamiento del lenguaje natural, presentan variaciones

significativas en sus características y en los parámetros necesarios para su óptimo desempeño.

- Una característica común observada entre los modelos es la cantidad significativa de datos requerida para su funcionamiento óptimo. Estos modelos generalmente necesitan un mínimo de 10,000 ejemplos etiquetados para generar resultados razonables.

- Como resultado de las pruebas y el entrenamiento, se puede afirmar que los modelos BERT, RoBERTa, DistilBERT y ALBERT utilizados en este proyecto han mostrado un rendimiento destacado. Esto se evidencia en las métricas de rendimiento obtenidas para cada modelo, las cuales han sido comparadas y visualizadas mediante tablas.

- Para llevar a cabo el proyecto, se utilizó una cuenta estándar de Google Colab, lo que representó una limitación en cuanto a los recursos necesarios para el entrenamiento de los modelos. Una de estas restricciones fue el tiempo máximo de uso del entorno de trabajo, lo que impidió extender el tiempo de duración de las epoch y afectando así también los resultados.

- En conclusión, es posible clasificar textos en idioma español de manera efectiva usando modelos pre-entrenados Transformers obteniendo grandes resultados dentro de la evaluación de rendimiento, pero destacando aun así a un modelo en específico DistilBERT quien obtuvo un porcentaje mayor dentro de las métricas evaluadas a comparación de sus otros competidores otorgándole, así como el modelo mejor valorado y adaptado en la detección de características en español.

RECOMENDACIONES

- En este proyecto se consideraron únicamente cuatro modelos Transformers. Sin embargo, existe una amplia variedad que podrían potencialmente ofrecer mejores resultados en la detección de emociones. Modelos como GPT-3, T5, y variantes específicas de BERT entrenadas en grandes conjuntos de datos multilingües, podrían proporcionar una mayor precisión y eficacia. En futuras investigaciones deberían explorar y evaluar una gama más amplia de modelos Transformers para identificar aquellos que son más adecuados y eficientes en la tarea específica de detección de emociones en textos en español.

- También es importante tener en cuenta que estos modelos están predominantemente orientados al idioma inglés. Por lo tanto, para realizar un estudio más preciso, se recomienda dedicar tiempo a evaluar qué modelos podrían responder de manera más efectiva al uso de otros idiomas. Esto implica considerar modelos específicamente adaptados o entrenados para el español u otras lenguas, asegurando así una mejor adaptación y rendimiento en contextos multilingües.

- En el proyecto, se utilizó un conjunto de datos relativamente pequeño para el entrenamiento de los modelos. Esto puede haber influido en los resultados obtenidos. Se recomienda, para cumplir con los requerimientos de los modelos en función del tamaño de los conjuntos de datos, utilizar técnicas de aumentación de datos o extraer información mediante web scraping.

- Para investigaciones futuras, se recomienda ampliar el estudio considerando también los emoticonos como texto gráfico. Debido a que los usuarios no solo comunican sus emociones mediante palabras, sino que también utilizan estos símbolos visuales para expresar sentimientos y estados de ánimo. Incluirlos en un estudio podría proporcionar una comprensión más completa y precisa de las emociones transmitidas en las comunicaciones digitales.

- Las restricciones de tiempo en el entorno de trabajo afectaron los resultados obtenidos en las métricas. Por ello, se recomienda utilizar una cuenta profesional de Google Colab.

REFERENCIAS

[1] Erika González García and Nazaret Martínez Heredia, "Redes sociales como factor incidente en el área social, personal y académica de alumnos de Educación Secundaria Obligatoria," Fundación Dialnet. Accessed: Oct. 25, 2023. [Online]. Available: https://dialnet.unirioja.es/servlet/articulo?codigo=6531604

[2] J. Serrano-Puche, "Internet and emotions: New trends in an emerging field of research," *Comunicar*, vol. 24, no. 46, pp. 19–26, Jan. 2016, doi: 10.3916/C46-2016-02.

[3] "'Plan Nacional de Desarrollo 2017-2021 Toda una Vida' de Ecuador," Observatorio Regional de Planificación para el Desarrollo de América Latina y el Caribe. Accessed: Oct. 26, 2023. [Online]. Available: https://observatorioplanificacion.cepal.org/sites/default/files/plan/files/EcuandorPlanNacionalTodaUnaVida20172021.pdf

[4] Elasticsearch B.V., "¿Qué es el análisis de sentimiento?," Elastic, The Search Analytics Company. Accessed: Mar. 24, 2024. [Online]. Available: https://www.elastic.co/es/what-is/sentiment-analysis

[5] César Espín Riofrio, María Pérez Pilco, Joshua Ávila Vacas, and Tania Peralta Guaraca, "Identificación de comentarios sexistas en textos en español utilizando un modelo pre-entrenado Transformer para Procesamiento de Lenguaje Natural," 2023.

[6] Lazo Vasquez and Ricardo Manuel, "Clasificación de la personalidad utilizando procesamiento de lenguaje natural y aprendizaje profundo para detectar patrones de notas de suicidio en redes sociales," Titulo Profesional, Universidad Católica San Pablo, Arequipa, 2022. Accessed: Oct. 25, 2023. [Online]. Available: https://renati.sunedu.gob.pe/handle/sunedu/3359968

[7] S. De Antón Santiago, “Aplicación de reconocimiento de emociones para personas vulnerables ante la soledad,” Trabajo de fin de grado, Universidad de Alicante, Alicate, 2023. Accessed: Oct. 25, 2023. [Online]. Available: https://rua.ua.es/dspace/handle/10045/135241

[8] Ana Estévez, Milena López Montón, Janire Momeñe, Iker Pastor López, Asier González Santocildes, and Borja Sanz Urquijo, *Comentarios negativos en redes sociales*, Míriam López. Madrid: Fundación MAPRE, 2022.

[9] Delroy L Paulhus and Kevin M Williams, “The Dark Triad of personality: Narcissism, Machiavellianism, and psychopathy,” *J Res Pers*, vol. 36, pp. 556–563, Dec. 2002, Accessed: Oct. 26, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0092656602005056

[10] Pino Diez Raúl, Gómez Gómez Alberto, and de Abajo Martínez Nicolás, *Introducción a la Inteligencia Artificial: Sistemas Expertos, Reade Neuronales Artificiales y Computación Evolutiva*. Servicio de Publicaciones Universidad de Oviedo, 2001.

[11] F. J. Chara Pelaez and V. Martel Anaya, “Sistema de atención de clientes casino y definición de campañas segmentadas mediante machine learning,” Universidad Peruana de Ciencias Aplicadas (UPC), Lima, 2020.

[12] Rojas Esperanza Manrique, “Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo,” *Revista Ibérica de Sistemas e Tecnologías de Información*, pp. 586–599, Apr. 2020.

[13] “Deep Learning,” in *Introduction to Environmental Data Science*, Cambridge University Press, 2023, pp. 494–517. doi: 10.1017/9781107588493.016.

[14] R. N, “Machine Learning for Natural Language Processing: Techniques and Applications,” in *Cutting-Edge Technologies in Innovations in Computer Science and Engineering*, San International Scientific Publications, 2023. doi: 10.59646/csebookc6/004.

[15] “Guide To Pysentimiento Toolkit | Text Classification Using Transformers,” analytics india mag. Accessed: Jun. 15, 2024. [Online]. Available:

https://analyticsindiamag.com/guide-to-pysentimiento-toolkit-text-classification-using-transformers/

[16] M. Kuroki, “Integrating data science into an econometrics course with a Kaggle competition,” *J Econ Educ*, vol. 54, no. 4, pp. 364–378, Oct. 2023, doi: 10.1080/00220485.2023.2220695.

[17] Joaquín Amat Rodrigo, “Análisis de texto (text mining) con Python,” Ciencia de Datos. Accessed: Mar. 24, 2024. [Online]. Available: https://cienciadedatos.net/documentos/py25-text-mining-python

[18] A. Vaswani *et al.*, “Attention Is All You Need,” Jun. 2017.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Oct. 2018.

[20] H. A. Camacho Villalva and L. A. Ramos Ramírez, “Extracción de capas de atención de un modelo basado en BERT para detectar texto escrito por humano o generado automáticamente,” Universidad de Guayaquil, Guayaquil, 2023. Accessed: May 11, 2024. [Online]. Available: http://repositorio.ug.edu.ec/handle/redug/71639

[21] “Model outputs,” Hugging Face. Accessed: May 11, 2024. [Online]. Available: https://huggingface.co/docs/transformers/main_classes/output

[22] M. Peters, M. Neumann, L. Zettlemoyer, and W. Yih, “Dissecting Contextual Word Embeddings: Architecture and Representation,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2018, pp. 1499–1509. doi: 10.18653/v1/D18-1179.

[23] “BERT (Language Model),” *Devopedia*, vol. 5, Jun. 2021, Accessed: May 12, 2024. [Online]. Available: https://devopedia.org/bert-language-model

[24] “BERT,” Hugging Face. Accessed: May 12, 2024. [Online]. Available: https://huggingface.co/transformers/v4.5.1/model_doc/bert.html

[25] C. ÖZKURT, “Comparative Analysis of State-of-the-Art Q\&A Models: BERT, RoBERTa, DistilBERT, and ALBERT on SQuAD v2 Dataset,” *ResearchGate*,

Feb. 2024, Accessed: May 12, 2024. [Online]. Available: https://www.researchgate.net/publication/378327924_Comparative_Analysis_of_State-of-the-Art_QA_Models_BERT_RoBERTa_DistilBERT_and_ALBERT_on_SQuAD_v2_Dataset/references

[26] "DistilBERT," Hugging Face. Accessed: May 12, 2024. [Online]. Available: https://huggingface.co/transformers/v4.5.1/model_doc/distilbert.html

[27] "RoBERTa," Hugging Face. Accessed: May 12, 2024. [Online]. Available: https://huggingface.co/transformers/v4.5.1/model_doc/roberta.html

[28] J. Li, B. Wang, and H. Ding, "Lijunyi at SemEval-2020 Task 4: An ALBERT Model Based Maximum Ensemble with Different Training Sizes and Depths for Commonsense Validation and Explanation," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, Stroudsburg, PA, USA: International Committee for Computational Linguistics, 2020, pp. 556–561. doi: 10.18653/v1/2020.semeval-1.69.

[29] "AlBERT," Hugging Face. Accessed: May 12, 2024. [Online]. Available: https://huggingface.co/transformers/v4.5.1/model_doc/albert.html

ANEXOS

Dataset

Column1	Column2
Emocion	Texto
tristeza	no me sentí humillado
tristeza	Puedo pasar de sentirme tan desesperado a tener tant...
enojo	Me estoy tomando un minuto para publicar. Me siento...
amar	Siempre siento nostalgia por la chimenea. Sabré que to...
enojo	me siento de mal humor
tristeza	Me he sentido un poco agobiado últimamente, no esta...
sorpresa	He estado tomando miligramos o la cantidad recomend...
miedo	Me siento tan confundido acerca de la vida como un ad...
feliz	He estado en Petronas durante años. Siento que Petron...
amar	yo también me siento romántico
tristeza	Siento que tengo que hacer que el sufrimiento que veo...
feliz	Siento que correr es una experiencia divina y que pued...

Preparación del entorno de trabajo

Librerías

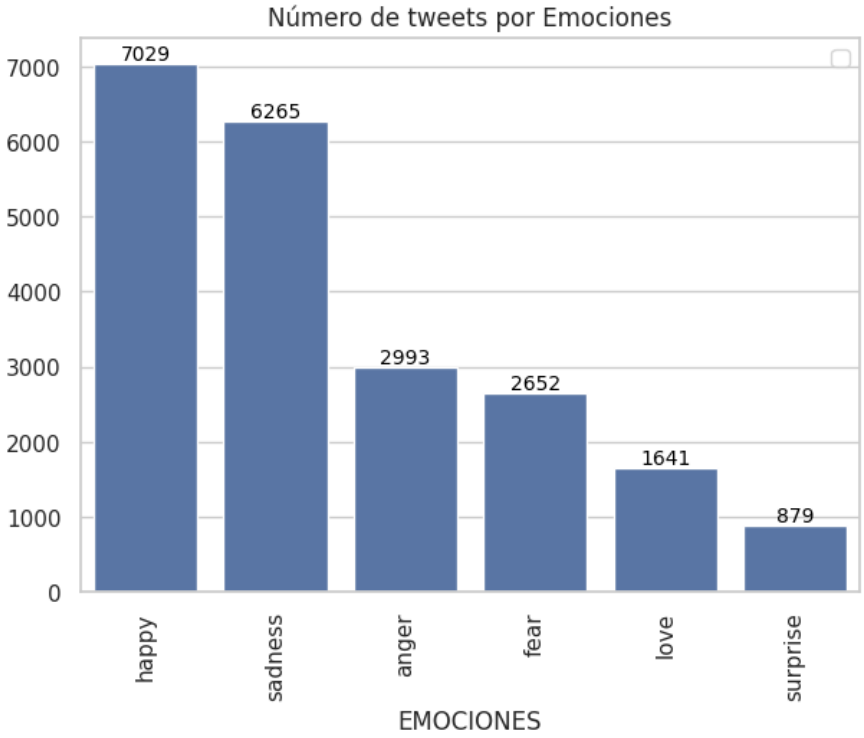
```
import pandas as pd
import numpy as np
import os
import matplotlib
import matplotlib.pyplot as plt
plt.style.use('ggplot')
import seaborn as sns
sns.set(style='whitegrid')
import keras

from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA, TruncatedSVD
from sklearn.metrics import classification_report,confusion_matrix

from collections import defaultdict
from collections import Counter

import re
import gensim
import string

from tqdm import tqdm
from keras.preprocessing.text import Tokenizer
from keras.models import Sequential
from keras.layers import Embedding, LSTM,Dense, SpatialDropout1D, Dropout
from keras.initializers import Constant

import tensorflow as tf
import warnings
warnings.simplefilter('ignore')
```

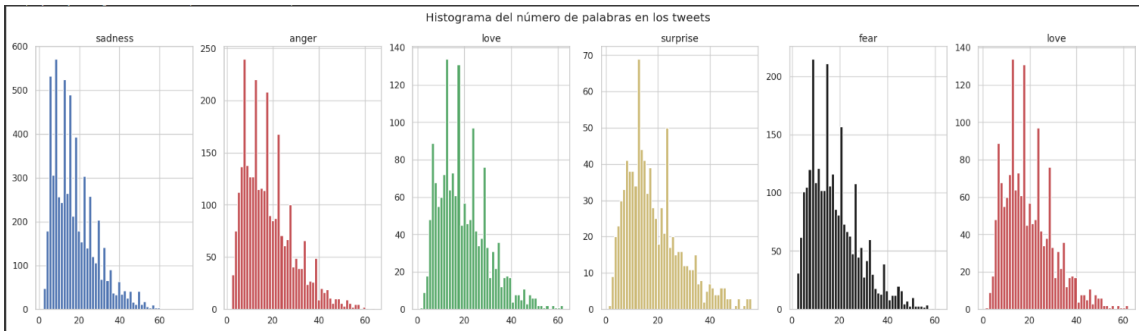
Descarga y preparación del Dataset

```
# Cargar el conjunto de datos y crear una nueva columna
dataset_read = pd.read_csv("/content/drive/MyDrive/csv_multiemotion.csv",sep=';', encoding='utf-8')[["Emocion","Texto"]]
dataset_read.columns = ["EMOCIONES", "COMENTARIOS"]
dataset_read.head(-5)
```

	EMOCIONES	COMENTARIOS
0	tristeza	no me sentí humillado
1	tristeza	Puedo pasar de sentirme tan desesperado a tene...
2	enojo	Me estoy tomando un minuto para publicar. Me s...
3	amar	Siempre siento nostalgia por la chimenea. Sabr...
4	enojo	me siento de mal humor
...	...	...
21449	miedo	Edouard lo miró y sintió un desánimo repugnante.
21450	miedo	Cuando Tina y los niños se mudaron a la escuel...
21451	miedo	Pero sintió "consternación y pena por tantas p...
21452	miedo	Se miraron unos a otros consternados, la peque...
21453	miedo	Cleo miró fijamente el contenido con tristeza.

21454 rows × 2 columns


Número de tweets por Emociones
7029
6265
2993
2652
1641
879
7000
6000
5000
4000
3000
2000
1000
0
happy
sadness
anger
fear
love
surprise
EMOCIONES


Histograma del número de palabras en los tweets
sadness
anger
love
surprise
fear
love

Preprocesamiento

```
[ ] dataset_read['COMENTARIOS_TOKEN'] = dataset_read['COMENTARIOS'].apply(lambda x: strip_links(x))

[ ] dataset_read['COMENTARIOS_TOKEN'] = dataset_read['COMENTARIOS'].apply(lambda x: strip_all_entities(x))

[ ] dataset_read['COMENTARIOS_TOKEN'] = dataset_read['COMENTARIOS'].str.replace(r'HASHTAG', '', regex=True)
    dataset_read['COMENTARIOS_TOKEN'] = dataset_read['COMENTARIOS'].str.replace(r'USER', '', regex=True)

[ ] def convert_to_lower(text):
        return text.lower()

[ ] dataset_read['COMENTARIOS_TOKEN'] = dataset_read['COMENTARIOS'].apply(lambda x: convert_to_lower(x))

    def remove_numbers(text):
        number_pattern = r'\d+'
        without_number = re.sub(pattern=number_pattern, repl=" ", string=text)
        return without_number

[ ] dataset_read['COMENTARIOS_TOKEN'] = dataset_read['COMENTARIOS'].apply(lambda x: remove_numbers(x))

[ ] def remove_punctuation(text):
        return text.translate(str.maketrans('', '', string.punctuation))

[ ] dataset_read['COMENTARIOS_TOKEN'] = dataset_read['COMENTARIOS'].apply(lambda x: remove_punctuation(x))

[ ] from nltk import word_tokenize #Método que realizará la tokenización de texto

[ ] # Descargar el conjunto de datos stop words
    ![ ! -f stopwordNarcisism.txt ] && wget https://raw.githubusercontent.com/SmartFastSolution/ModeloTextClassification/master/stopwordNarcisism.txt

[ ] # Pasar la lista de stop_words al array de vectores.
    stopword_es=pd.read_table('stopwordNarcisism.txt',header=None)
    stopword_es = np.asarray(stopword_es).ravel()
    print(stopword_es[:40])

    ['?' 'a' 'aahh' 'ah' 'al' 'algo' 'algun' 'algún' 'alguna' 'algunos' 'alla'
     'alta' 'alto' 'añado' 'ante' 'apenas' 'aquel' 'aquello' 'aquellos' 'as'
     'asi' 'así' 'atras' 'aun' 'aún' 'aunque' 'cada' 'cayere' 'chuzo' 'cierta'
     'cierto' 'como' 'con' 'cual' 'cualquier' 'cuan' 'cuando' 'cuenta' 'da'
     'de']
```

```
[ ] import emoji

    def remove_emoji(string):
        emoji_pattern = re.compile("["
                           u"\U0001F600-\U0001F64F" #emoticones
                           u"\U0001F300-\U0001F5FF" #símbolos y pictogramas
                           u"\U0001F680-\U0001F6FF" #Símbolos de transporte y mapas
                           u"\U0001F1E0-\U0001F1FF" #banderas
                           u"\U00002702-\U000027B0"
                           u"\U000024C2-\U0001F251"
                                   "]+", flags=re.UNICODE)
        return emoji_pattern.sub(r'', string)

[ ] dataset_read['COMENTARIOS_TOKEN'] = dataset_read['COMENTARIOS'].apply(lambda x: remove_emoji(x))

[ ] from cleantext import clean
    dataset_read['COMENTARIOS_TOKEN'] = dataset_read['COMENTARIOS'].apply(lambda x: clean(x, no_emoji = True))

    WARNING:root:Since the GPL-licensed package `unidecode` is not installed, using Python's `unicodedata` package which yields worse results.

[ ] dataset_read.head(-5)
```

	EMOCIONES	COMENTARIOS	TRAITS	COMENTARIOS_TOKEN
0	sadness	no me sentí humillado	0.0	no me senti humillado
1	sadness	Puedo pasar de sentirme tan desesperado a tene...	0.0	puedo pasar de sentirme tan desesperado a tene...
2	anger	Me estoy tomando un minuto para publicar. Me s...	1.0	me estoy tomando un minuto para publicar. me s...
3	love	Siempre siento nostalgia por la chimenea. Sabr...	2.0	siempre siento nostalgia por la chimenea. sabr...
4	anger	me siento de mal humor	1.0	me siento de mal humor
...	...	...	...	...
21449	fear	Edouard lo miró y sintió un desánimo repugnante.	4.0	edouard lo miro y sintio un desanimo repugnante.
21450	fear	Cuando Tina y los niños se mudaron a la escuel...	4.0	cuando tina y los ninos se mudaron a la escuel...
21451	fear	Pero sintió "consternación y pena por tantas p...	4.0	pero sintio "consternacion y pena por tantas p...
21452	fear	Se miraron unos a otros consternados, la peque...	4.0	se miraron unos a otros consternados, la peque...
21453	fear	Cleo miró fijamente el contenido con tristeza.	4.0	cleo miro fijamente el contenido con tristeza.

21454 rows × 4 columns

Preparación delo modelo para el entrenamiento

```
data.head(-3000)
```

	COMENTARIOS	TRAITS	LABEL_EMOTION	TWEETS
0	no me sentí humillado	0.0	0.0	0
1	Puedo pasar de sentirme tan desesperado a tene...	0.0	0.0	0
2	Me estoy tomando un minuto para publicar. Me s...	1.0	1.0	1
3	Siempre siento nostalgia por la chimenea. Sabr...	2.0	2.0	2
4	me siento de mal humor	1.0	1.0	1
...	...	...	...	...
18454	Lo siento por Peter, estaba convencido de su i...	5.0	5.0	5
18455	Estaba bien, incluso hice algunas costuras cer...	5.0	5.0	5
18456	Me siento bastante festivo aquí en el sur de F...	5.0	5.0	5
18457	Todavía me siento un poco sorprendido por la n...	3.0	3.0	3
18458	me siento muy orgulloso	5.0	5.0	5

18459 rows × 4 columns

"Dividir en conjuntos de entrenamiento y validación, dado que el archivo contiene más de 150 mil instancias, podemos considerar solo una pequeña parte de ellas como validación y aún así, el número es relativamente grande. Por eso, estableceremos el tamaño de prueba en un 20%, como sigue:"

```
[ ] data_train, data_val = train_test_split(data, test_size = 0.2)
```

```
print(data_train.shape)
print(data_val.shape)
```

```
(17167, 4)
(4292, 4)
```

Modelo BERT

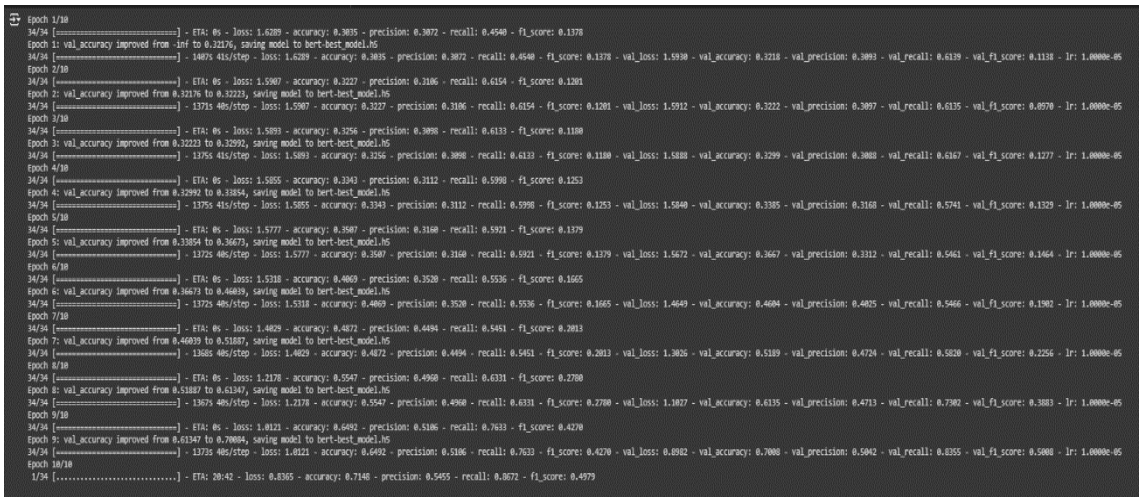
```
from transformers import TFBertModel,  BertConfig, BertTokenizerFast
```

+ Código + Texto

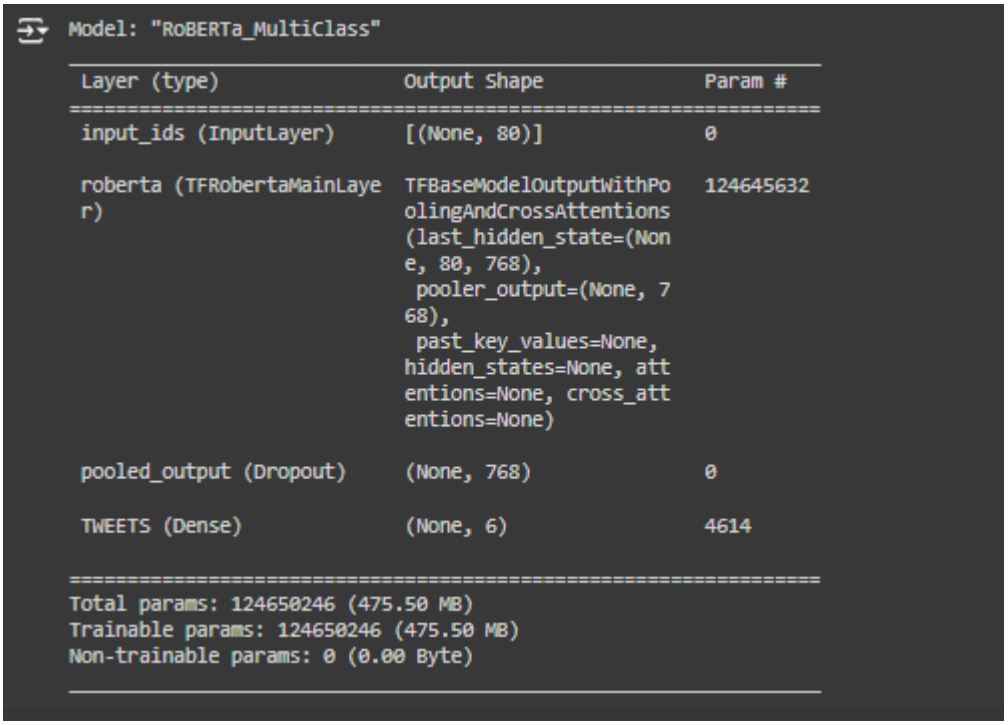
```
[ ] # Nombre del modelo BERT
model_name = 'bert-base-uncased'

# Longuitud maxima de los tokens
max_length = 80

# Cargar la configuración de transformers y establecer output_hidden_states en False
config = BertConfig.from_pretrained(model_name)
config.output_hidden_states = False

# Cargar el tokenizador de BERT
tokenizer = BertTokenizerFast.from_pretrained(pretrained_model_name_or_path = model_name, config = config)

# Cargar el modelo BERT de Transformers
transformer_bert_model = TFBertModel.from_pretrained(model_name, config = config)
```

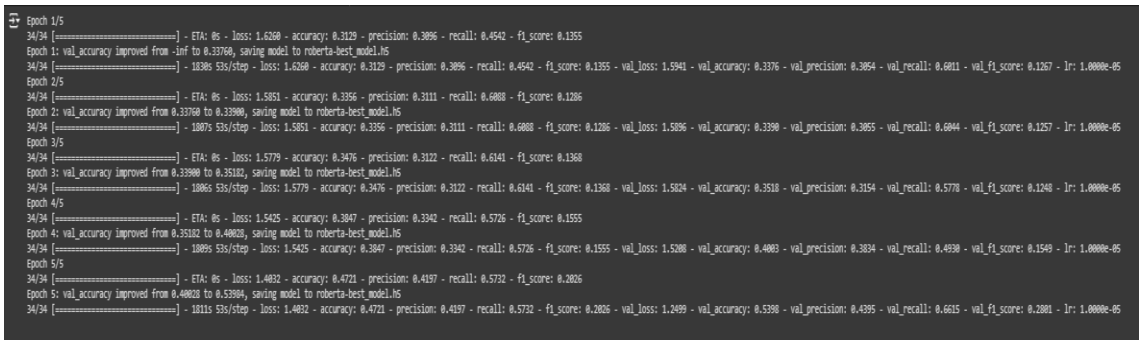
```
config.json: 100% 570/570 [00:00<00:00, 54.8kB/s]
tokenizer_config.json: 100% 48.0/48.0 [00:00<00:00, 5.19kB/s]
vocab.txt: 100% 232k/232k [00:00<00:00, 4.22MB/s]
tokenizer.json: 100% 466k/466k [00:00<00:00, 14.9MB/s]
model.safetensors: 100% 440M/440M [00:01<00:00, 290MB/s]
Some weights of the PyTorch model were not used when initializing the TF 2.0 model TFBertModel: ['cls.predictions.transform.dense.weight', 'cls.predictions.transform.Laye
- This IS expected if you are initializing TFBertModel from a PyTorch model trained on another task or with another architecture (e.g. initializing a TFBertForSequenceCla
- This IS NOT expected if you are initializing TFBertModel from a PyTorch model that you expect to be exactly identical (e.g. initializing a TFBertForSequenceClassificati
All the weights of TFBertModel were initialized from the PyTorch model.
If your task is similar to the task the model of the checkpoint was trained on, you can already use TFBertModel for predictions without further training.
```

```
Model: "BERT_MultiClass"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_ids (InputLayer)      [(None, 80)]              0

 bert (TFBertMainLayer)      TFBaseModelOutputWithPo   109482240
                             olingAndCrossAttentions
                             (last_hidden_state=(Non
                             e, 80, 768),
                              pooler_output=(None, 7
                             68),
                              past_key_values=None,
                             hidden_states=None, att
                             entions=None, cross_att
                             entions=None)

 pooled_output (Dropout)     (None, 768)               0

 TWEETS (Dense)              (None, 6)                 4614

=================================================================
Total params: 109486854 (417.66 MB)
Trainable params: 109486854 (417.66 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```



Modelo RoBERTa





Modelo DistilBERT

```
### --------- Modelo DistilBERT ---------- ###

model_name = 'distilbert-base-uncased'
max_length = 80

config = DistilBertConfig.from_pretrained(model_name)
config.output_hidden_states = False

tokenizer = DistilBertTokenizer.from_pretrained(pretrained_model_name_or_path = model_name, config = config)
transformer_distilbert_model = TFDistilBertModel.from_pretrained(model_name, config = config)
```

```
Model: "DistilBERT_MultiClass"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_ids (InputLayer)      [(None, 80)]              0

 distilbert (TFDistilBertMa  TFBaseModelOutput(last_   66362880
 inLayer)                    hidden_state=(None, 80,
                              768),
                              hidden_states=None, at
                             tentions=None)

 tf.__operators__.getitem (  (None, 768)               0
 SlicingOpLambda)

 pooled_output (Dropout)     (None, 768)               0

 TWEETS (Dense)              (None, 6)                 4614

=================================================================
Total params: 66367494 (253.17 MB)
Trainable params: 66367494 (253.17 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
Epoch 1/3
34/34 [==============================] - ETA: 0s - loss: 1.6002 - accuracy: 0.3225 - precision: 0.3086 - recall: 0.5703 - f1_score: 0.1305
Epoch 1: val_accuracy improved from -inf to 0.34063, saving model to distilBert-best_model.h5
34/34 [==============================] - 701s 20s/step - loss: 1.6002 - accuracy: 0.3225 - precision: 0.3086 - recall: 0.5703 - f1_score: 0.1305 - val_loss: 1.5794 - val_accuracy: 0.3406 - val_precision: 0.3160 - val_recall: 0.5944 - val_f1_score: 0.1100 - lr: 5.0000e-05
Epoch 2/3
34/34 [==============================] - ETA: 0s - loss: 1.5399 - accuracy: 0.3086 - precision: 0.3403 - recall: 0.5392 - f1_score: 0.1540
Epoch 2: val_accuracy improved from 0.34063 to 0.43150, saving model to distilBert-best_model.h5
34/34 [==============================] - 685s 20s/step - loss: 1.5399 - accuracy: 0.3086 - precision: 0.3403 - recall: 0.5392 - f1_score: 0.1540 - val_loss: 1.4682 - val_accuracy: 0.4315 - val_precision: 0.4397 - val_recall: 0.4257 - val_f1_score: 0.1868 - lr: 5.0000e-05
Epoch 3/3
34/34 [==============================] - ETA: 0s - loss: 1.0876 - accuracy: 0.6801 - precision: 0.5373 - recall: 0.6669 - f1_score: 0.4482
Epoch 3: val_accuracy improved from 0.43150 to 0.75349, saving model to distilBert-best_model.h5
34/34 [==============================] - 686s 20s/step - loss: 1.0876 - accuracy: 0.6801 - precision: 0.5373 - recall: 0.6669 - f1_score: 0.4482 - val_loss: 0.7449 - val_accuracy: 0.7535 - val_precision: 0.6013 - val_recall: 0.8642 - val_f1_score: 0.6815 - lr: 5.0000e-05
```

Modelo AlBERT

```
### --------- Modelo AlBERT ---------- ###

model_name = 'albert/albert-base-v2'

max_length = 80

config = AlbertConfig.from_pretrained(model_name)
config.output_hidden_states = False

tokenizer = AlbertTokenizer.from_pretrained(pretrained_model_name_or_path = model_name, config = config)

transformer_albert_model = TFAlbertModel.from_pretrained(model_name, config = config)
```

```
Model: "Albert_MultiClass"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_ids (InputLayer)      [(None, 80)]              0

 albert (TFAlbertMainLayer)  TFBaseModelOutputWithPo   11683584
                             oling(last_hidden_state
                             =(None, 80, 768),
                              pooler_output=(None, 7
                             68),
                              hidden_states=None, at
                             tentions=None)

 pooled_output (Dropout)     (None, 768)               0

 TWEETS (Dense)              (None, 6)                 4614

=================================================================
Total params: 11688198 (44.59 MB)
Trainable params: 11688198 (44.59 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
Epoch 1/5
34/34 [==============================] - ETA: 0s - loss: 1.6063 - accuracy: 0.3154 - precision: 0.3104 - recall: 0.5460 - f1_score: 0.0988
Epoch 1: val_accuracy improved from -inf to 0.33131, saving model to albert-best_model.h5
34/34 [==============================] - 1357s 40s/step - loss: 1.6063 - accuracy: 0.3154 - precision: 0.3104 - recall: 0.5460 - f1_score: 0.0988 - val_loss: 1.5772 - val_accuracy: 0.3313 - val_precision: 0.3150 - val_recall: 0.6300 - val_f1_score: 0.0830 - lr: 5.0000e-05
Epoch 2/5
34/34 [==============================] - ETA: 0s - loss: 1.5917 - accuracy: 0.3230 - precision: 0.3804 - recall: 0.6169 - f1_score: 0.1148
Epoch 2: val_accuracy improved from 0.33131 to 0.33178, saving model to albert-best_model.h5
34/34 [==============================] - 1339s 39s/step - loss: 1.5917 - accuracy: 0.3230 - precision: 0.3804 - recall: 0.6169 - f1_score: 0.1148 - val_loss: 1.5790 - val_accuracy: 0.3318 - val_precision: 0.3150 - val_recall: 0.6300 - val_f1_score: 0.0835 - lr: 5.0000e-05
Epoch 3/5
34/34 [==============================] - ETA: 0s - loss: 1.5911 - accuracy: 0.3201 - precision: 0.3804 - recall: 0.6169 - f1_score: 0.1175
Epoch 3: val_accuracy did not improve from 0.33178
34/34 [==============================] - 1341s 40s/step - loss: 1.5911 - accuracy: 0.3201 - precision: 0.3804 - recall: 0.6169 - f1_score: 0.1175 - val_loss: 1.5768 - val_accuracy: 0.3313 - val_precision: 0.3150 - val_recall: 0.6300 - val_f1_score: 0.0830 - lr: 5.0000e-05
Epoch 4/5
34/34 [==============================] - ETA: 0s - loss: 1.5913 - accuracy: 0.3221 - precision: 0.3804 - recall: 0.6169 - f1_score: 0.1121
Epoch 4: val_accuracy did not improve from 0.33178
34/34 [==============================] - 1344s 40s/step - loss: 1.5913 - accuracy: 0.3221 - precision: 0.3804 - recall: 0.6169 - f1_score: 0.1121 - val_loss: 1.5776 - val_accuracy: 0.3313 - val_precision: 0.3150 - val_recall: 0.6300 - val_f1_score: 0.0830 - lr: 5.0000e-05
Epoch 5/5
34/34 [==============================] - ETA: 0s - loss: 1.5907 - accuracy: 0.3214 - precision: 0.3804 - recall: 0.6169 - f1_score: 0.1116
Epoch 5: val_accuracy did not improve from 0.33178
34/34 [==============================] - 1349s 40s/step - loss: 1.5907 - accuracy: 0.3214 - precision: 0.3804 - recall: 0.6169 - f1_score: 0.1116 - val_loss: 1.5740 - val_accuracy: 0.3313 - val_precision: 0.3150 - val_recall: 0.6300 - val_f1_score: 0.0830 - lr: 5.0000e-05
```

Métricas del modelo BERT

```
Encoded Labels:
{'sadness': 0, 'anger': 1, 'love': 2, 'surprise': 3, 'fear': 4, 'happy': 5}

              precision    recall  f1-score   support

           0       0.34      0.35      0.35      1212
           1       0.00      0.00      0.00         0
           2       0.00      0.00      0.00         0
           3       0.00      0.00      0.00         0
           4       0.00      0.00      0.00         0
           5       0.76      0.34      0.47      3080

    accuracy                           0.35      4292
   macro avg       0.18      0.12      0.14      4292
weighted avg       0.65      0.35      0.44      4292
```

Métricas del Modelo RoBERTa

```
Encoded Labels:
{'sadness': 0, 'anger': 1, 'love': 2, 'surprise': 3, 'fear': 4, 'happy': 5}

              precision    recall  f1-score   support

           0       0.80      0.47      0.60      2101
           1       0.15      0.62      0.24       149
           2       0.00      0.00      0.00         0
           3       0.00      0.00      0.00         0
           4       0.07      0.75      0.13        52
           5       0.86      0.59      0.70      1990

    accuracy                           0.54      4292
   macro avg       0.31      0.41      0.28      4292
weighted avg       0.79      0.54      0.63      4292
```

Métricas del modelo DistilBERT

```
Encoded Labels:
{'sadness': 0, 'anger': 1, 'love': 2, 'surprise': 3, 'fear': 4, 'happy': 5}

              precision    recall  f1-score   support

           0       0.85      0.73      0.79      1423
           1       0.75      0.74      0.75       578
           2       0.30      0.69      0.42       147
           3       0.57      0.67      0.62       156
           4       0.62      0.78      0.69       414
           5       0.85      0.78      0.81      1574

    accuracy                           0.75      4292
   macro avg       0.66      0.73      0.68      4292
weighted avg       0.78      0.75      0.76      4292
```

AlBERT

```
Encoded Labels:
{'sadness': 0, 'anger': 1, 'love': 2, 'surprise': 3, 'fear': 4, 'happy': 5}

              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       0.00      0.00      0.00         0
           2       0.00      0.00      0.00         0
           3       0.00      0.00      0.00         0
           4       0.00      0.00      0.00         0
           5       1.00      0.33      0.50      4292

    accuracy                           0.33      4292
   macro avg       0.17      0.06      0.08      4292
weighted avg       1.00      0.33      0.50      4292
```